Implementing Physics-Informed Neural Networks to solve the Wave Equation and Predict the Propagation of Ocean Waves

Ashley Daniel Ulumben
University of Alabama
Dept. of Computer Science
adanielulumben@crimson.ua.edu

***Abstract*—The introduction of Physics-Informed Neural Networks (PINNs) in machine learning has reduced the difficulty of solving complex Partial Differential Equations (PDEs). This paper aims to implement PINNs to solve the second-order PDE wave equation. The paper systematically solves the 1-dimensional wave equation using the PINN model and then attempts to predict the accuracy of forward wave propagation of ocean waves at various time steps. Our research attempts to solve the linear wave equation using spatial and temporal data and creating an appropriate loss function to model one of the most fundamental modeling equations in oceanic wave monitoring applications. It applies synthetic data to train and test the model predictions and then modifies the Initial and Boundary conditions to train and test the model predictions on a small real-world dataset containing ocean behavior. Additionally, this paper has also utilized the traditional Finite-Difference Time Doman (FDTD) method to numerically solve the wave equation using synthetic data and compare the PINN model's results against the FDTD method. The results generated by the PINN model are accurately close to the true propagation and elevation values of the oceanic values. The resulting waves are visualized in 2-dimensional and 3-dimensional plots, that depict the crests and troughs of the wave. In this paper the solution implemented by the PINN model can help us understand ocean behavior during stormy and normal weather conditions, effectively detecting tsunami threats and preparing us for any natural disasters in coastal areas.**

***Index Terms*—Physics-Informed Neural Network (PINN), Partial Differential Equations (PDEs), wave equation**

I. Introduction

With the advancement of technology, we are now able to record the physical features of nature into data points that can be studied for patterns and anomalies by researchers. Water encompasses the entire planet, and understanding the behavior of oceans, seas, rivers, and lakes can help us predict the propagation and elevation of oceanic tsunami waves during disasters and the directional flow of these water bodies. The importance of understanding the water dynamic during calm and turbulent weather will help to safeguard human lives and valuable buildings. In the event of an Earthquake or cyclone that violently disrupts the ocean bed and water surface, we can evaluate the potential impacts of the waves on coastal communities and ensue the disaster management plan.

The Ocean waves are in perpetual movement and are at times subjected to multiple pressures that propagate them violently, such as wind speed and directions, twisters, storms, and earthquakes. Translating the Ocean dynamics to a model that can be computed by a Neural Network is a highly complex task that requires us to understand the physical factors (features) influencing the wave's behavior. It is possible to express the factors and behavior of water bodies using Mathematical models such as the Wave Equation and the Navier-Stokes equations.

The motivation behind solving the Wave equation stemmed from the reason to understand the physical laws governing water waves and what pressures and disruptive sources can make ocean waves dangerous for coastal areas. Additionally, the benefit of solving the Wave Equation using Deep Neural Networks is that it can be applied to analyze and predict Sound Waves, Lightning Waves, and Seismic Waves within their respective environments.

Physics-Informed Neural Networks (PINNs) can translate and encode the Physical laws of a given data set and perform function approximation to solve Partial Differential Equations (PDEs). An advantage of utilizing PINNs is that they are flexible and can solve multiple complex PDEs without undergoing specific modifications. This generalization of PINN is appealing since it can be used to solve variations of the wave equation ranging from water, sound, seismic, and light waves. The paper first identified the underlying physical behaviors of water waves and then utilized synthetic data to simulate water data that was then used to train and test the neural network. PINNs are trained by supervised learning techniques to solve the general nonlinear partial differential equations.

The approach we take is initially to solve the numerical solution of the wave equation using the Finite-Difference Time-Domain Method and then use that as the basis for training and testing the PINN model. Once the PINN model is validated by comparing it to the FDTD solution, we run the real-world ocean dataset on it. The strategy of employing traditional and modern machine learning techniques enhances the accuracy of the predictions. Also, it enables the application with minor modifications to work on other types of wave-based phenomena. The ultimate goal of our research is to create a robust predictive model that not only predicts the wave outcomes with high accuracy but also provides deeper insights into the physical processes driving these phenomena. This approach ensures that our PINN application will help in environmental science, and disaster management and offer a new method in predictive modeling of natural systems.

