
Temperature Estimation with Bio-Heat Equation using Physics-Informed Neural Networks

Students: Alessio Borgi, Eugenio Bugli, Alessandro De Luca
Department of Computer, Control and Management Engineering
Supervisors: G. Cappellini, M. Vendittelli



Abstract—This paper explores the estimation of temperature distribution using the Bio-Heat equation through the application of Physics-Informed Neural Networks (PINNs). The focus is on hyperthermia treatments, particularly in controlling the thermal energy applied to cutaneous and subcutaneous tissues. By utilizing PINNs, this research aims to simplify the resolution of the Pennes Bio-Heat equation, a partial differential equation (PDE), and improve the real-time estimation of temperature in 1D and 2D spatial domains. The simulations employ various neural network configurations, including different activation functions and optimization algorithms, such as Adam and L-BFGS, to enhance the convergence and accuracy of the model. The results show that PINNs offer a computationally efficient alternative for temperature prediction in hyperthermia therapy, with promising performance in handling time-dependent PDEs.

Index Terms—Bio-Heat equation, Physics-Informed Neural Networks (PINNs), temperature estimation, hyperthermia treatment, Pennes equation, deep learning, neural network optimization, thermal therapy, 1D and 2D heat diffusion, medical applications of PINNs.

I. INTRODUCTION

Thermotherapy treats diseases by altering the temperature of the skin, and a promising area of research and development is focused on skin tumours. These tumours can be weakened or even eradicated by this treatment, this can depend on the type of tumour and the type of thermotherapy applied, for example the main purpose of thermalablation is to overheat a certain area to destroy the tumour itself, with the risk of damaging the skin on the other hand. The other applications are: cryoablation, where the tumour is destroyed by applying a very low temperature, and finally hyperthermia, where heating weakens the tumour cells.

To control the thermal energy in both cutaneous and subcutaneous tissues, an antenna is manually placed on the skin and held in position by a passive robotic arm. A bolus of water is positioned between the subject and the antenna to prevent skin burns. The goal is to vary the skin's temperature according to a specific profile to weaken the tumor and enhance the effectiveness of other treatments such as chemotherapy, radiotherapy, and immunotherapy. The antenna aims to modulate the temperature between 39-45°C for approximately 50-60 minutes while the other treatments are being applied. This approach leverages the benefits of combined therapies to improve treatment outcomes for patients with skin tumors.

The objective of this research is to simplify the resolution of the Pennes equation using a PINN. The approach taken is one

of interpolation, whereby the equation is operated to create a dataset of input and output values. This is then used to train the neural network (NN) to understand the relationship between the values. While the NN may not be able to accurately reproduce the relationship between the values, this is not the objective. The NN is trained to calculate the results faster than would be possible with the actual formula, which is not provided to the NN. This is necessary because these kinds of threats are real-time operations, so waiting for the machine to generate the output is not optimal. The aim is to reproduce the Pennes equation in two dimensions (2D) by first recalculating a previous version of the one-dimensional (1D) [2] situation and then modifying the formula to expand the domain of interest. Consequently, the PINN will be able to control the overcited tissue and generate a temperature adequate for heating a 2D area. One significant challenge in this context is the control system. Given that the active operation of heating is non-invasive, it follows that the system control should also be non-invasive. In addition, the control technique should possess a number of desirable properties [6], including the ability to measure temperature at depth with high accuracy, ease of practical use, minimal bulk, and cost-effectiveness. Hyperthermia alone has been studied in the treatment of superficial tumors. While an occasional response is present, this modality is not effective as the joint use of Hyperthermia with Radiotherapy and Chemotherapy. The most common side effect is superficial or subcutaneous tissue burns, usually small in volume and generally first or second degree in character. There are also potential contraindications to Hyperthermia related to the physiologic stresses of heating, which have been compared to a moderately intense exercise workout.

II. BIOHEAT

In order to regulate and modify the temperature of the region of interest, Pennes' bio-heat equation is employed. Despite ongoing debate regarding the assumptions made by Pennes [4] [5], this remains the most straightforward and widely used representation of thermal flow phenomena. Despite three significant concerns, including discrepancies between experimental data and theoretical results, the heat transfer between capillary blood and tissue, and the potential anisotropy of the perfusion effect, Pennes' equation has prompted further investigation in several research areas. For example, he suggested that the rate of heat transfer between

blood and tissue is proportional to the product of the volumetric perfusion rate and the difference between the arterial blood temperature and the local tissue temperature [4]:

$$h_b = V \rho_b C_b (1 - \kappa) (T_a - T) \quad (1)$$

For the purposes of this study, we do not consider the water bolus that is typically used to prevent burns to the skin. The underlying formula remains valid and open to modification, with the addition of terminology such as a cooling factor that simulates the aforementioned water bolus. In the following section, we will describe the heating process as described by Pennes' equation [2].

$$\rho c \frac{\partial T}{\partial t} = k_{\text{eff}} \frac{\partial^2 T}{\partial x^2} - \rho_b c_b \omega_b (T - T_a) + Q, \quad x \in \Omega, t \in [0, t_f] \quad (2)$$

In the equation, ρ represents the density of the tissue. The specific heat capacity of the tissue is denoted by c . The term $\frac{\partial T}{\partial t}$ expresses the time variation of temperature T within the tissue. The effective thermal conductivity, k_{eff} , quantifies the tissue's ability to conduct heat. The second derivative $\frac{\partial^2 T}{\partial x^2}$ describes the diffusion of heat within the tissue due to thermal conductivity. For blood, ρ_b and c_b denote the density and specific heat capacity of the blood, respectively, and ω_b represents the perfusion rate of the blood. The temperature difference between tissue and arterial blood, $(T - T_a)$, models the heat exchange between tissue and blood due to perfusion. The term Q represents the generation of metabolic heat within the tissue and, in the case of hyperthermia treatments, the power due to the radiating electric field. This term can be divided into q_m and q_p , which represent the metabolic heat generation rate for the tissue and the energy deposition rate into the tissue from the heating source. Finally, $x \in \Omega$ indicates that the spatial variable x belongs to the Ω domain, the region of interest, while $t \in [0, t_f]$ specifies that time t is considered in the range from 0 to t_f , where t_f is the final time of interest for the analysis. The resolution of this partial differential equation (PDE) may result in numerical instability when traditional numerical methods, such as finite difference methods (FDM), finite element methods (FEM), or finite volume methods (FVM), are employed to solve the Pennes' equation. In order to mitigate numerical instability, a number of strategies can be employed. These include the use of smaller time steps, implicit methods and higher order methods, amongst others. The method chosen in this instance is that of scaling and normalisation, which employs the following definition [2]:

$$T' = T - T_a \quad \theta = \frac{T'}{T_M - T_a} X = \frac{x}{L_o} \quad (3)$$

That lead to the formulas:

$$\begin{aligned} \partial_{xx} T' &= \partial_{XX} T' (\partial_x X)^2 = \partial_{XX} T' \frac{1}{L_o^2} \\ \partial_t T' &= \partial_t \theta (T_M - T_a) \\ \partial_X T' &= \partial_\theta T' \partial_X \theta = (T_M - T_a) \partial_X \theta \\ \partial_{XX} T' &= (T_M - T_a) \partial_{XX} \theta \end{aligned} \quad (4)$$

Substituting equation (2) with (3) (4) and defining the thermal diffusivity as $\alpha = \frac{\rho c}{k_{\text{eff}}}$, the result is:

$$\partial_t \theta = \frac{1}{\alpha L_o^2} \partial_{XX} \theta - \frac{\rho_b c_b \omega_b}{\rho c} \theta + \frac{Q}{\rho c (T_M - T_a)} \quad (5)$$

Introduce a change in the time variable $\tau = \frac{t}{\tau_f}$, with $\tau_f = 1800s$, the typical time span of a treatment. Hence:

$$\partial_\tau \theta = a_1 \partial_{XX} \theta - a_2 W \theta + a_3 \quad (6)$$

where $a_1 = \frac{\tau_f}{\alpha L_o^2}$, $a_2 = \frac{\tau_f c_b}{\rho c}$, $a_3 = \frac{\tau_f Q}{\rho c (T_M - T_a)}$. The term $W = \rho_b \omega_b$ represents the volumetric perfusion rate of blood per unit volume of tissue. The values employed in the simulation are identical across all simulations and derived from the preceding study conducted by [2]:

| Parameter | Value | Parameter | Value |
|------------------|-------|------------|-----------------------|
| ρ | 1050 | T_a | 37 |
| ρ_b | 1043 | T_{max} | 45 |
| c | 3639 | Q | 0 |
| c_b | 3825 | w_b | 2.22×10^{-3} |
| k_{eff} | 5 | w_{min} | 0.43×10^{-4} |
| q_0 | 16 | w_{max} | 3.8×10^{-3} |
| β | 15 | L_o | 0.05 |
| t_f | 1800 | ΔT | 8 |

TABLE I: Values of parameters used in the model.

