Towards simulation of radio-frequency component with physics informed neural networks

Mathieu Riou, Mouhamed Coulibaly, Jean-Pierre Marcy

Towards simulation of radio-frequency component with physics informed neural networks

Mathieu Riou
SINCLAIR Lab
Thales Research and Technology
Palaiseau, France
mathieu.riou@thalesgroup.com

Mouhamed Coulibaly
SINCLAIR Lab
Thales Land and Air Systems
Limours, France
mouhamed.coulibaly@ensta-paris.fr

Jean-Pierre Marcy
Thales Land and Air Systems
Limours, France
jean-pierre.marcy@thalesgroup.com

Abstract—**Radio-frequency components such as transmission lines are present in many defense systems such as radars, communication systems or electronic warfare systems. The conceptions of radio-frequency components involves a simulation phase that generally consists in the resolution of harmonic Maxwell's equations. Presently, these simulations are performed with mesh based methods such as finite element or finite difference methods. The results of a simulation is specific to a given geometry and mesh and can not be reemployed for a similar problem. The recently introduced physics informed neural networks (PINNs) could offer a mesh-free alternative to these methods. While PINNs for fluid mechanics has been tested multiple times, there are only few examples of PINNs for harmonic Maxwell equations in two dimension. In this paper we present a first attempt of training a PINN for the three dimensions simulation of a transmission line and we highlight the specific difficulties of training a PINN for radio-frequency simulations.**

Index Terms—**deep learning, physics, physics informed neural network, radio frequency, electromagnetism**

I. Introduction

Radio-frequency components such as transmission lines are present in many defense systems such as radars, communication systems or electronic warfare systems. The design process of these components integrates a simulation phase. Simulations are performed with mesh based methods such as finite element or finite difference methods that involve a discretization of the space. In particular the computation of transmission property of these component may involve to compute the electric field for various boundary conditions. This computation requires to solve the harmonic Maxwell's equation. Such computation may be slow especially for higher frequency where the characteristic size of the mesh elements should be decreased. Moreover a complete characterization of component requires multiple simulations for instance varying the frequency of the field for analysing the frequency-dependent transmission and the shape of the component for sensibility analysis.

Physics informed neural network (PINN) [13] are an alternative to mesh based method to speed up simulations of filter. Firstly such methods are mesh free, and thus the tedious task of defining a mesh is avoided. Secondly they allow for leveraging all advantages of neural networks such as transfer learning which speed up the computation for similar problems, for instance similar geometries or similar frequencies. PINNs consists in integrating physical knowledge during the training step of the neural network. Such methods have been deployed for direct and inverse problem [13], [2]. PINN have been tested in multiple physics mostly in fluid mechanics [1], but also in solid mechanics, thermodynamics, chemistry and nano-optics [11], [3]. However, the use of PINN in electromagnetic is still at its infancy with examples for meta-material design [7],[10]. However these two examples present simple geometry only in two dimensions such as square domain and problems with Dirichlet boundary conditions where the field value is known in every boundaries. The design of radio frequency component for real world systems requires to model complex three dimensional geometry with non-Dirichlet boundary conditions. This study tests the implementation of PINN for the simulation of a resonant transmission line with a stub made in tri-plate air technology. Such transmission lines are used for high power radio-frequency circuits such as the one in long-range air defence radars. The simulation is in three dimension with non-Dirichlet boundary conditions and allows for challenging the PINN. The contributions of this work are:

- The implementation of a PINN for solving the harmonic Maxwell's equations on a transmission line.
- The highlight of the specific difficulties linked to training a PINN on non-Dirichlet Boundary conditions
- The test of an adaptive training strategy for balancing the different boundary conditions.

II. Principle of physics informed neural network

Also there were preliminary works [6], the seminal papers for the PINN were published by Raissi *et al* first in two paper in 2017 and than in 2019 [13]. The approach attracted much interest due its genericity (it can adapt to any problem described by a partial differential equation) and its relatively easy implementation as it leverages standards machine learning libraries to perform autoderivation for computing the residual that is at center of the approach. Considering a problem governed by an partial differential equation (PDE):

$$\mathcal{N}_x(u) = 0 \quad in \quad \Omega$$

Where $\mathcal{N}_x$ is a diferential operator and u is the solution and Ω is the domain where the solution is searched. The problem is also constrained by a set of boundary conditions:

$$f_b^i(u) = 0 \quad on \quad \partial\Omega_i \quad for \quad 1 \leq i \leq n_b$$