II. Differential Equation Overview

This paper aims to solve the 1-dimensional wave equation which is a second-order Partial Differential Equation (PDE) that describes the physical laws of the Mechanical and Electromagnetic waves. To solve the general 1-dimensional scalar wave equation which is modeled to solve the propagation of waves using the spatial and temporal data, the initial and boundary conditions of the physical environment must be specified. Below is the representation of the Wave Equation in Mathematical form.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

Fig. 1. Wave Equation

The above simple equation is the representation of the 1-dimensional wave equation, where u(x,t) represents the wave function, that is dependent on the spatial position 'x' and the variable time 't'. The speed of the wave is represented by the variable 'c', which is the constant speed for the chosen medium. It describes the acceleration of wave function in the given space and time, essentially specifying how the wave propagates at a constant speed through the medium. The wave equation models the system where the speed of the wave is influenced by the medium's properties. In terms of water waves, from any water bodies such as oceans, seas, lakes, rivers etc., the variable 'c' could potentially represent the speed at which the oceanic waves propagate.

The technique to solve the Wave Equation using Physics-Informed Neural Networks (PINNs), is to first identify and develop a Loss function that penalizes deviations from the wave equation and the discrepancies between the true and predicted values. The advantage of implementing PINNs is that they can dynamically handle the boundary conditions and learn from small datasets which makes it beneficial for other complex scenarios such as modeling the ocean wave dynamics under different environmental pressures.

III. Background and Related Work

The solution for modeling the elevation of the Ocean Wave in this paper has been the combination of multiple literature reviews from the papers, **M. Raissi et al 2019**, **de Wolff et al., 2021**, and **Liu, Yue et al., 2024** The Wave Equation can be interpreted to solve mechanical waves such as water, sound, and seismic waves and also be applied to electromagnetic light waves. This requires us to understand the underlying physical laws of the water waves since our research focuses on the elevation of oceanic waves. The solution derived for the wave equation reflects the characteristics of waves, such as their speed, their behavior under different boundary conditions, and external stimuli and pressures of the environment.

The traditional method for solving the wave equation is using the Finite Difference Time Doman (FDTD), finite element, and spectral methods. While these methods are quite powerful, they require computationally expensive resources and discretization schemes for complex boundary conditions and nonlinear equations. **Rasht-Behesht et al., 2022** describes FDTD as a widely utilized numerical technique for solving partial differential equations, especially effective for the second-order wave equation. It still produces highly accurate predictions but is slightly complex and domain-specific. Only after understanding the FDTD numerical method, we can create our deep-learning Physics-Informed Neural Network (PINN) model architecture. From the paper **S. Alkhadhr et al., 2021**, we have adapted the concept of comparing our PINN model to the traditional FDTD model that is trained on synthetic (simulated) wave data and then also used as the test results from the FDTD model as supervised learning for the PINN model.

Dirichlet Boundary Conditions

To solve Partial Differential Equations (PDEs) such as the Wave Equation, we also need to understand the important usage of the Dirichlet and Neumann boundary conditions that define the behavior of the solution at the boundaries of the wave domain. The Dirichlet boundary conditions specify the value of the solution at the boundary, for example, the displacement of the wave at the boundary to zero (u(x,t) = 0) simulates a fixed end where the wave cannot propagate beyond this measure, effectively reflecting the wave back into the medium.

Neumann Boundary Conditions

The Neumann boundary conditions specify the gradient of the solution at the boundary, which is also typically set to zero at the edges. This condition models a free end that allows the wave to move freely at the boundary which essentially simulates partially absorbing or reflective conditions. Both these conditions are important to accurately model the physical systems that influence the behavior of the wave and ensure that the numerical solution conforms to the expected natural physical laws of the wave.

The Courant-Friedrichs-Lewy (CFL)

It is a condition that is an important criterion for the stability of numerical methods used to solve partial differential equations, especially in those that simulate wave propagation, like the wave equation. It is important to understand this condition because during the code implementation, it will help in simulating a stable wave model in the controlled environment. The CFL condition is crucial because it ensures that the numerical method does not produce any instabilities, essentially helping to maintain accuracy and stability during the model's training.

IV. Methods

In this section, we will describe the method and techniques utilized to solve the second-order wave equation using Neural Networks and then apply the Physics-Informed Neural Network (PINN) model to predict the propagation and elevation of oceanic waves within a given timeframe. **M. Raissi et al 2019** describes the function approximation method to solve the scalar Wave Equation. The model was trained and tested in a controlled environment, by simulating a mesh structure in which the wave behavior can be studied. To solve the complex theoretical physics wave model, the first method utilized is to solve the general 1-dimensional scalar wave equation using the Finite-Difference Time Domain (FDTD) method.