The Equation (6) is the one used for the One dimensional case, while for the Two dimensional case we have taken into account the presence of two spatial coordinates:

$$\partial_\tau \theta = a_1 \nabla^2 \theta - a_2 W \theta + a_3 Q \quad (7)$$

The previous equation can be written as follows:

$$\partial_\tau \theta = a_1 (\partial_{XX} \theta + \partial_{YY} \theta) - a_2 W \theta + a_3 Q \quad (8)$$

III. IMPLEMENTATION PROCEDURE

In order to resolve the previous formulas, we rely on PINN (Physics-Informed Neural Networks) and, more specifically, on a data-driven approach. The data-driven solution refers to the use of data to solve complex partial differential equations (PDEs) in situations where traditional analytical or numerical solutions are challenging to obtain. The data in question were generated by using Wolfram Mathematica environment. These are essential in creating a robust pipeline for generating and verifying the data needed to effectively train a neural network, which will be able to predict the behaviour described by the Bio-Heat Equation. Inside this Environment, We define the equation along with the initial and boundary conditions. This details are extremely important since they describe the behaviour of the model at the initial instant of time and along the boundaries of our surface. The initial condition used to implement the Mathematica Model for the One and Two dimensional case are the following:

$$\theta(x, 0) = 0 \quad (9)$$

$$\theta(x, y, 0) = 0 \quad (10)$$

Since we would like to have an Observer able to reconstruct the behaviour of the Mathematica Model, we have used different initial condition during the implementation of our Neural Bio-Heat Observer (NHBO):

$$\theta(x, 0) = q_0 \frac{X^4}{4(T_{max} - T_a)} \quad (11)$$

$$\theta(x, y, 0) = q_0 \frac{X^4}{4(T_{max} - T_a)} \quad (12)$$

In the One dimensional case, the surface of our model has only two boundary conditions, which are specified when $X = 0$ and when $X = 1$.

$$\theta(0, t) = 0 \quad (13)$$

$$\partial_x \theta(1, t) = K(t + \frac{1-t}{1+e^{-s(t-t_c)}} - \theta(1, t)) \quad (14)$$

The first one is our Left boundary bondition and it is defined as a Dirichlet boundary condition, which specifies the value of the solution at the boundary itself. The second one is our Right boundary bondition and is defined as a Robin boundary condition, since It specifies a linear combination of the function value and its normal derivative on the boundary. In the Two dimensional case, our model has a square surface, hence we have four different boundary conditions, which are specified at $X = 0$, $X = 1$, $Y = 0$ and $Y = 1$.

$$\theta(0, y, t) = 0 \quad (15)$$

$$\partial_x \theta(1, y, t) = t \quad (16)$$

$$\partial_y \theta(x, 0, t) = \partial_y \theta(x, 1, t) = 0 \quad (17)$$

As we can see, the Left boundary condition (15) is modeled as a Dirichlet, while the Right (16), Lower and Upper (17) are all Neumann's, since they specify the rate of change of the function at the boundary. In both cases, we have used this boundary conditions for our Mathematica model and for the implementation of the NHBO. The NHBO implementation has been done by using the DeepXDE library, which embed a PDE into the loss of our neural network using automatic differentiation [7]. Since we are dealing with a Time-Dependent PDE we have implemented our PiNN by using the TimePDE class from the library. The initial and boundary conditions of both One and Two dimensional case have been implemented with their specific type. All the Comparative Plots shown in our report have been implemented directly by using Weight And Biases or by using Matplotlib and Scipy libraries. The simulations use a huge variety of activation functions, each with its respective initialization, to evaluate their influence on the convergence and accuracy of the model. In addition, hyper-parameter optimization is performed using the Skopt library, which allows for iterative refinement of parameters including learning rate, number of layers and neurons per layer, and choice of optimiser (e.g. Adam, SGD). This rigorous approach allows the identification of the most effective configuration for specific simulation requirements, thereby improving the model's ability to approximate complex

thermal behaviour. The accuracy of the model in simulating the actual thermal behaviour prescribed by the Pennes equation is evaluated using key performance metrics. The performance metrics used are Mean Absolute Error (MAE), Mean Squared Error (MSE), L2 Relative Error (L2RE) and Maximum Absolute Percentage Error (MAX_APE). Taken together, these metrics provide a robust assessment of the accuracy of the model, allowing identification of areas where the model excels or needs improvement. In order to facilitate the interpretation of these metrics, a composite score has been devised that integrates the above metrics into a weighted sum:

$$\text{score} = \frac{\text{metric_1} - \min(\text{metric_2})}{\max(\text{metric_3}) + \min(\text{metric_4})} \quad (18)$$

The weights have been assigned as follows: each metric is considered equally important from our perspective. In order to facilitate comparative analysis and enhance understandability, it is necessary that the elements sum to 1. Consequently, each element will cover 0.25 of the total score. To facilitate a fuller understanding of these metrics, a brief overview of their operating principles is provided below:

- MAE: measures the average absolute error without taking into account the size of individual errors.
- MSE: penalises larger errors more because it squares the individual errors.
- L2RE: is a measure of the relative error in the L2 standard and behaves similarly to the MSE.
- MAX_APE: measures the maximum absolute percentage error, highlighting the largest single deviation as a percentage of the true value.

The overall metric may change depending on the metric of interest selected. The current analysis has no preference, indicating that both mean error and contour error are of equal interest.

As previously stated, the main library used to implement the NHBO is DeepXDE. The algorithm used by this library has a special handling related to the loss functions. Minimizing this loss function means minimizing all its components. Our loss function contains different residuals related to the boundary and initial conditions along with the ones from the PDE equation.

$$\mathcal{L}(\theta, \mathcal{T}) = \omega_f \mathcal{L}_f(\theta, \mathcal{T}_f) + \omega_b \mathcal{L}_b(\theta, \mathcal{T}_b) \quad (19)$$

The equation (19) specifies the structure used by DeepXDE library. The leftmost part represents the residual loss related to the domain, while the rightmost part represents the residual loss of the boundary and initial conditions. In our case we will have four components for the One dimensional case, since we have two boundary conditions, one initial condition and one component related to the equation, and six components for the Two dimensional case, which has two more boundary conditions.

IV. ADAM & L-BFGS

The application of optimisation algorithms is of pivotal importance in the training of neural networks, as they

determine the manner in which the model's parameters are adjusted in order to achieve a reduction in the loss function. Two optimisers that are widely used in this field are Adam (which stands for Adaptive Moment Estimation) and L-BFGS (which stands for Limited-memory Broyden–Fletcher–Goldfarb–Shanno). While Adam is a first-order optimiser that is widely employed in deep learning, L-BFGS is a quasi-Newton method, a second-order optimisation technique. Each of these algorithms possesses distinctive characteristics, strengths, and limitations, rendering them appropriate for disparate scenarios, including their deployment in Physics-Informed Neural Networks (PINNs). Adam, an acronym for Adaptive Moment Estimation, is a first-order optimisation algorithm that has gained considerable popularity in the field of deep learning. The core concept of Adam is to integrate the advantages of two existing algorithms, namely AdaGrad and RMSProp, through the implementation of adaptive learning rates for each parameter. Adam employs estimates of both the first moment (mean) and the second moment (uncentered variance) of the gradients to adaptively adjust the learning rate for each parameter. This enables Adam to demonstrate greater resilience to noisy gradients, which are a common occurrence in deep learning tasks.