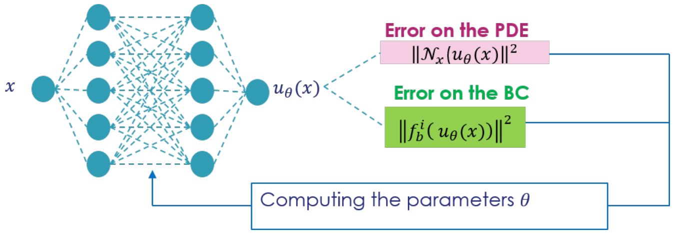
x
$u_\theta(x)$
Error on the PDE
$\|\mathcal{N}_x(u_\theta(x))\|^2$
Error on the BC
$\|f_b^i(u_\theta(x))\|^2$
Computing the parameters θ

Fig. 1. Principle of a PINN

$\partial\Omega_i$ is a boundary of Ω and n_b is the number of different boundary conditions. The idea behind PINN is to use a neural network $u_\theta(x)$ to approximate the solution of the problem $u(x)$. The motivation for using a neural network as solution basis comes from the universal approximator property of the feedforward neural networks [9], [4]. A neural network with a unic hidden layer and finite number of neurons can approximate arbitrary closely any continuous function. The neural networks parameters θ should than be trained in order to approximate the solution. The second characteristic of PINN is to embed the physics bias in the cost function used to learn the parameters of the network. The Figure 2 illustrates the working principle of a PINN.

The parameters θ of the neural network are optimized such that the error terms on the PDE, on the boundary conditions (BC) and on potential known data are minimized. The architecture of PINN are generally feed forward neural networks, and the automatic derivation methods are used to compute the loss term associated to the PDE. The computation of the losses involve also a sampling step, where points in the domain for the evaluation of the PDE error and boundary condition error are chosen. PINNs with only the PDE loss and the boundary conditions loss can solve direct problems [13]. With the addition of observation data but with unknowns in the PDE, PINNs can solve inverse problems or retrieve hidden physics [2], [14].

III. Usecase definition

This work study a relatively simple transmission line with a stub. This stub gives rise to a resonant behavior. The problem formulation is illustrated in Figure 2. The electromagnetic field should respect a PDE (the harmonic Maxwell's equations) and 3 boundary conditions on different domains detailed later. For this problem, the domain can be thus divided in 4 areas. For each domain one of the four physical constraints must be respected. In Figure 2 for each area of the transmission line, we represent the condition respected by the electrical field. Simulations will be carried out for a frequency of 4GHz.

In the airbox the electric field $\vec{E}$ has to verify the Maxwell equation in harmonic regime (highlighted in yellow in Figure 2):

$$\boldsymbol{\nabla} \times \left(\frac{1}{\mu_r}\boldsymbol{\nabla} \times \overrightarrow{E}\right) - k_0^2\epsilon_r\overrightarrow{E} = \vec{0}$$

Were $\vec{E}$ is the electric field, μ_r is the relative permeability of the medium, k_0 is the wave number in vacuum, ϵ_r is the relative permittivity of the medium. In our case we consider that the electric field is in vacuum so $\mu_r = 1$ and $\epsilon_r = 1$. The dependency to the frequency of the signal is embedded in the wave number k_0:

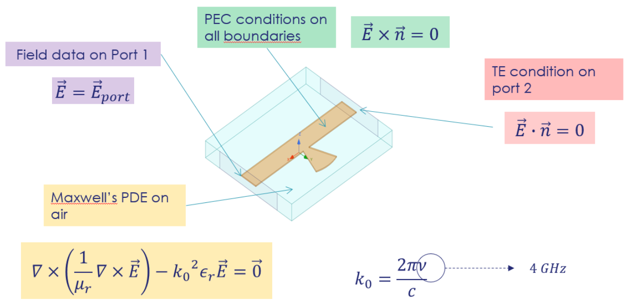
Field data on Port 1
$\vec{E} = \vec{E}_{port}$
PEC conditions on all boundaries
$\vec{E} \times \vec{n} = 0$
TE condition on port 2
$\vec{E} \cdot \vec{n} = 0$
Maxwell's PDE on air
$\nabla \times \left(\frac{1}{\mu_r}\nabla \times \vec{E}\right) - {k_0}^2\epsilon_r\vec{E} = \vec{0}$
$k_0 = \frac{2\pi\nu}{c}$
4 GHz

