

Weak Formulation for Physics-Informed Neural Networks in the Resolution of Analysis Problems in Electromagnetics

S. Barmada¹, S. Dodge¹, A. Formisano²

¹DESTEC, University of Pisa, Pisa, Italy

²Dept. of Engineering, University of Campania “Luigi Vanvitelli”, Aversa, Italy

A new class of neural networks has been recently introduced to solve problems based on some physical model. Such networks are called “Physics-Informed Neural Networks” (PINN). PINN are trained not based on input-output data, but rather by explicitly enforcing the model physical laws. More in detail, they are trained to minimize the “equations residual” in the physical model in each point of the domain rather than the discrepancy with known data. In this contribution, we propose to formulate the equations of ElectroMagnetism (EM) embedded in a PINN by using a weak formulation approach. This helps convergence of the training process, thanks to the lower derivation order required in the error computation. In addition, the presence of different materials in the domain is easily treated. In this digest, the approach is described in its fundamentals, and a simple example is presented.

Index Terms—Physics-Informed Neural Networks, Electromagnetic problems, Weak formulations.

I. INTRODUCTION

IN RECENT YEARS the application of Neural Networks (NNs) has become increasingly prevalent in a variety of fields, particularly for modeling complex physical systems [1, 2, 3]. In many applications, however, gathering sufficient training data may reveal an overwhelming task. A particularly exciting recent development in the field of machine learning is the concept of Physics-Informed Neural Networks (PINNs). Unlike conventional NN that rely solely on data for training, PINNs incorporate *a-priori knowledge* of the governing physical laws into the training process [4]. The goal is to ensure that the trained NN not only fits the data but also satisfies the physical constraints imposed by the model governing equations. In the case of PINNs, the “error” function is defined as the sum of two components: the first is the data-based error (typically a mean squared error between the NN’s output and the true observed values, for example on the domain frontier, where boundary conditions are assigned), and the second is the residual error that arises from the governing equations of the physical system. The residual error is computed by substituting the NN’s output into the physical equations, which are then evaluated at various points within the domain of interest. The application of the differential operator defining the physical equations is achieved thanks to the Automatic Differentiation (AD) concept, making it possible to easily derive the output of the NN [5]. Note that the presence of second order derivatives could lead to difficulties during the training phase, especially when complex geometries or material inhomogeneity are present.

While traditional PINN training techniques focus on minimizing the residual error pointwise, the method proposed here introduces a “weak-formulation” approach, where a weighting function is applied and an integral error over the domain is computed, for a full set of weighting functions. The idea is that by applying different weights to the error at each point in the domain, the network can better balance the contribution of each data point and more effectively regularize the solution. A simple example of this method is presented in the digest. A more comprehensive exposition, and multiple case studies will be provided in the full paper.

II. PHYSICS-INFORMED NEURAL NETWORKS

Let us consider a simple non-dynamical problem in EM field computation, described by Poisson equation, with suitable boundary conditions:

$$\begin{cases} \nabla^2 \varphi(\mathbf{x}) = f(\mathbf{x}) & \forall \mathbf{x} \in \Omega \\ \varphi = \varphi^* & \text{on } \partial\Omega_D \\ \partial_n \varphi = \psi^* & \text{on } \partial\Omega_N \end{cases} \quad \begin{matrix} (1a) \\ (1b) \\ (1c) \end{matrix}$$

where $\varphi(\mathbf{x})$ is an unknown (scalar) potential, Ω is the definition domain, $\partial\Omega_D$ and $\partial\Omega_N$ are the Dirichlet and Neumann parts of the boundary, respectively. Eqs. (1) describe a non-dynamical model, but the PINN approach is general enough to treat also dynamical cases. A NN model of (1) receives as input the coordinates $\mathbf{x}$ of a point within Ω and yields as output the value of the approximate solution $\tilde{\varphi}(\mathbf{x}, \mathbf{p}_{NN})$, where $\mathbf{p}_{NN}$ represents the vector of weights and biases describing the NN.

