A Deep Learning Framework for Solving Rectangular Waveguide Problems

Xiaolin Hu
School of Communication and Information Engineering
Shanghai University
Shanghai, China
xiaolinhu@shu.edu.cn

Nicholas E. Buris
School of Communication and Information Engineering
Shanghai University
Shanghai, China
neburis@shu.edu.cn

***Abstract*—We employ a class of deep learning, known as Physics Informed Neural Networks (PINNs), to solve rectangular waveguide problems. In particular, we successfully apply PINNs to the task of solving electric and magnetic fields, which can be described by partial differential equations (PDEs). PINNs are neural networks that can naturally encode PDEs into the loss function, while partial derivatives with respect to input variables can be calculated through automatic differentiation (AD) built in modern deep learning libraries. We also show the applicability of the framework for predicting the unknown parameters such as wavenumber. We validate our model in terms of the convergence speed and accuracy. In addition, we suggest an approach to scale the input vector in order to accelerate training.**

Keywords—Rectangular waveguide, Deep learning, Physics-informed neural networks, Boundary value problem.

I. INTRODUCTION

Over the past few years, deep learning has been used very successfully in a broad range of applications including computer vision [1], natural language processing and cognitive sciences [2]. In these applications, large amounts of data are needed to train the neural network by minimizing the distance (loss) between the network output and the ground truth [3]. In many instances of analyzing physical systems [4], the cost of data acquisition is high, but the physical laws of the system are known to us. To empower the network model with known physical laws, Raissi et al. [5] proposed Physics Informed Neural Networks (PINNs), which aims to solve direct and inverse problems governed by several different types of PDEs. The loss function of a PINN consists of several terms, including different governing equations and boundary conditions. In particular, the training data used in this framework can be sampled from the domain of definition.

PINNs have been used for solving direct and inverse problems governed by PDEs in various fields, such as material science [6] and fluid mechanics [7]. In this paper, we focus on the application of PINNs to solve the mode problem in rectangular waveguides [8]. Our intention here is not to substitute the traditional solutions to eigenvalue problems [9]. Rather, we aim at investigating the applicability of PINNs to well-known electromagnetic boundary value problems [10].

II. PROBLEM SET-UP AND METHODOLOGY

A. Rectangular Waveguide Problem

Consider a rectangular waveguide problem represented in Fig. 1, where a and b are the width and the height of the rectangular cross section, respectively. Seeking solutions representing propagating waves along the waveguide, the z component of the magnetic fields is given by $H_z(x,y,z) = H_z(x,y)e^{-j\beta z}$. Let $\Omega \in \mathbb{R}^2$, then the original 3-D problem reduces to a 2-D Helmholtz equation:

$$\frac{\partial^2 H_z}{\partial x^2} + \frac{\partial^2 H_z}{\partial y^2} + (k^2 - \beta^2)H_z = 0, \forall x, y \in \Omega. \qquad (1)$$

Where $k = 2\pi f/c$, f is the frequency, c is the speed of sound, $H_z \equiv H_z(x,y)$ is the unknown 2-D field and β is the unknown wavenumber. The object of this study is to solve for $H_z(x,y)$ and β using PINNs.

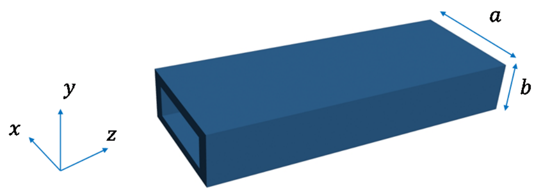

Fig. 1 Geometry of the rectangular waveguide.

As a special case, we restrict ourselves to the Transverse Electric, TE_{mn}, modes, which have $E_z = 0$. All other components of electromagnetic (EM) fields can be derived from Hz using the following equations

$$\begin{bmatrix} E_x \\ E_y \end{bmatrix} = \frac{j\omega\mu}{k^2-\beta^2}\begin{bmatrix} -\frac{\partial H_z}{\partial y} \\ \frac{\partial H_z}{\partial x} \end{bmatrix}, \begin{bmatrix} H_x \\ H_y \end{bmatrix} = -\frac{j\beta}{k^2-\beta^2}\begin{bmatrix} \frac{\partial H_z}{\partial x} \\ \frac{\partial H_z}{\partial y} \end{bmatrix} \qquad (2)$$