Fig. 2. Physics of the problem

$$k_0 = \frac{2\pi\nu}{c}$$

With ν the field frequency and c the speed of light. On both boundaries of the airbox and on transmission lines the electric field has to respect the perfect electric conductor (PEC) condition which implies that the transverse terms of the electric field should be equal to zero at the surface:

$$\overrightarrow{E} \times \overrightarrow{n} = \vec{0}$$

On the exit port, transverse electric field (TE) conditions is imposed, which means that the normal component of the electrical field is null.

$$\overrightarrow{E}.\overrightarrow{n} = 0$$

The entry port the electric field is known:

$$\overrightarrow{E} = \overrightarrow{E}_{port}$$

IV. PINN with a non-adaptive training

A. Neural network architecture

The physics informed neural network should take as input the variables of the field and should return as an output the field of interest. In our case the field is the electrical field $\vec{E}$ which has three components (E_x, E_y, E_z). $\vec{E}$ is a function of the coordinates (x, y, z). The input of the neural network is thus the coordinates (x, y, z) and the output is a prediction of $\vec{E}$ that are referred as $\hat{E} = (\hat{E}_x, \hat{E}_y, \hat{E}_z)$. Relying on what have been done in the literature a fully Connected neural network is used to train our model (Figure 3). For convenience only four layers of neurons are shown in Figure 3 including an input layer for the coordinates (x, y, z), an output layer for the prediction $(\hat{E}_x, \hat{E}_y, \hat{E}_z)$ and two hidden layers. The neurons for hidden layers are differentiable non-linear functions called activation functions. The choice of a neural network architecture consist in choosing the number of layers, the type of layers (dense or for example convolution etc...), the number of neurons per layer and the type of activation function

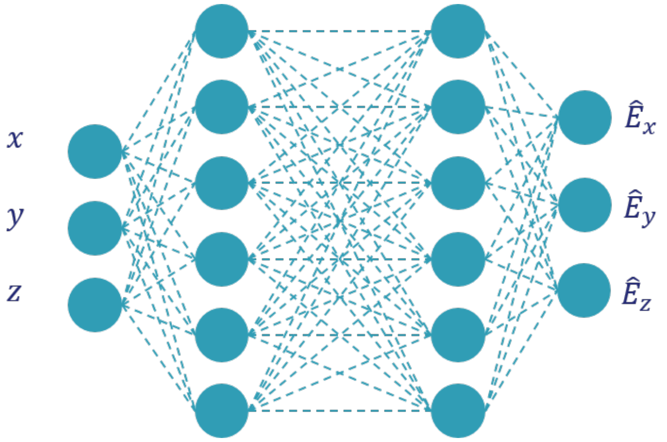

Fig. 3. Fully connected neural network architecture for the prediction of electrical field

(for example hyperbolic tangent). For the rest of our study, a neural network 10 layers of 50 neurons with hyperbolic tangent activation function will be used. The fact the neural network takes $\vec{x}$ as input allows for computing $\nabla \times \nabla \times \hat{E}$ by derivating twice the output of the neural network with regards with its input. This derivation is performed with the automatic differentiation libraries that are also used for the training of neural networks.

B. Loss terms

To adapt PINN method in our context the different terms of physics knowledge should be integrated during the learning process. Information on physics will be taken into account by computing the residual on each physics constraint (see Figure 2). Four different losses enforce the physical conditions described in the previous section on the four different model *ie* respectively the Maxwell's differential equation in the airbox (higlighted in orange), the source term on the entry port (highlighted in purple), the perfect electric conductor condition highlighted in green and the transverse electric field condition highlighted in red. $\mathcal{L}_{PDE}$ is the loss measuring the error of the neural network with regard to the Maxwell's equation in the airbox. This loss correspond to the norm of the residual that the neural network makes on Maxwell's equation.

$$\mathcal{L}_{PDE} = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{\nabla} \times \left(\frac{1}{\mu_r}\boldsymbol{\nabla} \times \hat{E}(\vec{x_i})\right) - k_0^2\epsilon_r\hat{E}(\vec{x_i})\|^2$$

Here $\hat{E}$ is the electric field predicted by the neural networks. $\vec{x_i}$ design coordinates vectors of points sampled inside the airbox. n is the number of points chosen in the airbox to evaluate this loss.