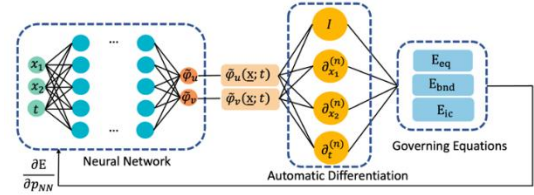


Fig. 1 General architecture of a PINN for solving a PDE. E_{ic} represents errors due to initial conditions in the case of dynamical problems. Subscripts u and v refer to partial derivatives with respect to inner representation variables.

A possible approach to consider the *a-priori* information about (1) is to define a representation error E including a term related to the “residual” of the equation and a second term related to boundary conditions:

$$E(\mathbf{p}_{NN}) = E_{bnd}(\mathbf{p}_{NN}) + E_{eq}(\mathbf{p}_{NN})$$

where:

$$E_{bnd}(\mathbf{p}_{NN}) = \|\tilde{\varphi}(\mathbf{x}, \mathbf{p}_{NN}) - \varphi^*(\mathbf{x})\|_{\mathbf{x} \in \partial\Omega_D} + \|\partial_n \tilde{\varphi}(\mathbf{x}, \mathbf{p}_{NN}) - \psi^*(\mathbf{x})\|_{\mathbf{x} \in \partial\Omega_N} \quad (2)$$

and

$$E_{eq} = \|\nabla^2 \tilde{\varphi}(\mathbf{x}, \mathbf{p}_{NN}) - f(\mathbf{x})\|_{\mathbf{x} \in \Omega} \quad (3)$$

where $\mathbf{x}$ represents the array of sampling points coordinates where loss function is evaluated, and the term E_{eq} can be computed using AD. Fig.1 shows a general view of the training process of a PINN inspired by (1)-(3).

III. WEAK FORMULATION PINN

Let us now modify equations (1) to provide a weak-form formulation. Following the weighted residuals approach [6], we multiply the left and right terms in (1a) by a weighting function $\psi(\mathbf{x})$ and integrate over Ω . By using the Green's lemma we obtain:

$$\left\{ \begin{aligned} \int_{\Omega} \nabla \tilde{\varphi} \cdot \nabla \psi d\Omega &= \int_{\partial\Omega_N} \psi^* \partial_n \tilde{\varphi} d\Sigma - \int_{\Omega} f \psi d\Omega \end{aligned} \right. \quad (4a)$$

$$\varphi = \varphi^* \text{ on } \partial\Omega_D \quad \forall \psi \in \Psi \quad (4b)$$

where ψ is a smooth weighting function, allowing to compute integrals in (4a) for any unknown potential φ . Details about the general requirements on the weighting functions set Ψ will be given in the full paper. The residual loss $E_{eq}(\mathbf{p}_{NN})$ is consequently modified substituting eq. (4a) to eq. (1a). The error is formed for each training batch by summing up the contribution in each point within Ω . Then a new batch starts by changing the weighting function, and so on, until all the weighting functions have been considered. The sequence is then started again, until convergence, by taking care that at each iteration the batches sequence is varied randomly. The Dirichlet part of the boundary is treated similarly to the standard PINN, reflecting the difference between natural and essential boundary conditions, typical of weighted residuals approaches.

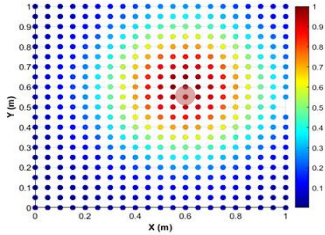


Fig. 2. A Sketch of a sample rectangular domain Ω , together with the definition domain of a sample test function. Evaluation points are indicated with dots; the gaussian center is highlighted with a red circle.