The pertinent boundary conditions (BC) are

$$E_x(x,0) = E_x(x,b) = 0 \qquad (3)$$

$$E_y(0,y) = E_y(a,y) = 0 \qquad (4)$$

In order to investigate the applicability of PINNs to this problem, we choose to excite the waveguide with a current distribution which we know is going to generate pure TE_{m0} modes. Thus, H_x is specified by

$$H_x(x,y) = J_{sy} sin\left(\frac{m\pi}{a}x\right), \forall x, y \in \Omega. \qquad (5)$$

where J_{sy} is an arbitrary amplitude. From equations (2-5), we can obtain the constrains on H_z

$$\frac{\partial H_z}{\partial x} = J_{sy}\frac{k^2-\beta^2}{-j\beta} sin\left(\frac{m\pi}{a}x\right), \forall x, y \in \Omega. \qquad (6)$$

$$\frac{\partial H_z}{\partial x} = 0, at\ y = 0\ and\ y = b \qquad (7)$$

$$\frac{\partial H_z}{\partial y} = 0, at\ x = 0\ and\ x = a \qquad (8)$$

The well-known and exact wavenumber, β, is given by

$$\beta = \sqrt{k^2 - \left(\frac{m\pi}{a}\right) - \left(\frac{n\pi}{b}\right)} \quad (9)$$

By combining (2) and (5) we can obtain the exact H_z

$$H_z(x,y) = \frac{k^2-\beta^2}{j\beta}\frac{\alpha}{m\pi}J_{sy}\cos\left(\frac{m\pi}{a}x\right) \quad (10)$$

B. Introduction to Physics-Informed Neural Network

In this section, we briefly review the architecture of PINNs, following prior published work [5]. Let $\mathcal{N}^L(\mathbf{x};\mathbf{W},\mathbf{b}):\mathbb{R}^{d_{in}}\rightarrow\mathbb{R}^{d_{out}}$ be an $L-$layer neural network with input vector $\boldsymbol{x}$, network parameters $\mathbf{W},\mathbf{b}$ and N_ℓ neurons in the $\ell - th$ layer ($N_0 = d_{in}, N_L = d_{out}$). In this study, we use the feedforward neural network (FNN) [11], which is defined as follows

$$\mathcal{N}^0(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^{d_{in}} \quad (11)$$

$$\mathcal{N}^\ell(\mathbf{x}) = \sigma^\ell(\mathbf{W}^\ell\mathcal{N}^{\ell-1}(\mathbf{x}) + \mathbf{b}^\ell), \text{ for } 0< \ell < L \quad (12)$$

$$\mathcal{N}^\ell(\mathbf{x}) = \mathbf{W}^L\mathcal{N}^{L-1}(\mathbf{x}) + \mathbf{b}^L \in \mathbb{R}^{d_{out}} \quad (13)$$

where $\mathcal{N}^0(\mathbf{x}) \equiv \mathbf{x}$ is the input vector. $\mathbf{W}^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, $\mathbf{b}^\ell \in \mathbb{R}^{N_\ell}$ are the weight matrix and bias vector in $\ell - th$ layer, respectively. The function σ^ℓ denotes the nonlinear activation function in layer ℓ, which operates element-wise.

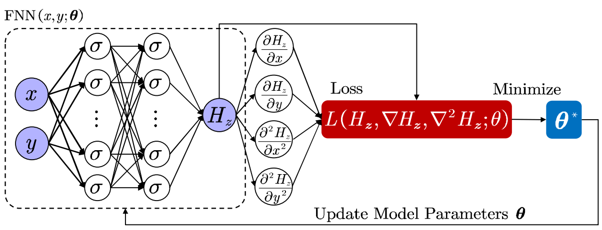

Fig. 2 Schematic of a PINN for solving rectangular waveguide problem based on partial differential equations. The left neural network $FNN(x,y;\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$ represents surrogate models whose output is the PDEs solution.

In Fig. 2 we show the architecture of PINNs. The left part $\text{FNN}(x,y;\boldsymbol{\theta})$ takes coordinates (x,y) as inputs and is trained to predict the solution H_z, where $\boldsymbol{\theta} = \{\mathbf{W}^\ell,\mathbf{b}^\ell\}_{\ell=0}^L$ represents the trainable parameters in the network. By applying the chain rule, the derivatives of H_z with respect to the inputs can be computed automatically [12]. Then the parameters $\boldsymbol{\theta}$ of the neural network can be learned by minimizing the mean squared error loss

$$L = L_{PDE} + L_{data} + \lambda L_{BC} \quad (14)$$

$$L_{PDE} = \left\|\frac{\partial^2 H_z}{\partial x^2} + \frac{\partial^2 H_z}{\partial y^2} + (k^2-\beta^2)H_z\right\|_2^2, (x,y)\in\Omega \quad (15)$$