$\mathcal{L}_{source}$ is the loss measuring the respect of the source term at the entry port (port 1). This loss is simply the quadratic error between the prediction of the neural network and the real electric field.

$$\mathcal{L}_{source} = \frac{1}{m}\sum_{k=1}^{m} \|\hat{E}(\vec{x_l}) - \vec{E}(\vec{x_l})\|^2$$

$\vec{E}$ is the real electric field. $\vec{x_l}$ are coordinate vectors of point on the port 1. m is the number of point chosen in on the port 1.

$\mathcal{L}_{PEC}$ is the loss ensuring that the perfect electrical condition is enforced on the metallic boundaries (boundaries of the airbox excepted to the port and boundaries of the transmission lines).

$$\mathcal{L}_{PEC} = \frac{1}{p}\sum_{k=1}^{p} \|\hat{E}(x_k) \times \vec{n}(\vec{x_k})\|^2$$

The loss is computed as the norm of the vector product between the electric field predicted by the neural network $\hat{E}$ and the normal vector $\vec{n}$. This norm is evaluated on m points whith coordinates $\vec{x_l}$ belonging to the metallic boundaries domain.

Finally, $\mathcal{L}_{TE}$ is the loss ensuring that the transverse electrical field condition is respected by the predicted field.

$$\mathcal{L}_{TE} = \frac{1}{q}\sum_{h=1}^{q} \|\hat{E}(x_h).\vec{n}(x_h)\|^2$$

It is expressed as the norm of the scalar product between the predicted field $\hat{E}$ and the normal vector $\vec{n}$ and is evaluated on q points belonging to the exit port (port 2) of coordinate $\vec{x_h}$. At this point it is worth noticing that the knowledge of the filter geometry is crucial to sample the point $\vec{x_i}$, $\vec{x_l}$, $\vec{x_k}$ and $\vec{x_h}$. The constraints of the geometry is embedded in the different loss terms because of their dependency to the choice of the points $\vec{x_i}$, $\vec{x_l}$, $\vec{x_k}$ and $\vec{x_h}$ also referred as collocation points. In order to embed the physics of the problem, the parameters should be trained such that the four different loss terms $\mathcal{L}_{PDE}$, $\mathcal{L}_{source}$, $\mathcal{L}_{PEC}$ and $\mathcal{L}_{TE}$ are minimized.

The most naive way to aggregate the different loss terms in a single loss function is to sum them:

$$\mathcal{L} = \mathcal{L}_{PDE} + \mathcal{L}_{PEC} + \mathcal{L}_{TE} + \mathcal{L}_{source}$$

The following section will present the results of the training choosing this loss.

C. Training and results

The training is performed on 30 000 epochs for the fully connected neural network with only $tanh$ activation functions. The batch size for each error terms is 500 for about 40 000 collocation points. In order to select the collocation points, the geometry is first divided in canonical geometrical entities (here portion of cylinders and rectangular parallelepiped). Point are then sampled randomly in these canonical elements using random sampling in euclidean coordinates (x, y, z) for rectangular parallelepiped and random sampling in cylindrical coordinates (r, θ, z) for portion of cylinder. The Figure 4 shows result of the predicted field on the entry port (lower part of the Figure) compared with expected one (upper part of the Figure) for the E_x (left), E_y (middle) and E_z (right). The PINN struggles to fit data from entry port (Port1) as thee neural network converges to the null function.

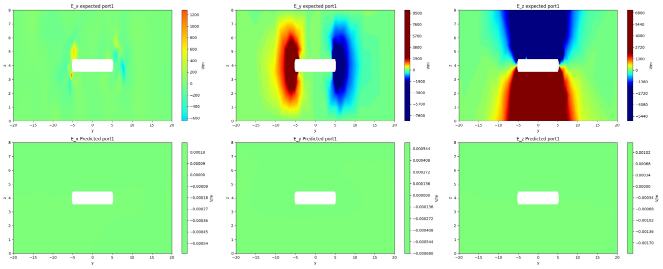

Fig. 4. Electric field on the entry port: expectations(upper part) versus predictions (lower parts) of the neural network after training with gradient sum for respectively (from left to right) E_x, E_y and E_z