To solve the scalar wave equation represented in Fig 1.0, where the function u(x,t) represents the displacement of the wave at position x and time t, and then c is the constant wave speed in that particular medium. The FDTD method begins by discretizing the wave equation on the spatial and temporal grid. This discretization method approximates the second derivates using the central differences. The approximation of the second-order spatial derivative is shown in the figure below.

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u^j_{i+1} - 2u^j_i + u^j_{i-1}}{(\Delta x)^2}$$

Fig. 2. Second-order spatial derivative

Similarly, Figure 3 represents how the Second-order temporal derivative is approximated.

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u^{j+1}_{i} - 2u^j_i + u^{j-1}_{i}}{(\Delta t)^2}$$

Fig. 3. Second-order Temporal derivative

By equating the spatial and temporal approximations according to the wave equation, we can solve for 'u_i^{j+1}', the displacement at the next time step: The code then initializes the

$$u_i^{j+1} = 2u_i^j - u_i^{j-1} + (c^2 \frac{(\Delta t)^2}{(\Delta x)^2})(u_{i+1}^j - 2u_i^j + u_{i-1}^j)$$

Fig. 4. Displacement at the next time step

wave profile at t=0 using the customized function that defines the initial shape of the wave. The first timestep "u_i^{j+1}" then initializes the time evolution, considering the second derivative in time:

The Boundary conditions are applied at each timestep. The Dirichlet conditions set the boundary value to zero, which essentially reflects the boundary, and the Neuman conditions maintain the gradients across the boundary. In our code this computation is looped over the time steps, updating the wave

$$u_i^{j+1} = u_i^j + 0.5 \cdot (c^2 \frac{(\Delta t)^2}{(\Delta x)^2})(u_{i+1}^j - 2u_i^j + u_{i-1}^j)$$

Fig. 5. Second derivative in time

profile at each grid point using the discretized wave equation. This method then updates all the grid points based on their previous values and the values of their adjacent neighbors.

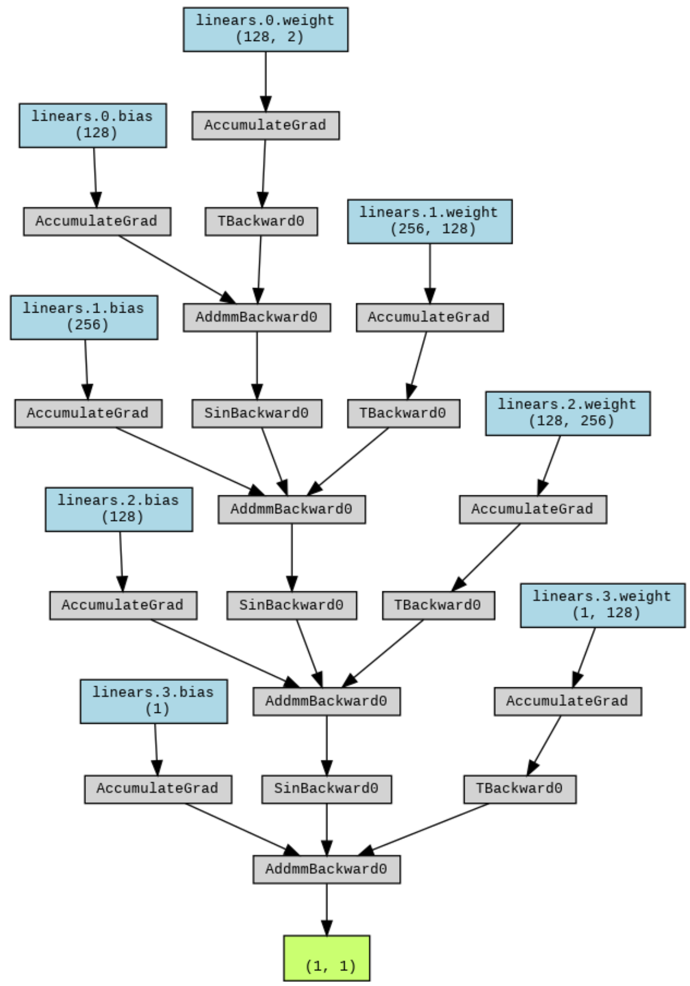

Fig. 6. Neural Network Architecture

A. *Model Architecture*

The Physics-Informed Neural Network (PINN) and its layers and activation functions are a complex structure to model since the model requires understanding the underlying physical aspects of the water wave. The model employed in this paper, visualized in Fig 6, comprises of a sequential arrangement of fully connected layers. The input layer accepts a 2-dimensional input, indicating the spatial and temporal dimensions being fed into the neural network. The network has three hidden layers with dimensions of 128, 256, and 128 neurons. The neural network utilizes the sinusoidal activation function to capture the wave-like behavior similar to the physical phenomena of waves under analysis.