$$L_{data} = \left\|\frac{\partial H_z}{\partial x} - J_{sy}\frac{k^2-\beta^2}{-j\beta}sin\left(\frac{m\pi}{a}x\right)\right\|_2^2, (x,y)\in\Omega \quad (16)$$

$$L_{BC} = \left\|\frac{\partial H_z}{\partial x}\right\|_2^2 + \left\|\frac{\partial H_z}{\partial y}\right\|_2^2, (x,y)\in\partial\Omega \quad (17)$$

where L_{data}= is given by equation (6), L_{PDE} and L_{BC} are the loss components corresponding to the residual of Helmholtz equation (1) and boundary conditions, respectively. The constant λ is the weight of L_{BC} and $\partial\Omega$ is the boundary (not to be confused with the wavelength).

To accelerate the training of Neural Network, the input vector is multiplied by a scaling factor, γ

$$\mathbf{x}^* = \gamma\mathbf{x} \quad (18)$$

where vector $\mathbf{x}$ presents the original 2-D coordinates (x,y) and $\mathbf{x}^*$ is the scaling coordinates. The wavenumber result is then scaled back appropriately.

III. Results and Discussion

In this section we provide a collection of numerical tests on solving rectangular waveguide problem with PINNs. Consider a computational domain of $[0,a]\times[0,b] = [0, 0.05\text{m}]\times[0, 0.03\text{m}]$, we solve waveguide problem governed by Helmholtz equation and estimate the unknown wavenumber . The performance of PINNs with scaling inputs is evaluated in terms of the convergence speed and the accuracy. For computing the total loss, we randomly select the position of training points. In all experiments, the training of PINNs is performed using FNN with 4 layers, each with 50 neurons. The numbers of training points in the domain and on the boundary are 10,000 and 600, respectively. Moreover, we employ sinusoidal activation functions and train the networks using the Adam optimizer [13] with learning rate 1e-4. We set the weight of boundary condition $\lambda = 20$, scaling factor of input vector $\gamma = 30$. All algorithms are implemented in PyTorch [14] and the computations are performed on a single Nvidia 2080Ti GPU.

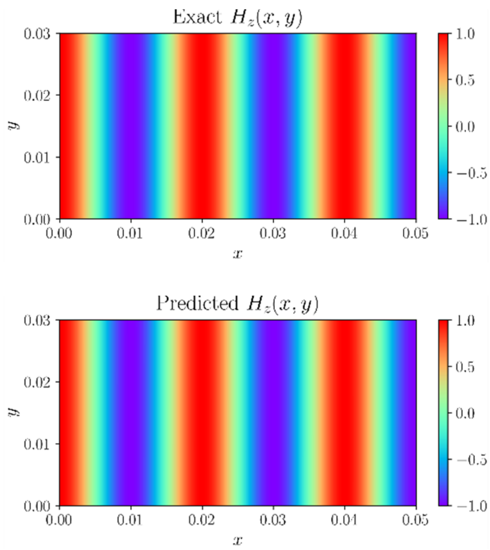

Fig. 3 Exact normalized solution versus the predicted solution of PINN model after 4000 iterations of training. The relative L^2 error is 7.63e-05.

Fig. 3 provides a comparison between the ground truth H_z field component solution and the predicted one, obtained after 4000 iterations at TE_{50} mode. We use frequency $f = 27.57\ GHz$ in this mode. The results show that the predicted solution achieves very good agreement with the exact one and the prediction error is measured at 7.63e-05 in the relative L^2 $-$norm. The exact β is 482.87 rad/m and the predicted β is 482.83 rad/m. For this experiment, the PINNs are trained for 4000 iterations requiring approximately 4 minutes.

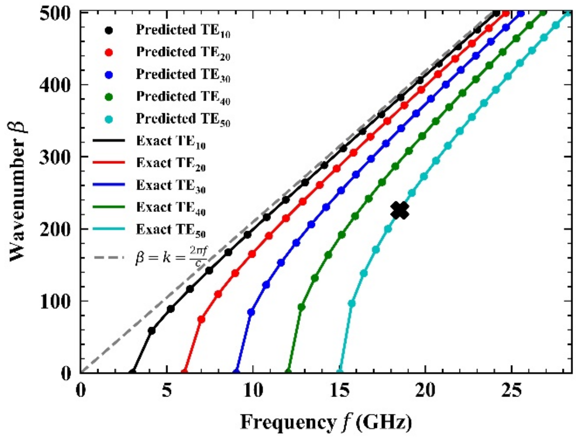

Fig. 4 Exact and predicted dispersion relations for different TE modes.