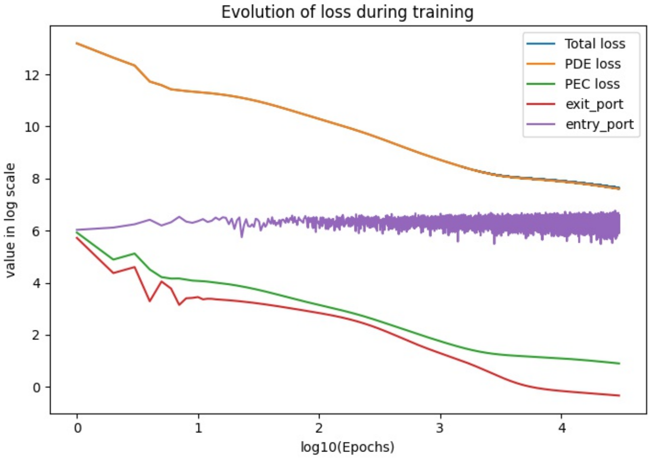

Fig. 5. Loss evolution in log scale for training with gradient sum

By analyzing the training process (Figure 5) we can see that terms of the loss have different range of values. The optimizer will choose the easiest way to reduce the value of the loss even if it means that one term of the loss will be overlooked. In our case the data term $\mathcal{L}_{source}$ (purple) is overlooked which led to the convergence to the null function.

Indeed one can see that the optimization is driven by the loss of the PDE $\mathcal{L}_{PDE}$ in orange and the field converge on a trivial but nonphysical solution of the problem because the boundary condition at the entry port is not respected. Predicting a null field is the most trivial way to respect the Maxwell equations, explaining that the neural network converge towards this solution. A null field also respect the PEC condition and the TE condition, which explains that the decrease of the loss curves for $\mathcal{L}_{PEC}$ in green and $\mathcal{L}_{TE}$ in red is correlated to the decrease of $\mathcal{L}_{PDE}$.

To address this issue, balancing the terms of the loss during the training will be considered in the next section. The choice of those coefficients to balance terms in the loss rely on multi-objective optimization theory.

V. PINN with adaptive training strategy

A. *adaptive training algorithm*

A first attempt was made by using the the learning rate annealing method for PINN which is an heuristic algorithm presented by [16]. It is designed to tackle multi-tasks optimization by balancing each gradients so that their contribution will be equal. The motivation under a renormalization of the gradients instead of a renormalization of the loss terms comes from the training procedure of the neural networks. Let $\mathcal{L}$ be the loss for training expressed as a linear combination of the loss terms:

$$\mathcal{L} = \mathcal{L}_{PDE} + \sum_{i=1}^{M} \lambda_i \mathcal{L}_i$$

where $\mathcal{L}_{PDE}$ is the loss of the PDE and $\mathcal{L}_i$ are respectively the different boundary loss terms $\mathcal{L}_{source}$, $\mathcal{L}_{PEC}$ and $\mathcal{L}_{TE}$. The parameters are updated through gradient descent algorithm:

$$\theta_{n+1} = \theta_n - \nabla_\theta \mathcal{L}_{PDE} - \sum_{i=1}^{M} \lambda_i \nabla_\theta \mathcal{L}_i$$

Thus having an equal contribution of each loss term for the parameters update requires for each gradient term to have a similar scale. The learning rate annealing algorithm consists in weighting by using gradient statistics to appropriately balance all terms.
The algorithm is the following:

Algorithm 1 Learning rate annealing for physics-informed neural networks

Consider a physics-informed neural network $f_\theta(\boldsymbol{x})$ with parameters θ and a

$$\mathcal{L}(\theta) := \mathcal{L}_{PDE}(\theta) + \sum_{i=1}^{M} \lambda_i \mathcal{L}_i(\theta)$$

where $\mathcal{L}_{PDE}(\theta)$ denotes the PDE residual loss, the $\mathcal{L}_i(\theta)$ correspond to data-fit terms (e.g., measurements, initial or boundary conditions, etc.), and $\lambda_i = 1, i = 1, \ldots, M$ are free parameters used to balance the interplay between the different loss terms. Then use S steps of a gradient descent algorithm to update the parameters θ as: $n = 1, \ldots, S$ Compute $\hat{\lambda}_i$ by

$$\hat{\lambda}_i = \frac{\max_\theta \left\{ \left| \nabla_\theta \mathcal{L}_{PDE}\left(\theta_n\right) \right| \right\}}{\overline{\left| \nabla_\theta \lambda_i \mathcal{L}_i\left(\theta_n\right) \right|}}, \quad i = 1, \ldots, M,$$

where $\overline{\left| \nabla_\theta \lambda_i \mathcal{L}_i\left(\theta_n\right) \right|}$ denotes the mean of $\left| \nabla_\theta \lambda_i \mathcal{L}_i\left(\theta_n\right) \right|$ with respect to parameters θ. Update the weights λ_i using a moving average of the form

$$\lambda_i = (1 - \alpha)\lambda_i + \alpha \hat{\lambda}_i, \quad i = 1, \ldots, M.$$