Pros:

- Increased robustness to noisy gradients
- Efficient for large-scale problems
- Adaptive learning rates
- Good performance in non-stationary environments

Cons:

- Sensitivity to hyperparameters
- Suboptimal convergence
- Difficulty with saddle points

L-BFGS, or Limited memory Broyden Fletcher Goldfarb Shanno, is a quasi-Newton optimisation method that approximates the second derivatives (Hessian matrix) to provide guidance for parameter updates. In contrast to first-order methods, L-BFGS utilises past gradients to approximate the Hessian matrix, thereby facilitating an enhanced comprehension of the curvature of the loss surface. This second-order information enables L-BFGS to make more informed updates, which may ultimately result in a more precise convergence.

Pros:

- High precision in convergence
- Rapid convergence for well-behaved problems
- Effective for smaller datasets
- Better for problems with smooth surfaces

Cons:

- High computational cost
- Not ideal for very large models or datasets
- Less adaptable to noisy environments
- Complexity in implementation

We have selected L-BFGS for its reputation for providing high precision in the solution of partial differential equations (PDEs) through the utilisation of a quasi-Newton method,

which approximates the Hessian matrix to facilitate more informed updates. This results in more accurate convergence compared to first-order methods like Adam. Furthermore, L-BFGS can achieve faster convergence due to its use of second-order information, which enables it to reach precise solutions with fewer iterations. This is particularly advantageous in PINNs, where training is typically conducted on smaller datasets with high-dimensional inputs, making full-batch training feasible. In such scenarios, L-BFGS is particularly effective as it utilises the entire dataset in each iteration, leading to efficient and accurate updates. In several research studies, due to the computational expense of L-BFGS, it is often employed as a first Adam to initiate the learning procedure, and then switched to L-BFGS as in [8].

V. 1 DIMENSIONAL CASE

As we have seen in the previous Chapters, the Pennes equation with One dimension spatial domain can be written as follows:

$$a_1 \partial_\tau \theta = \partial_{XX} \theta - a_2 \theta + a_3 \quad (20)$$

In this context we have not introduced the presence of the metabolic heat within the tissue. This means that we have set $Q = 0$, hence we will have $a_3 = 0$. We have used two different initial condition, which specify the temperature distribution within the tissue at the start of the observation period when $\tau = 0$:

$$\theta(X, 0) = 0 \quad (21)$$

This is the one used inside the Mathematica Simulation, while the following one is what we have used inside the NHBO model.

$$\theta(X, 0) = q_0 \frac{X^4}{4(T_{max} - T_a)} \quad (22)$$

The surface of our model has only two boundary conditions. The Left Boundary condition is defined as follows:

$$\theta(0, \tau) = 0 \quad X = 0, \tau \geq 0 \quad (23)$$

This condition has the structure of a Dirichlet condition, since it specifies the temperature value on an edge of the spatial domain. The Right boundary condition, the one that specifies the behaviour of the flux when $X = 1$, is defined as a Robin boundary condition :

$$\partial_x \theta(1, \tau) = K(\tau + \frac{1 - \tau}{1 + e^{-s(t-t_c)}} - \theta(1, \tau)) \quad \tau \geq 0 \quad (24)$$

This type of conditions are similar to the Neumann conditions due to their structure. The main difference lies on the presence of the function value along with the derivative on the boundary in the Robin's case, while we only have the derivative in Neumann's case.

$$a(x)T(1, \tau) + b(x)\partial_x \theta(1, \tau) = c(x) \quad x \in \partial\Omega \quad (25)$$

$$\partial_x \theta(1, \tau) = h(x) \quad x \in \partial\Omega \quad (26)$$

This two equation represent a simple case for a Robin (25) condition and for a Neumann (26) condition, both defined on a boundary $\partial\Omega$.

VI. 1D RESULTS WITHOUT Q

As previously stated, the NHBO model is trained using the DeepXDE library, which is able to handle temporal and spatial variations in temperature through the application of deep learning techniques. This framework ensures that the model is capable of predictively adapting to the complexity of heat distribution, which is essential for planning effective Hyperthermia treatments. In this initial approach, we exclude the Q factor, which eliminates internal heat production. This simplified version of the problem occurs when there is no metabolic heat generation within the tissue and no heat absorption due to the heating source. From this point forward, our objective is to assess the efficacy of distinct combinations of activation functions and weight initialization schemes. The principal objective is to ascertain the optimal configurations that will yield the best performance for the network. To investigate the influence of network depth, the parameter NumDenseLayers was modified, with configurations evaluated using 2, 3, and 4 dense layers. Each activation function was paired with a specific weight initialization scheme, with a single optimal initialization identified for each function, except for APTx, which requires two distinct initialization strategies:

- He normal: paired with ELU, GELU, ReLU, SiLU, Swish, Mish, and APTx.
- Glorot normal: paired with SELU, Sigmoid, sin, tanh, Softplus, and APTx.

He normal initialisation is designed for activation functions such as ReLU, whereby the weights are set based on a distribution that promotes stable gradients in deep networks. The Glorot normal (or Xavier normal) initialisation is suited for activation functions such as Tanh and Sigmoid, with the aim of maintaining a consistent gradient flow throughout the network during training. Additionally, different numbers of training iterations were explored to assess their impact on performance. For networks with two and three dense layers, the number of Iterations tested ranged from 50 to 500 in increments of 25 or 50. In the case of networks with four dense layers, the iteration counts tested were 250, 500, 750, and 1000. The following tables illustrate the results obtained:

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|-------------|---------------|----------------|
| APTx | 500 | 1072 | 0.0324 | He normal |
| APTx | 1000 | 1193 | 0.2017 | Glorot normal |
| Softplus | 1000 | 1363 | 0.3666 | Glorot normal |
| Mish | 1000 | 1645 | 0.0274 | He normal |
| tanh | 1000 | 1982 | 0.4867 | Glorot normal |
| Swish | 1000 | 2117 | 0.0280 | He normal |
| sin | 1000 | 2303 | 0.4794 | Glorot normal |
| SiLU | 1000 | 2440 | 0.0280 | He normal |
| Sigmoid | 1000 | 2632 | 0.4294 | Glorot normal |
| SELU | 1000 | 2747 | 0.3524 | Glorot normal |
| ReLU | 50 | 3081 | 0.1091 | He normal |
| GELU | 500 | 3137 | 0.0264 | He normal |
| ELU | 1000 | 3255 | 0.0418 | He normal |

TABLE II: 2 Dense Layers

As highlighted, the bolded values represent the optimal results within each category. Table II showcases the results of using different activation functions (AF) and initialization methods with 2 dense layers. The performance is measured in terms of runtime and score. The GELU activation function with He normal initialization yields the best score (0.0264) in this configuration, closely followed by Mish and Swish with slightly higher runtimes but similarly low scores. It is evident that the choice of activation function, along with the appropriate initialization, significantly influences both the runtime and the model's accuracy.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|-------------|---------------|----------------|
| APTx | 1000 | 1005 | 0.0708 | He normal |
| APTx | 1000 | 1234 | 0.0619 | Glorot normal |
| Softplus | 1000 | 1445 | 0.2858 | Glorot normal |
| Mish | 1000 | 1820 | 0.0918 | He normal |
| tanh | 1000 | 2224 | 0.1152 | Glorot normal |
| Swish | 1000 | 2350 | 0.0848 | He normal |
| sin | 1000 | 2563 | 0.1025 | Glorot normal |
| SiLU | 1000 | 2698 | 0.0848 | He normal |
| Sigmoid | 25 | 3038 | 0.4324 | Glorot normal |
| SELU | 1000 | 3040 | 0.1323 | Glorot normal |
| ReLU | 25 | 3416 | 0.0918 | He normal |
| GELU | 750 | 3448 | 0.0746 | He normal |
| ELU | 1000 | 3575 | 0.1390 | He normal |