The final layer consists of a single output neuron, that represents the solution to the physical wave problem. The Xavier Normal Initialization method is used to initialize all the weights of the PINN, this helps in achieving an effective gradient flow during the training process. To begin with, the biases are initialized to zero, which ensures a neutral starting point for bias optimization. Throughout the network's backward pass, the gradient accumulation is carefully managed, with each layer's gradients being calculated by a combination of tensor operations and backpropagation algorithms, that ensure precise adjustment of network parameters through optimization techniques.

V. Implementation and Experimentation

A. Code Implementation and PINN Model Setup

The code utilizes multiple Python libraries and modules to build the Physics-Informed Neural Network. The primary library used is PyTorch, which provides the ecosystem for constructing and training the neural networks. The 'autograd' module from PyTorch is utilized for automatic differentiation and calculation of the gradients, which enables the backpropagation algorithm required for neural network training. It computes the loss of the weights in the network and updates the model's parameters during optimization. The Tensor module of PyTorch is the fundamental data structure that is fed into the neural network for training.

The First step in the code implementation is to preprocess the Dataset and normalize the required values. We utilized the Pandas and Numpy Libraries to preprocess and normalize the values, 'Longitude' for the spatial domain, 'DateTime' is translated to the 'Timestep' (seconds) of the wave for the temporal domain. The target value is the 'SignificantWaveHeight'. The code standardizes the positional data to a specified range and normalizes the input features to improve the model's training convergence.

The next phase is to set the training parameters of the model designed to solve the PDE. We have initially set the number of iterations of the training model to be 10,000 steps. The learning rate is set to a very low value. The spatial domain size extends from -1 to 1 meter, with a discretization by a mesh size 'dx' of 0.01 meters. The temporal domain is defined from 0 to 0.2 seconds, with 'dt', the timestep size set to 20 percent of the spatial mesh size which adheres to the Courant-Friedrichs-Lewy (CFL) condition for numerical stability. The central position of the initial point of the wave disturbance is set to 0, and the 'sigma' value is set to 0.05 which controls the spread of the wave's initial distribution.

The code begins by defining the initial condition of the wave at time t = 0, using a Gaussian distribution centered around the position of the source, with its spread controlled by the sigma value which defines the width of the source. This setup is important to simulate the evolution of the waveforms in the controlled environment. The velocity of the medium is set as constant across the entire spatial domain. Then the boundary conditions are specified for both the left and right ends. The code uses the Neumann Boundary Condition.

The code implements the numerical Finite Difference Time Domain (FDTD) method to solve the wave equation. The FDTD method discretizes the wave equation over a grid defined by spatial (Δx) and temporal (Δt) steps, adhering to the Courant-Friedrichs-Lewy (CFL) condition for stability. The code initializes the arrays to store the values at different timesteps. The initial profile of the wave is defined and set across the spatial domain, and the first step in time is calculated using a Taylor expansion, which considers the wave equations second nature. The reason for implementing the numerical solution using the FDTD method is to compare the error rate or the accuracy of the PINN model. This comparison is done on synthetic data, which means predefined values of the physical properties of the wave.

The main computation loop, iterates over timesteps, updating the wave profile at each grid point based on the second-order central difference approximations for both the spatial and temporal derivatives. The boundary conditions are applied at each time step, which ensures the simulation adheres to the physical constraints of the domain's edges. The computed wave profiles for each time step are stored in an array that is used for further analysis of and visualization of the waves.

B. Initial Conditions

The initial condition of the code is represented by an array of spatial points within the domain [-1, 1], evaluated by using a Gaussian distribution function to represent the wave profile at time t=0. The initial wave is shaped like a bell curve, centered at the point x_0, and spreads out on both sides. The width of this curve tells us signifies the rise and fall and is determined by the value of Sigma σ. This logic helps us understand the starting shape of the wave before it begins to move within the built environment.

C. Boundary Conditions

The code implements two types of Boundary conditions respectively Dirichlet and Neumann Boundary conditions to simulate the wave dynamics. By implementing the Neuman Boundary condition we set the rate of change of the wave's height in the space to zero, which means that these edges of the wave are neither rising nor falling. The boundaries x=-1 and x=1 are set to zero. This is used because it aligns with the physical constraints of the wave fluctuations within a controlled environment. Both the initial and boundary conditions are incorporated within the loss function of the PINN model to enforce the physical laws that simulate the wave behavior at the boundaries during the training process.