Update the parameters θ via gradient descent

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta \mathcal{L}_{PDE}\left(\theta_n\right) - \eta \sum_{i=1}^{M} \lambda_i \nabla_\theta \mathcal{L}_i\left(\theta_n\right)$$

The values used are $\eta = 10^{-3}$ and $\alpha = 0.7$.

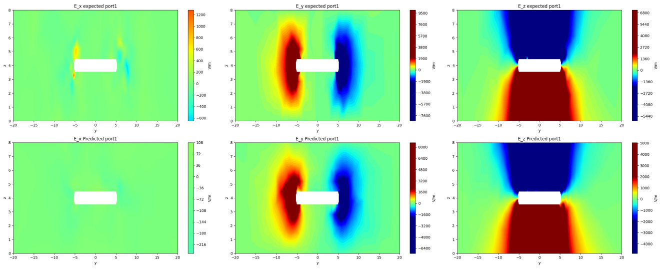

Fig. 6. Electric field on the entry port: expectations(upper part) versus predictions (lower parts) of the neural network after training with learning rate annealing for respectively (from left to right) E_x, E_y and E_z

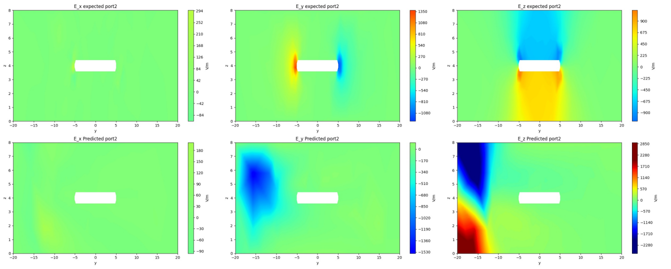

Fig. 7. Electric field on the output port: expectations(upper part) versus predictions (lower parts) of the neural network after training with learning rate annealing for respectively (from left to right) E_x, E_y and E_z

The $\hat{\lambda}_i$ is designed such that each gradient have the same scale as the one for the PDE. λ_i is only partially updated with value $\hat{\lambda}_i$ using the hyperparameter α. The partial update prevents the training from being unstable.

B. Training and results

Using the adaptive trainin procedure, better performances of PINN were noticed on the entry port (Port 1 see Figure 6). the neural net prediction (Figure 6 down) is able to reproduce accurately the reference electric field imposed on the entry port 1 Figure 6 up).

However even by using learning rate annealing method the neural net still struggles to learn accurately the electric field inside the airbox. In Figure 7 electric Field at the exit port shifted to the side. However, one can notice that the TE condition is well respected as the x component of the electric field is approximately zero. $\hat{E}_x$ on the left lower part of the Figure 7 is in good agreement with the reference field E_x on left upper part of the Figure 7. Figure 8 displays the loss curves for the learning of the four different loss terms. When compared to the first attempt with non-adaptive control, the condition at the entry port is learned and decrease especially starting from 90-100 epochs (Figure 8 purple line). The loss curve is also very noisy, showing the difficulty to learn this condition. The three other loss terms have a non monotonic evolution especially with an increase of their value also in 90-100 epochs. The value of these losses are also higher, especially the loss for the partial differential equation in orange. Overall it is much harder to train the neural network in such conditions, also it offers the possibility to converge toward a non-trivial zero field. The Figure 9 shows the evolution of the λ_i values respectively for PEC (green), exit port (red) and entry port (purple) error gradients. One may notice the great difference in order of magnitude for the different values. The exit port has the most important values in the order of 10^8 to 10^{11}. Lambda value for entry port and PEC is of the order of 10^7 to 10^9. The lower the value of the loss term, the greater the value of the lambda. The noisiness of the loss evolution seems to correspond to the one of lambda values. As values of lambda coefficients varies in several order of magnitude during training, it highlights that it is not possible to find a fix value of coefficient to balance the gradients during training.