TABLE III: 3 Dense Layers

Table III presents the results with 3 dense layers. Here, the APTx activation function with Glorot normal initialization shows a notable improvement in score (0.0619) compared to its performance with 2 dense layers, indicating that a deeper network benefits from this configuration. Interestingly, the GELU activation function, which performed best with 2 dense layers, continues to perform well but does not achieve the top score in this deeper network. This suggests that the optimal configuration may shift as the network complexity increases, underscoring the importance of experimenting with different combinations.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| APTx | 250 | 806 | 0.0755 | He normal |
| APTx | 1000 | 895 | 0.4448 | Glorot normal |
| Softplus | 1000 | 1145 | 0.1643 | Glorot normal |
| Mish | 500 | 1830 | 0.0946 | He normal |
| tanh | 1000 | 2165 | 0.1311 | Glorot normal |
| Swish | 750 | 2419 | 0.0688 | He normal |
| sin | 1000 | 2647 | 0.1135 | Glorot normal |
| SiLU | 750 | 2912 | 0.0688 | He normal |
| Sigmoid | 1000 | 3153 | 0.1628 | Glorot normal |
| SELU | 750 | 3439 | 0.3369 | Glorot normal |
| ReLU | 1000 | 3732 | 0.2362 | He normal |
| GELU | 150 | 4088 | 0.0690 | He normal |
| ELU | 500 | 4274 | 0.1389 | He normal |

TABLE IV: 4 Dense Layers

Table IV shows the outcomes for 4 dense layers. As seen, the Swish and SiLU activation functions, both paired with He normal initialization, achieve the lowest scores (0.0688), indicating superior performance for deeper networks. The GELU function, which performed well in previous

configurations, also shows strong performance but with a slightly higher runtime, possibly due to increased network complexity. This table emphasizes that deeper networks may benefit from activation functions that maintain stability and support effective gradient flow, as seen with Swish and SiLU in this case.

| AF | Iter. | Runtime | Score | Initialization |
|-------|-------|-----------|---------------|----------------|
| APTx | 500 | 14 | 0.9999 | He normal |
| GELU | 500 | 38 | 0.0000 | He normal |
| Mish | 1000 | 19 | 0.5000 | He normal |
| Swish | 1000 | 110 | 0.5000 | He normal |
| SiLU | 1000 | 159 | 0.5000 | He normal |

TABLE V: Dense Nodes 75

Table V explores the impact of varying the number of dense nodes, in this case using 75 nodes. The GELU activation function, paired with He normal initialization, achieves the perfect score (0.0000), indicating that it is exceptionally well-suited for this specific configuration. The APTx function, although showing very low runtime, produces a significantly higher score, suggesting that while it is computationally efficient, it may not be the most accurate in this configuration.

| AF | Iter. | Runtime | Score | Initialization |
|-------|-------|-----------|---------------|----------------|
| Mish | 1000 | 17 | 0.9999 | He normal |
| Swish | 1000 | 147 | 0.0000 | He normal |
| SiLU | 1000 | 210 | 0.0000 | He normal |
| APTx | 500 | 10 | 0.9998 | He normal |
| GELU | 500 | 41 | 0.0000 | He normal |

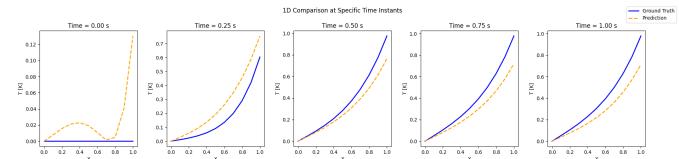
TABLE VI: Dense Nodes 100

Table VI presents the results when using 100 dense nodes. Once again, the Swish and SiLU activation functions, both with He normal initialization, achieve perfect scores (0.0000). The APTx function, although having the fastest runtime, does not achieve a similarly low score, suggesting that its efficiency may come at the cost of accuracy. An observation of the table reveals that SiLU and GELU are returning identical values (0.0000). GELU is the best model with 500 iterations and SiLU is one of the best models with 1000. To facilitate a more comprehensive evaluation of these two methods, two principal metrics may be employed to ascertain the behavioural preference of the aforementioned models:

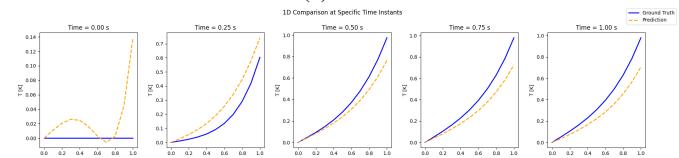
| Network Activation | L2RE | MSE |
|--------------------|--------------|---------------|
| SiLU | 0.304959534 | 0.01476774416 |
| GELU | 0.3080251058 | 0.01506613866 |

TABLE VII: Comparison of L2RE and MSE for SiLU and GELU

Although the discrepancies are slight, the results indicate that SiLU exhibits a slight advantage over GELU in both L2RE and MSE, suggesting that it is the superior activation function for reducing prediction errors in this context. The subsequent section presents the graphs illustrating the performance of the two models in question.



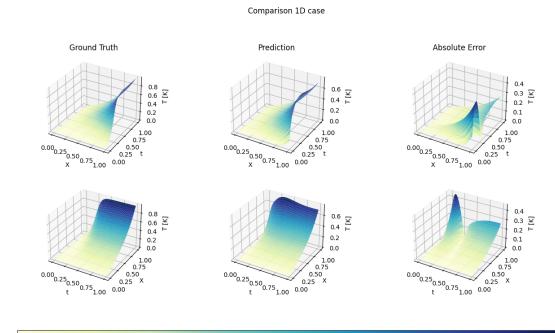
(a) GELU



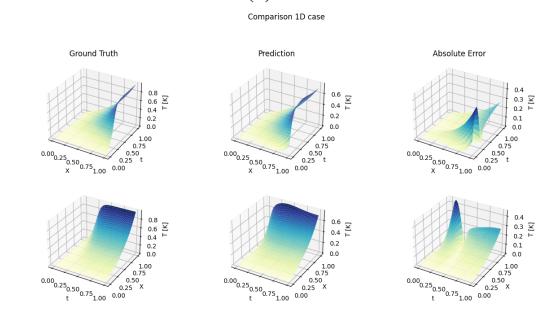
(b) SiLU

Fig. 1: Comparison of GELU, SiLU in 2D

The graphs demonstrate a high degree of similarity. They illustrate the discrepancy between the anticipated and actual values. The initial configuration is depicted in the first graph of each sequence. In the case of SiLU (1b), the value falls below zero, which is not a viable outcome. It represents an inaccurate approximation made by the program, which has no bearing on the subsequent results.



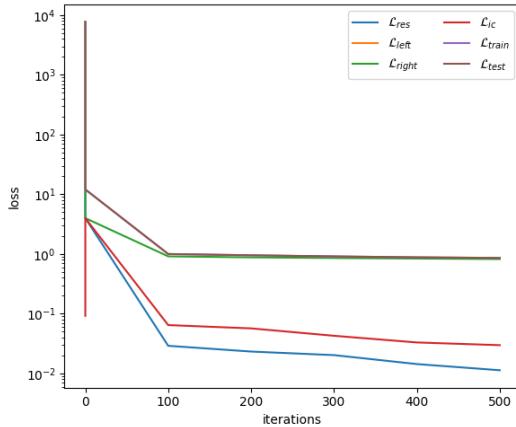
(a) GELU



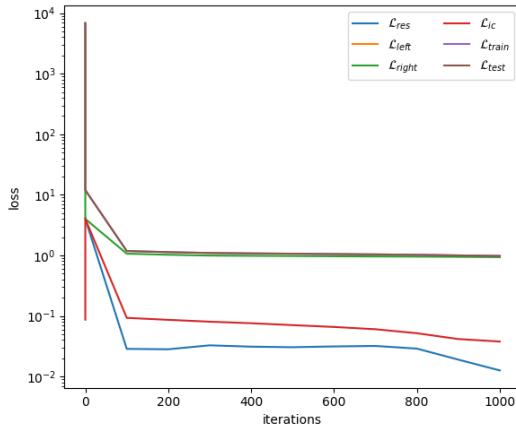
(b) SiLU

Fig. 2: Comparison of GELU, SiLU in 3D

Four graphs have been constructed for the purpose of facilitating a more profound comprehension of the profiles of the simulations.