D. Loss Function

The Loss function is crucial while implementing PINNs to solve the wave equation. It is designed to enforce the physical correctness and accuracy of the solution. A mean squared error (MSE) loss is incorporated by the model. The 'mean' reduction method averages the squared differences between the predicted and true values across all the input samples. The loss function is expanded to address the different aspects of the physical model, such as the initial and boundary conditions, and the adherence to the physical laws of the wave equation.

E. Experimentation

The PINN model was initially trained for 2000 steps on synthetic data, with a low learning rate. The resulting Ocean wave generated was compared to the numerical solution of the FDTD method. Then we fed the PINN model the preprocessed Ocean Dataset values and trained it for 1000 epochs. Another training cycle of 1000 steps was executed on a smaller dataset to see the variance of the Ocean Waves from different coastal areas, which can be observed in Figures 8 and 9. The training time lasted an hour while using Google Colabs free GPU for processing. The Ocean Wave is visualized by using the 'Matplotlib' library to plot the mesh in which the Wave propagation is recorded for 0 to 0.2 seconds.

VI. Results

The Results section comprises the visualizations of the Physics-Informed Neural Network (PINN) model predictions on the wave behavior taken from the real-world dataset and the synthetic data. There is also the comparison of the accuracy of the PINN model against the traditional Finite-Difference Time Domain FDTD method. The PINN model's loss rate also differs from different ocean wave datasets.

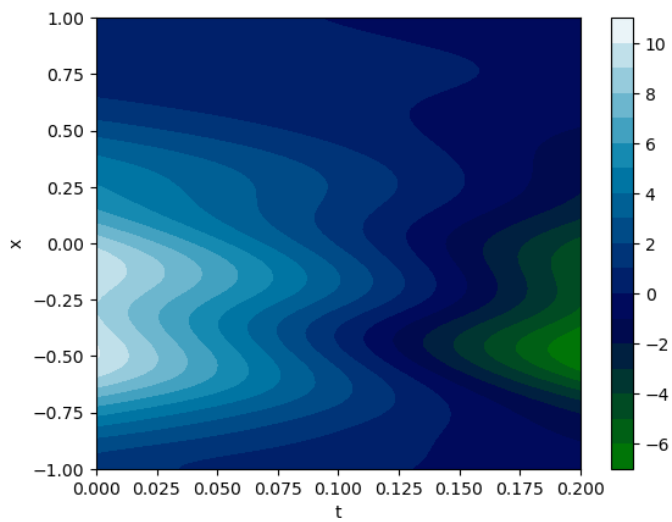

Fig. 7. 2-dimensional Plot of the Ocean Wave

The first plot depicted in Fig 7, is a contour plot, a technique to display the ocean wave's surface on a two-dimensional plane. Fig 8 is the 3-dimensional representation of the same ocean wave. The figures displayed effectively visualize the amplitude of the wave as a function of position 'x' and time 't'. This visualization of the PINN model's prediction on synthetic data is later compared to the numerical solution of the wave equation using the FDTD method, which compares the accuracy of both models.

Characteristics of the Ocean Waves 2D and 3D plot:

- The x-axis labeled as 't' represents the timesteps of the wave. The wave elevation and propagation time that is measured from 0 to 0.2 seconds.
- The y-axis labeled as 'x' represents the spatial dimensions of the wave in the controlled medium. It depicts the origin of the Wave and how the shape of the wave changes from negative to positive values.
- The white-blue region indicates the highest point of the wave and the dark blue-green represents the lowest point of the wave curve.
- The color scale signifies the magnitude of the wave function at any given point in space and time.
- The concentric patterns in the figure represent areas of similar value and the closer these lines are the steeper the change in the value, which indicates a higher gradient or rate of change in the wave's magnitude.
- For the 3D plot the t-axis represents the timestep and the x-axis represents the positional data of the wave (spatial dimension), while the z-axis represents the magnitude of the wave function u(x,t).

Figure 8 is the 3-dimensional visualizations of the PINN model prediction on the real-world ocean dataset. The chosen dataset describes the oceanic waves from the different coastal regions of Queensland.