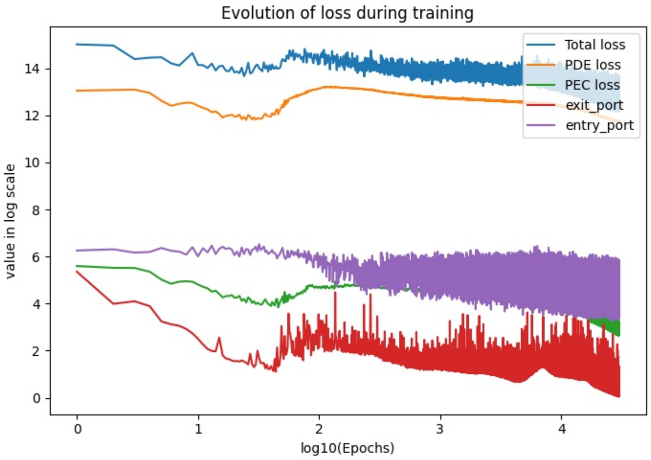

Fig. 8. Loss evolution in log scale for training with learning rate annealing method

VI. Discussion

From these results we can assert that the PINN is sensitive to the optimization method. Finding an efficient multi-objective optimization algorithm will speed up the convergence and provide us a solution close to the physical problem.
To understand the problem we may focus on the training process. During training process (Figure 8), while the loss related to the entry port decreases the loss related to the PDE increases, meaning that the two terms might be conflicting and the loss related to entry port prevails over the loss related to the PDE. Training PINN is known to be difficult and the use case of the transmission line is particularly challenging as they are three different boundary conditions including two where the value of the electric field is only partially known. A difficulty of training the PINN for the transmission line could reside also in the relatively high derivative order of the partial differential equation and the depth of the neural net. The PINN use backpropagation of the gradients from the output to the input neural network to compute the derivative of the partial differential equation. For the second order derivative, the gradient should be back-propagated twice in the

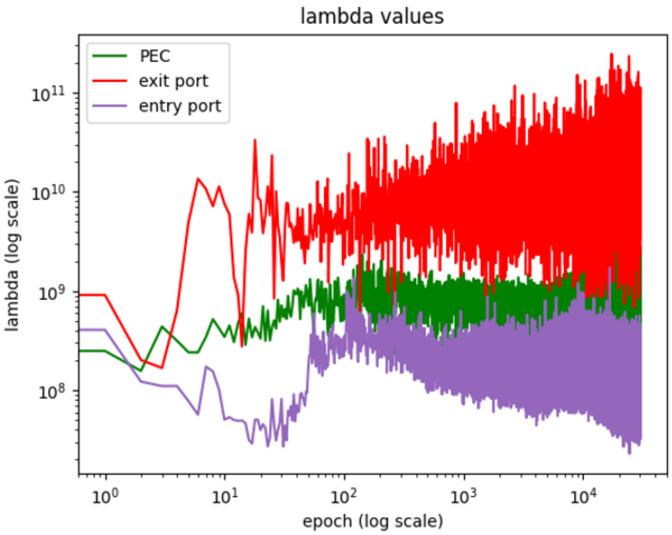

Fig. 9. Lambda evolution in log scale for training with learning rate annealing method

neural network architecture increasing the risk of vanishing or exploding gradient. While the formulation of the neural network parameters optimisation process have been tested in this work, notably using the learning rate annealing gradient algorithm, the selection of the training examples also called collocation points and the architecture of the neural network have been overlooked. Potential future research direction may involve to test residual based adaptive refinements methods for the choice of collocation points [12], [17]. Some works also propose to integrate the boundary conditions as a hard constraint in the neural network architecture [15], suppressing the multi-task optimisation problem for the PINN. Also, the proposition only include Dirichlet boundary conditions. Such method should be adapted if one seek to integrate perfect electrical conductor or transverse electric field conditions in the neural network architecture. An other method proposed to use generative adversial architecture with physics based error term [5]. In such approach the generator aims to generate the solution to a PDE by fooling the discriminator. The PDE constrained is treated as a discriminator input through a physics consistency score, while the boundary condition are treated as examples labelled ”true" for the discriminator training. Such approach does not require a balance between a PDE error term and a boundary condition error terms. However as the approach [15], it is well suited for problem with only Dirichlet boundary conditions and can not be applied directly to our use case where there three different kinds of boundary conditions. Concerning the comparison with traditional simulation method, the training of a PINN on the aforementioned use-case is much longer than solving it with standard finite element method, as training the PINN requires several hours as the finite element computation takes only few tens of minutes. The present problem is rather simple for finite element method as it requires only about 3000 elements in the mesh to solve the problem. As highlighted in [8], PINNs approach tends to be slower than finite element methods on simple problem, while they may be more efficient on more complex geometries. The interest of this relatively simple problem is to select the appropriate method for multi-objective optimization before to move to more complex geometries.