(a) GELu



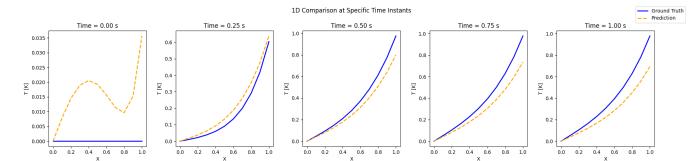
(b) SiLU

Fig. 3: Comparison of losses: GELU, SiLU

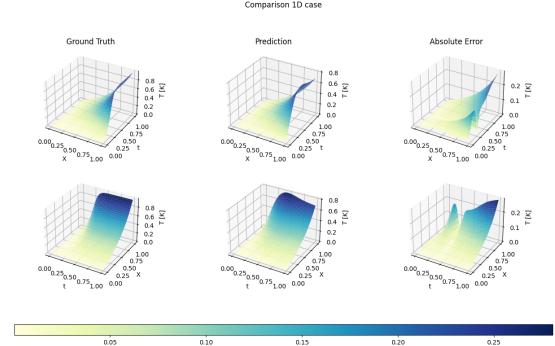
To conclude this section of the analysis, we have also tried L-BFGS as the optimization algorithm. These runs have been tuned using the starting configuration with 50 nodes and 2 dense layers, focusing on the number of iterations.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|----------|---------------|----------------|
| APTx | 25 | 59 | 0.1024 | He normal |
| APTx | 25 | 51 | 0.0379 | Glorot normal |
| Softplus | 25 | 22 | 0.5489 | Glorot normal |
| Mish | 25 | 147 | 0.0841 | He normal |
| tanh | 25 | 37 | 0.0205 | Glorot normal |
| Swish | 25 | 60 | 0.0750 | He normal |
| sin | 25 | 42 | 0.0199 | Glorot normal |
| SiLU | 25 | 54 | 0.0750 | He normal |
| Sigmoid | 25 | 27 | 0.2810 | Glorot normal |
| SELU | 25 | 9 | 0.5151 | Glorot normal |
| ReLU | 25 | 4 | 0.1762 | He normal |
| GELU | 25 | 52 | 0.0960 | He normal |
| ELU | 25 | 11 | 0.0369 | He normal |

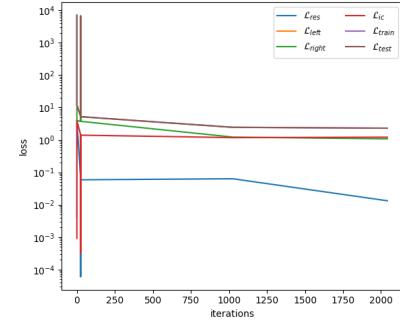
TABLE VIII: LBFGS with 100 Dense Nodes



(a) Comparison 2D between prediction and ground truth for the sin model



(b) Comparison 3D between prediction and ground truth for the sin model



(c) Loss of the sin model

Fig. 4: Sin results

The results indicate that the sin model (0.0199) exhibits the optimal performance, although it remains below the values observed in the absence of LBFGS. This may be attributed to the limited number of iterations, despite the algorithm's suitability for this task. However, an examination of the profiles of the 3D plot (3b) reveals that the behaviour of the predicted model has a lower error with respect to GELU (2a and SiLU 2b) models, where L-BFGS was not used. In addition to these tests, another parameter was modified, the Injection Gain, between 50 (default), 75, and 100. Despite this modification, the resulting data were of no interest, since they had a similar behaviour to the data presented in previous tests.

VII. 2 DIMENSIONAL CASE

Starting from the 1D chapter V, the general formulation of our equation, must be reformulated.

$$a_1 \partial_\tau \theta = \partial_{XX} \theta - a_2 \theta + a_3 \quad (27)$$

In this instance, the equation must be transformed into a two-dimensional representative model that adheres to the same structural principles as before:

$$a_1 \partial_\tau \theta = \nabla^2 \theta - a_2 \theta + a_3 \quad (28)$$

Which can be simplified into:

$$a_1 \partial_\tau \theta = \partial_{XX} \theta + \partial_{YY} \theta - a_2 \theta + a_3 \quad (29)$$

Since our equation is composed by two spatial coordinates, it is possible to define a Square-like surface. As a consequence there will be Four different boundary conditions instead of Two, like in the One dimensional case. These conditions represent the behaviour of the Temperature on the left, right, upper and lower boundary of the square. The Initial condition is the same as the one defined previously:

$$\theta(X, Y, 0) = 0 \quad X \in [0, 1], Y \in [0, 1], \tau = 0 \quad (30)$$

The Upper (31) and Lower (32) conditions are both defined on the normal derivative to the boundary, hence they adhere to the Neumann's structure.

$$\partial_Y \theta(X, 0, \tau) = 0 \quad Y = 0, \tau \geq 0 \quad (31)$$

$$\partial_Y \theta(X, 1, \tau) = 0 \quad Y = 1, \tau \geq 0 \quad (32)$$

The Left (33) condition has the structure of a Dirichlet's and it is equal to the one used in the One dimensional case.

$$\theta(0, Y, \tau) = 0 \quad X = 0, \tau \geq 0 \quad (33)$$

The Right (34) condition defines a linear behaviour, with respect to time, of the normal derivative to the boundary, hence it is another Neumann condition.

$$\partial_X \theta(1, Y, \tau) = \tau \quad X = 1, \tau \geq 0 \quad (34)$$

From the equation (29), one of the main difference with respect to the One dimensional case can be found in the value of $a_3 = \frac{Q L_0^2}{(T_M - T_a) k_{\text{eff}}}$. This coefficient contains the Metabolic heat Q within the tissue, which was directly set to zero in the previous case. The Two dimensional case has been analyzed in both configurations. As previously seen, the Metabolic heat is composed by two elements:

$$Q = q_m + q_p \quad [\text{W/m}^3] \quad (35)$$

The first one, q_m represent the metabolic heat generation of the tissue, while the q_p represent the energy deposition rate from the heating procedure. In our analysis we have chosen the Breast tissue and its characteristic to perform our experiments. Breast cancer is the most common type of cancer in the world with an estimated incidence rate of 12.5%, which amounts to around 25% of all the cancer diagnoses in women. Different modalities have been used as treatments like Radiotherapy, Chemotherapy and more recently Hyperthermia, which has been used a support of the other two treatments. This type of tissue is heterogeneous, composed by fat and fibroglandular tissue. The amount of fatty tissue increases with age and is

highly variable between individuals. The heterogeneity of the tissue makes it difficult to generate standard numerical and physical model. The tumour tissue is variable, since it depends on the type of cancer. Since data are highly specific to the subject breast cancer, general value have been chosen for the metabolic heat generation rates.

| Tissue Type | Normal Tissue (W/m^3) | Cancerous Tissue (W/m^3) |
|-------------|----------------------------------|-------------------------------------|
| q_m Value | 1.1×10^5 | 2.0×10^4 |

TABLE IX: Values used for q_m in normal and cancerous tissues

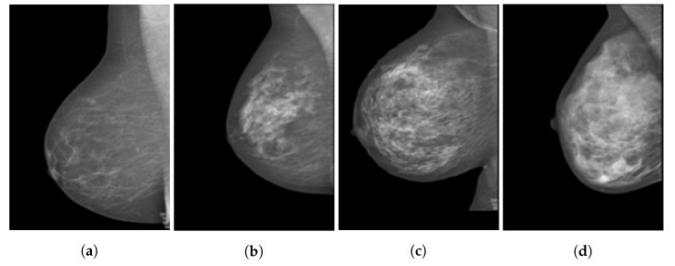


Fig. 5: Sagittal cross-section of breast MRI scans showing the breast density variations: (a) almost entirely fat, (b) scattered fibroglandular tissue, (c) heterogeneous fibroglandular tissue, and (d) extreme fibroglandular tissue

On the other hand, in order to define the value of the energy deposition rate, some assumptions have been made. Related to the Heating procedure, we have chose the ALBA ON4000D radiative hyperthermia system as the reference machine. It is characterized by Microwave Heating with a fixed frequency of 434 MHz. We have assumed that the energy absorption efficiency of the breast tissue was about 70% while the power erogated by the machine was about 400 W. We have made this assumption because no data were available related to this two aspects. The value of q_p has been computed as follows, by assuming a cancer volume of 35 cm^3 .

$$q_p = \frac{(\text{Absorption Efficiency})(\text{Power Erogated})}{\text{Cancer Volume}} \quad (36)$$