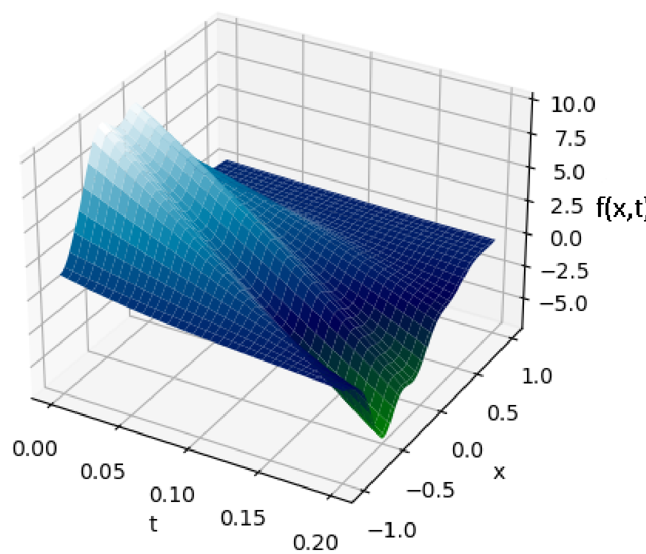

Fig. 8. 3-dimensional Plot of Ocean Wave

From Fig 8 we observe that the lower 't' values of the wave's amplitude appear to be closer to 0, indicating the start of the wave propagation. As the time increases the wave's amplitude also increases significantly reaching a peak at the middle of the time axis, before going down again. The timeline of the wave indicates the rise and fall pattern of the wave within 0.2 seconds. This plot is based on the real-world dataset from Queensland coastal areas. The consistent shape of the wave as it extends from negative to positive values in the spatial domain suggests that the wave is uniform within the simulated medium (mesh). The Peaks and troughs represent the crests and valleys of the wave as it propagates through the ocean.

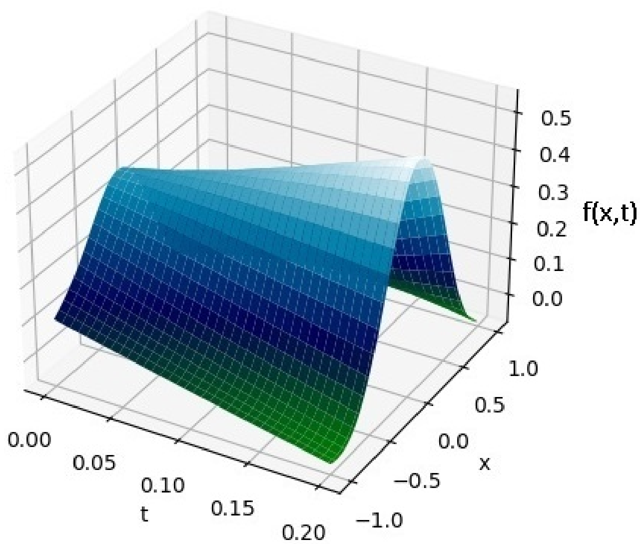

Fig. 9. 3D plot based on a smaller dataset

The wave in Fig 8 was trained and tested on a much larger dataset consisting of 6000 values spanning across 15 regions of the coast and the training cycle lasted for 2000 iterations, approximately an hour. Our current PINN model can predict the wave's elevation and propagation for only a tiny point in the ocean for a split second, and hence we have to further work on expanding the simulated environment, for better wave predictions.

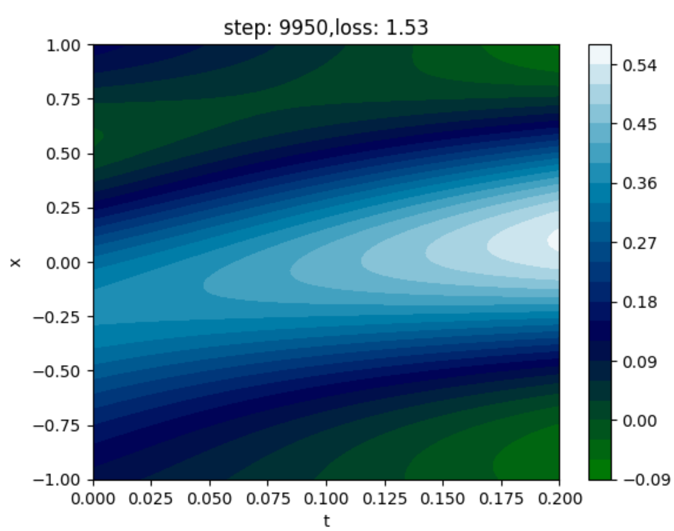

Fig. 10. 2D plot based on a smaller dataset

Figure 9 is another visualization of a small part of the Queensland Wave Dataset for 1000 training iterations. This plot considered approximately 3000 values from the dataset spanning across 9 regions. We can observe, it has a smooth tide elevation and curvature. The time evolution of the wave might indicate how the wave amplitude grows and decays, possibly due to the energy input and dissipative forces of the water. Fig 10 is the 2D contour plot representing the same ocean wave modeled from the real-world dataset shown in Figure 9. It has a loss of 1.53%