VII. Conclusion

In this paper we have presented a first attempt to train a PINN for the simulation of the electric field surronding a transmission line. The field is obtained solving the harmonic Maxwell equations while respecting three different boundary conditions including two non-Dirichlet boundary condition. Training the PINN for transmission lines is a complex multi-task optimisation problem due to the multiplicity of the boundary conditions. A naive approach of the training consist in optimising the neural network parameter using the sum of the four loss terms gradient. In such case, the neural network predict a trivial electric field that is null every where. While this solution is very different from the expected field, it respect three physical constrains on four, namely the partial differential equation, the perfect electrical conductor, the transverse electric field condition. On the other hand the electric field that is impose at the entry port of the transmission line is not respected as the gradient of this error term barely participates to the parameter update. In order to take into account the field at the entry port, we tested an adaptive training strategy that balance the gradients of the different error terms such that they contribute equally to the parameter update. In such case the field at the entry port is well predicted but decreasing the loss on the partial differential equation is much harder. Moreover conflicts between loss terms appears as the decrease of the loss term on the entry port is correlated with an increase of the loss on the partial differential equation. In conclusion, while PINN are promising for solving differential equations, there application to the simulation of transmission line would require to solve issues on the multi-task optimization that is involved. Overcoming this difficulty would allow in the future to fully leverage the advantages of neural networks in the field of simulation such as the generated mesh free solutions to simulation problem with the possibility to transfer knowledge to similar tasks. Design of radio-frequency components for instance for long-range air defence radars could benefit from this method.

VIII. Acknowledgement

The authors thank Jean-Yves Morineau for the simulation of reference data for the electric field and his expertise on problem formulation.

References

[1] Shengze Cai et al. “Physics-informed neural networks (PINNs) for fluid mechanics: A review”. In: *Acta Mechanica Sinica* 37.12 (2021), pp. 1727–1738.

[2] Yuyao Chen et al. “Physics-informed neural networks for inverse problems in nano-optics and metamaterials”. In: *Optics express* 28.8 (2020), pp. 11618–11633.

[3] Salvatore Cuomo et al. “Scientific machine learning through physics–informed neural networks: where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88.

[4] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[5] Arka Daw, M Maruf, and Anuj Karpatne. “PID-GAN: A GAN Framework based on a Physics-informed Discriminator for Uncertainty Quantification with Physics”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 237–247.

[6] MWMG Dissanayake and Nhan Phan-Thien. “Neural-network-based approximations for solving partial differential equations”. In: *communications in Numerical Methods in Engineering* 10.3 (1994), pp. 195–201.

[7] Zhiwei Fang and Justin Zhan. “Deep physical informed neural networks for metamaterial design”. In: *IEEE Access* 8 (2019), pp. 24506–24513.

[8] Tamara G Grossmann et al. “Can physics-informed neural networks beat the finite element method?” In: *arXiv preprint arXiv:2302.04107* (2023).

[9] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.

[10] Xiang Huang et al. “Solving partial differential equations with point source based on physics-informed neural networks”. In: *arXiv preprint arXiv:2111.01394* (2021).

[11] George Em Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.

[12] Lu Lu et al. “DeepXDE: A deep learning library for solving differential equations”. In: *SIAM review* 63.1 (2021), pp. 208–228.

[13] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[14] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations”. In: *Science* 367.6481 (2020), pp. 1026–1030.

[15] Luning Sun et al. “Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data”. In: *Computer Methods in Applied Mechanics and Engineering* 361 (2020), p. 112732.

[16] Sifan Wang, Yujun Teng, and Paris Perdikaris. “Understanding and mitigating gradient flow pathologies in physics-informed neural networks”. In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3055–A3081.

[17] Chenxi Wu et al. “A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 403 (2023), p. 115671.