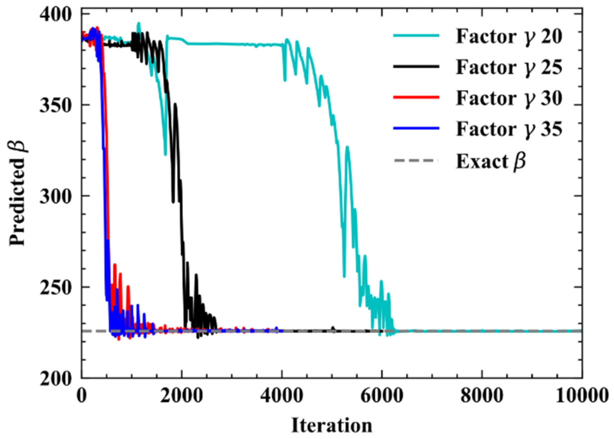

Fig. 5 Convergence of predicted β for different scaling factors at TE_{50} mode.

Fig. 4 depicts the predicted and exact dispersion relations for different TE modes. The number of iterations required for all these results is less than 20,000 and the errors between the predicted β and the exact β are all lower than 1%. We observe that the convergence of PINNs becomes slower when the frequency f is set close to the cutoff frequency. The thick black "x" indicates the point on the dispersion curve where the convergence study shown in Fig. 5 has been conducted.

Fig. 5 shows the convergence of the wavenumber as a function of the scaling factor used in the numerical calculations. Before training, the initial value of β is set to k. The results show the predicted β can converge to the exact value after up to 6000 iterations if the scaling factor is in the range of 20~35. We find that PINNs with $\gamma = 1$ fail to solve this waveguide problem at TE_{50} mode. For TE_{10} mode, PINNs with $\gamma = 1$ need to be trained for approximately 1,000,000 iterations to get solutions. This indicates that the scaling factor greatly speeds up the training. This peculiarity is reminiscent of observations by other researchers [15] and further studies are in progress to better understand it. It is interesting to note that, regardless of the value of the scaling factor, the convergence of the wavenumber is not smooth.

IV. CONCLUSIONS

We investigated the applicability of PINNs to the rectangular waveguide problem. We excited a rectangular waveguide so as to generate TE_z modes and, specifically, the first few of the n=0 modes (i.e. no variation on the y-axis). We demonstrated that PINNs indeed can be applied to solve this problem. We further identified the importance of a scaling technique in the convergence of the numerical results. Additional studies are required to better characterize the mechanics of this convergence.

REFERENCES

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097--1105.

[2] B. M. Lake, R. Salakhutdinov and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," Science, vol. 350, no. 6266, pp. 1332--1338, 2015.

[3] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, The MIT Press, 2016.

[4] M. Raissi, A. Yazdani and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," Science, vol. 367, no. 6481, pp. 1026--1030, 2020.

[5] M. Raissi, P. Perdikaris and 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, vol. 378, pp. 686-707, 2019.

[6] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, "Machine learning for molecular and materials science," Nature, vol. 559, no. 7715, pp. 547--555, 2018.

[7] M. Brenner, J. Eldredge and J. Freund, "Perspective on machine learning for advancing fluid mechanics," Physical Review Fluids, vol. 4, no. 10, p. 100501, 2019.

[8] R. E. Collin, "Foundations for microwave engineering," John Wiley & Sons, 2007.

[9] V. Lombardi, M. Bozzi and L. Perregrini, "A novel variational meshless method with radial basis functions for waveguide eigenvalue problems," IEEE Transactions on Microwave Theory and Techniques, vol. 66, no. 8, pp. 3714--3723, 2018.

[10] H. J. Eom, Electromagnetic wave theory for boundary-value problems, Springer, 2004.

[11] D. Svozil, V. Kvasnicka and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," Chemometrics and intelligent laboratory systems, vol. 39, no. 1, pp. 43--62, 1997.

[12] A. G. Baydin, B. A. Pearlmutter, A. A. Radul and J. M. Siskind, "Automatic differentiation in machine learning: a survey," The Journal of Machine Learning Research, vol. 18, no. 1, pp. 5595--5637, 2017.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein and L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," in Advances in neural information processing systems, Vancouver, BC, Canada, 2019, pp. 8026--8037.

[15] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," arXiv preprint arXiv:1711.00165, 2017.