Comparison of FDTD method Vs PINN model

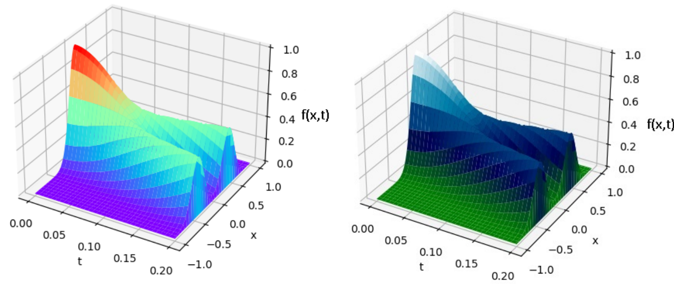

Fig. 11. 3D Comparison Plot FDTD Vs PINN Model

The paper has also implemented the traditional Finite Difference Time Domain (FDTD) method for a comparative analysis of the PINN model's accuracy. We have utilized numerical synthetic data to mimic the properties of the wave and solve it using the FDTD method. This same numerical synthetic data is fed into the PINN model for training and testing.

In Figure 11 the FDTD solution is represented by the Rainbow color while the PINN model is in blue and green. It illustrates the FDTD and PINN model's solution of the wave equation on synthetic numerical data. We can observe that the PINN model's wave plot is the exact replica of the FDTD method's solution to the wave equation. Since we use numerical data and apply the long-trusted traditional mathematical FDTD method to solve the wave equation, the solution we receive is precise and reliable. When the PINN model's solution is compared to the FDTD solution the result received is the same which signifies that our PINN model is also accurate and reliable and successfully predicts wave propagation and elevation.

These visualizations help us understand the characteristics of the wave solution and the behavioral pattern of the waves at a given time. The plots depict the propagation of the waves and how they travel through space over time. It clearly shows the height of the wave crests and the depths of the troughs. It highlights the frequency characteristic of the wave, visualizing how often the wave crests occur within a given time and the wave speed which is implied by the slope of the wave crest in the 2D contour plot and the incline of the waves in the 3D plot. However, this is a basic PINN model, and it would require a much more complex model and simulation environment to predict the wave propagation and elevation patterns near the coastal areas.

PINN model Loss

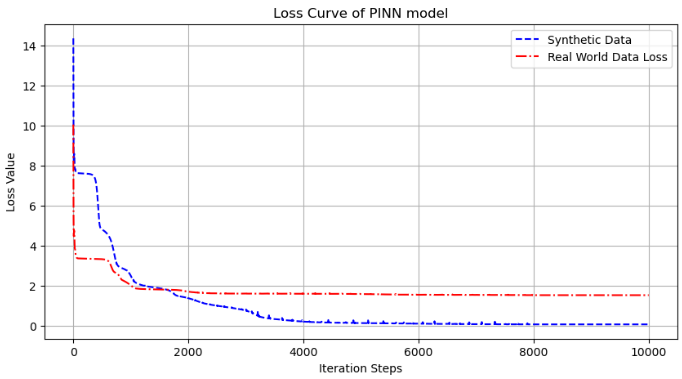

Fig. 12. Loss Curve of the PINN model

Fig 11 depicts the loss rate of the PINN model during training. The Red dotted line represents the PINN model's loss on synthetic data and the blue dotted line represents the model's loss on real-world ocean data. For both instances, the number of iteration steps (training period) of the PINN model is 1000 steps. For real-world ocean data it can be observed that at the beginning of the training, there was a loss of about 3 percent which decreased and normalized with the training time, with the loss rate below 2 percent. While the PINN model's loss on the synthetic data was much lower, approximately less than 0.5% during the initial training phase it had a higher loss rate of about 8%Ṫhe loss rate is higher for real-world data because there are multiple factors such as wind speed, current direction, and speed affecting the wave's shape and physical properties that are not easily emulated by the synthetic data. From the 3D plot visualizations and the loss curve of the PINN model, we can clearly understand that our model performs well, and that the accuracy is high and the loss rate is quite low.