Different choices of the weighting functions are possible. In this work, we will create a partition of Ω , and consider (possibly overlapping) Gaussian functions ψ , each centered in one element of the partition, similarly to meshless approaches [7]. Fig. 2 shows a sketch of a domain, with indications of the definition domain of a sample weighting function ψ .

IV. EXAMPLE

In the numerical example, a simple 3 layers NN is considered for the sake of illustration, and the test functions are assumed as Gaussian functions, whose centers and standard deviations are pre-defined to completely cover Ω , with partial superpositions, much like in the meshless approaches.

The test case domain and numerical results are shown in Fig. 3: as in [8] and L shaped domain is considered, with Dirichlet

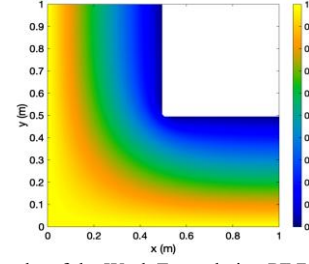


Fig. 3 Numerical results of the Weak Formulation PINN, Laplace equation.

conditions $\varphi = 1$ at $x = 0, y = 0$, $\varphi = 0$ at $x = 0.5, y = 0.5$ and Neumann conditions elsewhere; the region is source-free, so the Laplace equation is solved.

TABLE 1 COMPARISON OF THE EFFICIENCY BETWEEN REGULAR PINN AND WEAK FORMULATION PINN

	Sampling points	Epochs
Regular PINN	441	100
Weak formulation PINN	441	60

Comparing the weak formulation PINN and regular PINN approach, the authors verified that the proposed approach leads to the same accuracy (compared to a FEM model implemented on a commercial code) in a lower CPU time consumption, due to the reduction of complexity in evaluating the derivatives. Additionally, the weak formulation PINN integrates the equation over the domain, averaging out localized errors and minimizing the need for deeper networks. As a result, faster convergence with a simpler architecture and simplified numerical computation are key advantages of this approach, leading to significant savings in computational resources and time. In the full version, the impact of the choice of representation and weighting functions will be addressed.

ACKNOWLEDGEMENT

This work is partly funded by Italian Ministry for Education, University and Research under the project PRIN 2022 grant number 2022CL7HM, and the European Union – Next Generation EU.

REFERENCES

- [1] H. Sato and H. Igarashi, "Fast Multi-Objective Optimization of Electromagnetic Devices Using Adaptive Neural Network Surrogate Model," in *IEEE Transactions on Magnetics*, vol. 58, no. 5, pp. 1-9, May 2022.
- [2] A. Kovacs et al., "Magnetostatic and micro magnetism with PINNs," *Magnetostatics and micromagnetics with physics informed neural networks*, Journal of Magnetism and Magnetic Materials, Volume 548, 2022.
- [3] S. Barmada, P. Di Barba, A. Formisano, M.E. Mognaschi, M. Tucci, "Learning-Based Approaches to Current Identification from Magnetic Sensors", *Sensors*. 2023; 23(8):3832
- [4] A. Khan and D. A. Lowther, "Physics Informed Neural Networks for Electromagnetic Analysis," *IEEE Transactions on Magnetics*, 2022.
- [5] A. G. Baydin et al., "Automatic Differentiation in Machine Learning: a Survey", *J. of Machine Learning Res.*, vol. 18(153), pp. 1-43, 2018.
- [6] R. F. Harrington, "Field Computation by Moment Methods," Macmillan, New York, 1968.
- [7] S. L. Ho, S. Yang, J. M. Machado and H. C. Wong, "Application of a meshless method in electromagnetics," in *IEEE Transactions on Magnetics*, vol. 37, no. 5, pp. 3198-3202, Sept. 2001.
- [8] S. Barmada, S. Dodge, M. Tucci, A. Formisano, P. Di Barba and M. Evelina Mognaschi, "A Novel Hybrid Boundary Element—Physics Informed Neural Network Method for Numerical Solutions in Electromagnetics," in *IEEE Access*, vol. 12, pp. 171444-171457, 2024.