VIII. 2D RESULTS WITHOUT Q

This chapter presents an in-depth investigation into the optimization of the two-dimensional problem. The methodology employed in this research is consistent with that used in previous studies (Chapter: VI). The results will be evaluated based on a set of pre-defined parameters, with detailed commentary provided to facilitate the identification of the optimal combination.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| ELU | 500 | 7632 | 0.0000 | He normal |
| GELU | 500 | 5487 | 0.2470 | He normal |
| ReLU | 400 | 5084 | 0.3804 | He normal |
| SELU | 500 | 4586 | 0.1726 | Glorot normal |
| Sigmoid | 125 | 4441 | 0.2492 | Glorot normal |
| SiLU | 150 | 3973 | 0.1061 | He normal |
| sin | 50 | 3670 | 0.2329 | Glorot normal |
| Swish | 150 | 3110 | 0.1061 | He normal |
| tanh | 400 | 2477 | 0.0714 | Glorot normal |
| Mish | 150 | 2264 | 0.1506 | He normal |
| Softplus | 125 | 1859 | 0.7726 | Glorot normal |
| APTx | 300 | 1271 | 0.2762 | Glorot normal |
| APTx | 300 | 834 | 0.276 | He normal |

TABLE X: 2 Dense Layers

Table X shows the performance of various activation functions (AF) using 2 dense layers. The ELU activation function with He normal initialization clearly outperforms others, achieving a perfect score of 0.0000, which indicates exceptional accuracy for this configuration. Conversely, Softplus with Glorot normal initialization performs the worst with a high score of 0.7726, reflecting significant errors. Interestingly, functions like SiLU and Swish, both with He normal initialization, show moderate performance, indicating their potential for further optimization. APTx demonstrates efficiency in runtime but does not achieve the lowest scores, suggesting a trade-off between speed and accuracy.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| ELU | 500 | 20731 | 0.0193 | He normal |
| GELU | 500 | 20271 | 0.1421 | He normal |
| ReLU | 500 | 19836 | 0.4709 | He normal |
| SELU | 500 | 19374 | 0.3973 | Glorot normal |
| Sigmoid | 75 | 19305 | 0.0814 | Glorot normal |
| SiLU | 500 | 18511 | 0.0681 | He normal |
| sin | 50 | 18473 | 0.3534 | Glorot normal |
| Swish | 500 | 17620 | 0.0681 | He normal |
| tanh | 50 | 17585 | 0.2139 | Glorot normal |
| Mish | 500 | 16703 | 0.1073 | He normal |
| Softplus | 500 | 16231 | 0.8569 | Glorot normal |
| APTx | 500 | 15803 | 0.1858 | Glorot normal |
| APTx | 500 | 700 | 0.1858 | He normal |

TABLE XI: 3 Dense Layers

Table XI demonstrates the performance with 3 dense layers. The ELU function again proves to be the most effective, achieving the lowest score of 0.0193, although it comes with a considerable increase in runtime. SiLU and Swish show consistent results, both achieving identical scores (0.0681) but with different runtimes, suggesting that SiLU might be slightly more efficient in this context. The APTx function, although having the fastest runtime, does not provide the lowest score, indicating that further improvements in accuracy could be necessary when prioritizing performance.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| ELU | 1000 | 2310 | 0.0236 | He normal |
| GELU | 1000 | 2129 | 0.0352 | He normal |
| ReLU | 1000 | 1964 | 0.6648 | He normal |
| SELU | 1000 | 1789 | 0.5267 | Glorot normal |
| Sigmoid | 1000 | 1635 | 0.7915 | Glorot normal |
| SiLU | 1000 | 1464 | 0.0225 | He normal |
| sin | 250 | 1423 | 0.1659 | Glorot normal |
| Swish | 1000 | 1119 | 0.0225 | He normal |
| tanh | 250 | 1072 | 0.0390 | Glorot normal |
| Mish | 1000 | 746 | 0.0375 | He normal |
| Softplus | 750 | 610 | 0.0508 | Glorot normal |
| APTx | 1000 | 405 | 0.0917 | Glorot normal |
| APTx | 1000 | 249 | 0.0917 | He normal |

TABLE XII: 4 Dense Layers

Table XII examines the impact of 4 dense layers. SiLU and Swish, both with He normal initialization, achieve the best scores (0.0225), indicating that these activation functions are highly effective in deeper networks. ELU continues to perform well, albeit with a slightly higher score than SiLU and Swish. This table highlights the importance of choosing the right activation function as the network depth increases, where functions like SiLU and Swish maintain performance while managing complexity.

| AF | Iter. | Runtime | Score | Initialization |
|-------|-------|------------|---------------|----------------|
| ELU | 250 | 13913 | 0.0042 | He normal |
| tanh | 1000 | 571 | 0.2514 | Glorot normal |
| SiLU | 750 | 453 | 0.9069 | He normal |
| Swish | 750 | 306 | 0.9069 | He normal |
| Mish | 750 | 136 | 0.7563 | He normal |

TABLE XIII: Dense Nodes 75

Table XIII explores the effect of increasing the number of dense nodes to 75. ELU, once again with He normal initialization, achieves the best score (0.0042), confirming its robustness across different configurations. However, functions like SiLU and Swish perform poorly with higher scores (0.9069), indicating that these functions may not scale well with increased node numbers in this context. Mish shows some promise with a relatively low runtime but does not surpass ELU in accuracy.

| AF | Iter. | Runtime | Score | Initialization |
|-------|-------|------------|----------------|----------------|
| ELU | 1000 | 814 | 0.04576 | He normal |
| tanh | 1000 | 653 | 0.2769 | Glorot normal |
| SiLU | 1000 | 472 | 0.1583 | He normal |
| Swish | 1000 | 299 | 0.1583 | He normal |
| Mish | 1000 | 100 | 0.2721 | He normal |

TABLE XIV: Dense Nodes 100

Table XIV increases the number of dense nodes to 100. Here, ELU continues to perform well but with a slightly higher score (0.04576) compared to previous configurations, suggesting that increasing nodes may introduce diminishing returns. SiLU and Swish, with identical scores (0.1583), show moderate performance, indicating that their effectiveness plateaus with increased nodes. Mish, despite having the fastest runtime, does not achieve the best score, further underscoring the trade-off between speed and accuracy.

| AF | Iter. | Runtime | Score | Initialization |
|-------|-------|------------|---------------|----------------|
| ELU | 500 | 7155 | 0.0000 | He normal |
| tanh | 500 | 569 | 0.5413 | Glorot normal |
| SiLU | 1000 | 375 | 0.5391 | He normal |
| Swish | 1000 | 242 | 0.5391 | He normal |
| Mish | 750 | 134 | 0.7689 | He normal |

TABLE XV: Output Injection Gain 75

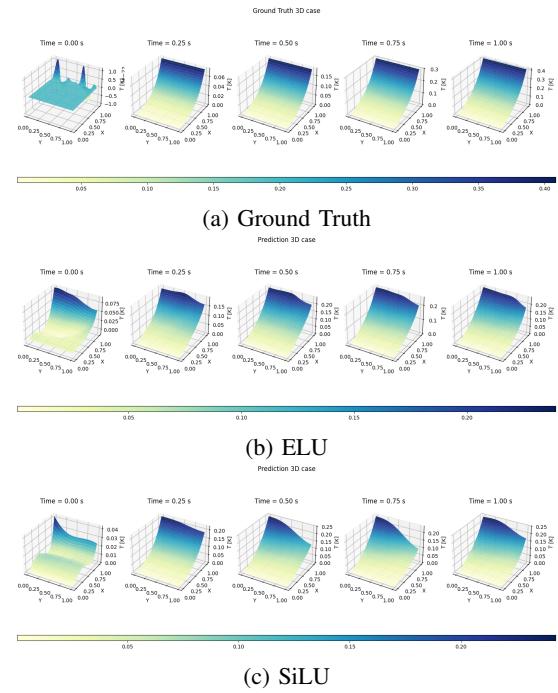
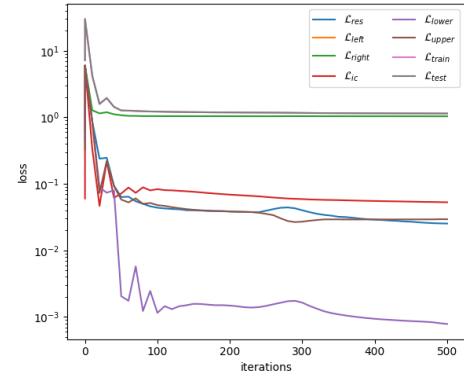


Table XV explores the impact of increasing the Output Injection Gain to 75. The ELU function once again demonstrates its superiority with a perfect score (0.0000), reaffirming its consistency and reliability across different settings. Other activation functions, such as SiLU and Swish, show comparable performance with identical scores (0.5391), yet they fall short of ELU's precision. Mish, although efficient in terms of runtime, does not perform as well in terms of accuracy.