VII. Discussion

The methods to reduce the complexity of solving the 1-dimensional wave partial differential equation using the Physics-Informed Neural Networks (PINNs) architecture were discussed with peers. We extended the solution derived from the PINN model to predict the propagation of waves at various time steps. We also discussed how different wave equation datasets must be preprocessed into a uniform structured format for the PINN model to evaluate. The process to normalize the dataset was complex, we first had the understand the underlying physical laws of the Wave, and the features (factors) that influence the propagation of water waves and then concatenate our features into a single input tensor for time and spatial data. Hence we first trained and tested the model using synthetic data and then applied them to sea water Dataset.

The PINN model in this paper was only tested on the ocean Dataset of particular Queensland coastal regions, and further training and testing must be done on datasets from different geographical areas with different topographical and atmospheric conditions. Additionally, the features extracted from the dataset were limited to a few, only the Positional elements such as the Latitude and Longitude, the timestep representing the temporal data, and the target variable 'SignificantWaveHeight'. The Future research aspects would be to factor in the surface and ground tension features such as the force of the wind on water waves and the vibrations from the ocean bed.

Challenges and Limitations

The challenges faced during the coding implementation were to initially comprehend the physical laws of the wave equation and then understand and translate the numerical solution into programmable logic for modeling the neural network. The next challenging step was to build a simulated controlled environment (the mesh), in which a subset of the wave's dataset could be trained and tested. Normalizing the dataset value to match the grid parameters of the simulated environment is difficult since the dataset provides Latitude and Longitude which need to be scaled down and normalized to represent the spatial parameters of the wave. We utilize the Longitude to represent the positional space of the wave at different time intervals within the mesh. Additionally, the training of the PINN model is computationally expensive and hence takes a long time to train on regular CPUs, and we had to make use of High-Performance Computers to train and test the model.

VIII. Conclusion

We implemented a technique to solve the linear wave equation by modeling its underlying physical laws into the Physics-Informed Neural Networks (PINNs). The PINN captures the information of the physical properties of the water waves from the Queensland dataset and predicts the propagation of the waves from different sectors of the coast. The code implemented the numerical solution of the water wave's Partial Differential Equation within the PINN model resulting in accurate wave predictions. The PINN model was successfully able to predict the propagation and elevation of the wave with minimal error loss. The PINN model introduced in this paper can potentially help in wave monitoring applications, and help in notifying the coastal areas for abnormalities in the wave patterns. By utilizing PINNs we are able to capture the complex physical properties of real-world elements and compute them in a highly efficient and accurate manner compared to the traditional methods used to solve partial differential equations.

IX. Acknowledgements

The authors of this paper would like to express their sincere gratitude to Professor Jiaqi Gong, whose expert guidance and advice were instrumental in the completion of this Project. Professor Gong's knowledge and teaching on Partial Difference Equations helped us understand the role of Deep Neural Networks in Physics. We are deeply appreciative of his support and insights throughout the development of this research.

REFERENCES

[1] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, Volume 378, 2019, Pg 686-707, ISSN 0021-9991, https://doi.org/10.1016/j.jcp.2018.10.045.

[2] Rasht-Behesht, M., Huber, C., Shukla, K., and Karniadakis, G. E. (2022). Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions. Journal of Geophysical Research: Solid Earth, 127, e2021JB023120. https://doi.org/10.1029/2021JB023120

[3] S. Alkhadhr, X. Liu and M. Almekkawy, ”Modeling of the Forward Wave Propagation Using Physics-Informed Neural Networks,” 2021 IEEE International Ultrasonics Symposium (IUS), Xi’an, China, 2021, pp. 1-4, doi: 10.1109/IUS52206.2021.9593574.

[4] de Wolff, Taco, Hugo Carrillo, Luis Martí, and Nayat Sanchez-Pi. ”Towards optimally weighted physics-informed neural networks in ocean modeling.” arXiv preprint arXiv:2106.08747 (2021).

[5] Liu, Yue and Zhang, Xiantao and Dong, Qing and Chen, Gang and Li, Xin. (2024). Phase resolved wave prediction with linear wave theory and physics informed neural networks. Applied Energy. 355. 121602. 10.1016/j.apenergy.2023.121602.

[6] **Dataset Resource** - The State of Queensland, Environment, S., and I. (2024) Coastal Data System – near real-time wave data - wave data - open data portal: Queensland government, Wave data - Coastal Data System – Near real-time wave data -. Available at: https://www.data.qld.gov.au/dataset/coastal-data-system-near-real-time-wave-data