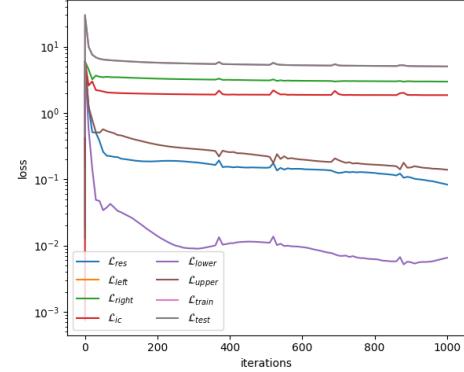
Fig. 6: Comparison between ELU and SiLU on the predictions

| AF | Iter. | Runtime | Score | Initialization |
|-------|-------|------------|---------------|----------------|
| ELU | 500 | 7155 | 0.0000 | He normal |
| tanh | 500 | 569 | 0.5413 | Glorot normal |
| SiLU | 1000 | 375 | 0.5391 | He normal |
| Swish | 1000 | 242 | 0.5391 | He normal |
| Mish | 750 | 134 | 0.7689 | He normal |

TABLE XVI: Output Injection Gain 100



(a) ELU



(b) SiLU

Fig. 7: Comparison between ELU and SiLU on the losses

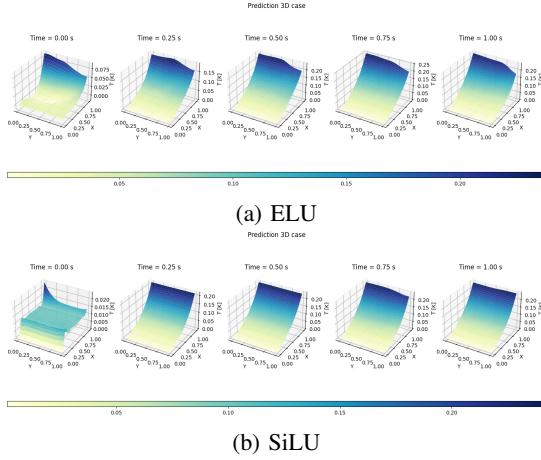


Fig. 8: Comparison between ELU and SiLU on the predictions with L-BFGS

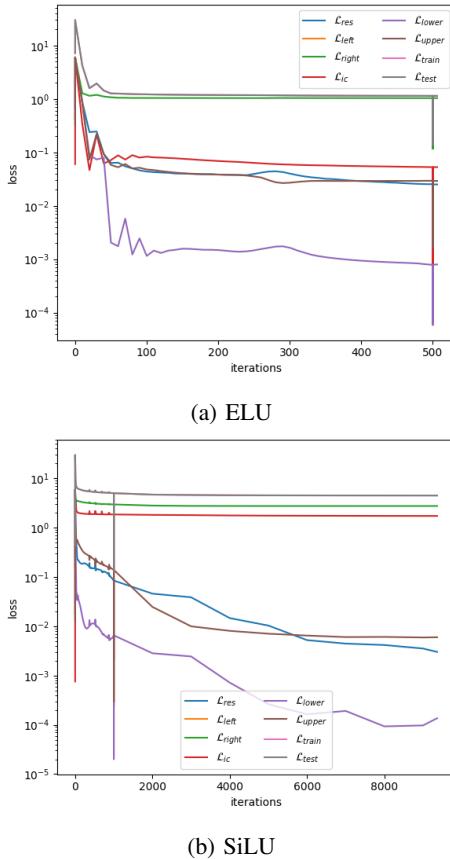


Fig. 9: Comparison between ELU and SiLU on the losses with LBFGS

The results demonstrate that ELU exhibited the optimal configuration, with a notable advantage over the other settings. This is also evident in the graphs, particularly when focusing on Time 0.75 seconds (6b). It can be observed that the profiles of SiLU generate a significant error in the value of Y=1 and X=1. The general difference in performance is evident in the

losses, which exhibit a considerable discrepancy. However, this difference is less pronounced when the lbfsgs optimisation algorithm is introduced (9). The 3D image 8 displays enhanced smoothness and exhibits a closer alignment with the ground truth profile.

IX. 2D RESULTS WITH Q

In this chapter, as in the preceding one, we shall analyse the results obtained by the simulations. The techniques and hyperparameters employed are the same as those used previously. Our focus shall be solely on the results.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|-------------|---------------|----------------|
| ELU | 25 | 41830 | 0,0295 | He normal |
| GELU | 25 | 26908 | 0,5346 | He normal |
| ReLU | 25 | 26476 | 0,1213 | He normal |
| SELU | 25 | 26060 | 0,1029 | Glorot normal |
| Sigmoid | 1000 | 25363 | 0,0856 | Glorot normal |
| SiLU | 25 | 25335 | 0,0211 | He normal |
| sin | 25 | 24939 | 0,1020 | Glorot normal |
| Swish | 25 | 24546 | 0,0211 | He normal |
| Tanh | 25 | 3641 | 0,0679 | Glorot normal |
| Mish | 25 | 3090 | 0,0315 | He normal |
| SoftPlus | 1000 | 1943 | 0,0350 | Glorot normal |
| APTx | 25 | 1917 | 0,0587 | Glorot normal |
| APTx | 25 | 1496 | 0,0587 | He normal |

TABLE XVII: 2 Dense Layers

In the table XVII, the best score was achieved by both SiLU and Swish, with a score of 0.0211. These activation functions, paired with He normal initialization, offer an optimal configuration for networks with two dense layers. The lowest runtime was recorded for SoftPlus with Glorot normal initialization, indicating its efficiency in computational time.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| SiLU | 50 | 17843 | 0,0099 | He normal |
| Swish | 50 | 1056 | 0,0099 | He normal |
| ELU | 50 | 831 | 0,1199 | He normal |
| Mish | 50 | 600 | 0,0237 | He normal |
| SoftPlus | 75 | 306 | 0,2110 | Glorot normal |

TABLE XVIII: 3 Dense Layers

In the table XVIII, SiLU and Swish again deliver the best score of 0.0099. However, Swish also shows a significantly lower runtime, making it the superior choice in this configuration. SoftPlus, despite a higher score, achieves the lowest runtime at 306, which may be preferred in scenarios where computational efficiency is critical.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| SiLU | 50 | 63985 | 0,0205 | He normal |
| Swish | 50 | 983 | 0,0205 | He normal |
| ELU | 50 | 774 | 0,0841 | He normal |
| Mish | 50 | 561 | 0,0186 | He normal |
| SoftPlus | 50 | 355 | 0,0245 | Glorot normal |

TABLE XIX: 4 Dense Layers

In the table XIX, Mish emerges as the top performer with a score of 0.0186. Additionally, SoftPlus maintains its trend

of achieving the lowest runtime. This suggests that SoftPlus could be a strong candidate in configurations requiring four dense layers, especially when runtime is a crucial factor.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| SiLU | 1000 | 1023 | 0,0394 | He normal |
| Swish | 1000 | 820 | 0,0394 | He normal |
| ELU | 50 | 783 | 0,3171 | He normal |
| Mish | 50 | 580 | 0,0954 | He normal |
| SoftPlus | 500 | 196 | 0,1695 | Glorot normal |

TABLE XX: Dense Layers 75

In the table XX, SiLU and Swish are tied with the best score of 0.0394, demonstrating consistency across different network configurations. SoftPlus, while not having the best score, once again achieves the lowest runtime, highlighting its computational efficiency.

| AF | Iter. | Runtime | Score | Initialization |
|----------|-------|------------|---------------|----------------|
| SiLU | 50 | 16306 | 0,0099 | He normal |
| Swish | 50 | 1085 | 0,0099 | He normal |
| ELU | 50 | 864 | 0,1199 | He normal |
| Mish | 50 | 623 | 0,0237 | He normal |
| SoftPlus | 100 | 277 | 0,2059 | Glorot normal |

TABLE XXI: Dense Layers 100

In the final table XXI, SiLU and Swish consistently achieve the best score of 0.0099, reinforcing their effectiveness in configurations with a higher number of dense layers. SoftPlus, despite a higher score, continues to deliver the lowest runtime, making it a viable option in scenarios where time efficiency is prioritized. In this comparison, the difference in the weighted score is zero, as illustrated in the 3D representation, which depicts the profile picture with greater specificity in relation to the aforementioned examples. Each profile is distinct and does not indicate a preference for any particular outcome. Upon examination of the results, it can be inferred that SiLU, with its smooth profile and balanced axes, is the optimal choice due to its ability to reach the goal with greater efficiency.

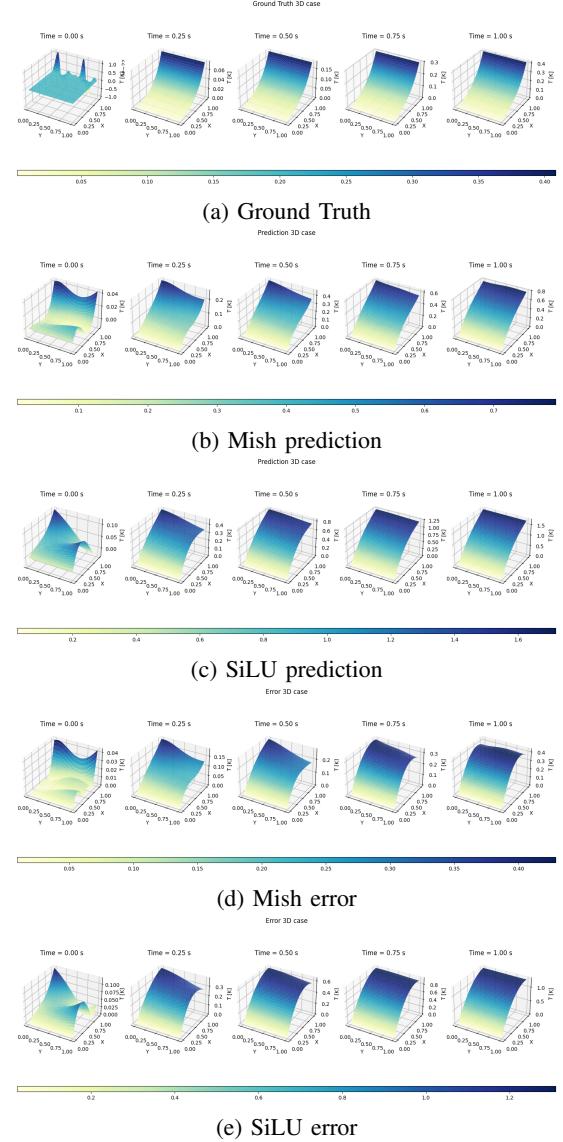
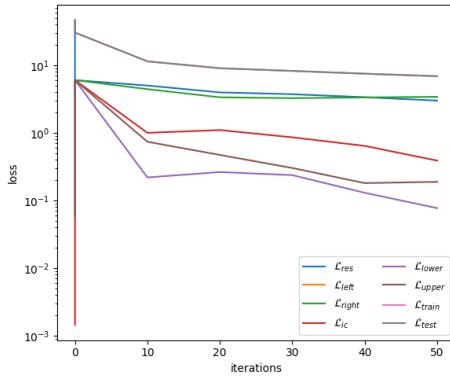
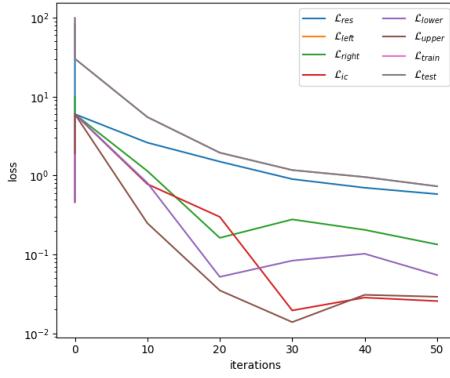


Fig. 10: Comparison between Mish and SiLU on predictions and error

As we can see, the performance obtained here have an higher error with respect to the prediction obtain in the Two dimensional case. Probably this is related to all the assumption that we have made related to the enery deposition rate with respect to the heating source. Our predictions reach higher values of temperature with respect to what we have in our Ground Truth model. This behaviour is the opposite of what happened in both One and Two dimensional case without \mathcal{Q} , since only in one specific case we obtained the correct final value of the temperature.



(a) Mish



(b) SiLU

Fig. 11: Comparison between Mish and SiLU on the losses

X. CONSUMES

The following chapter presents an analysis of various aspects pertaining to the system's consumers. Specifically, three aspects are examined: CPU utilisation, disk utilisation, and system memory utilisation. The chapter presents the system's consumables in a well-ordered manner, facilitating a comprehensive analysis and comparison.

XI. 1D CONSUMES

The following graph illustrates the CPU usage over time for various configurations tested by our research team on a 1D problem. Each line corresponds to a unique combination of weight initialization and activation function used in the neural network model to address the Bio-Heat equation in a one-dimensional domain. The x-axis represents time in seconds, while the y-axis shows CPU usage as a percentage.

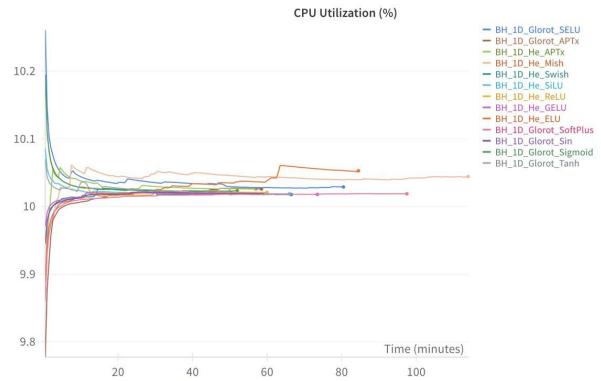


Fig. 12: CPU Utilization (%)

The following section will present two distinct approaches to memory utilisation, namely memory swap and no-swap. The distinction between memory swap and non-swap lies in the manner by which the operating system oversees the management of physical memory (RAM) and disk space. In contrast, non-swap memory refers to data stored directly in RAM, which is a relatively fast storage medium but has a limited capacity. When the RAM is full, the system utilises swap memory, which involves moving less critical or inactive data to disk storage (swap space) to free up RAM. Although swap space prevents the system from running out of memory, it is significantly slower than RAM because disk access is much slower than accessing physical memory. Swap memory helps maintain system stability but can reduce performance if overused.

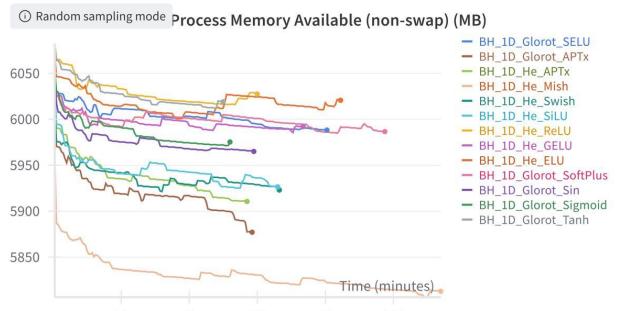


Fig. 13: Memory with no Swap (%)

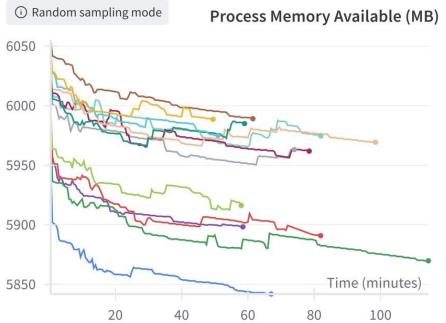


Fig. 14: Memory with Swap (%)

It is notable that the dependency of the Mish Activation Function on RAM is significantly greater in non-swap utilisation than in swap utilisation. This is similar to the behaviour observed in SiLU and other main AF with high scores.

XII. 2D CONSUMES WITHOUT Q

The following graph depicts the CPU utilisation over time for a variety of configurations pertaining to a 2D problem utilised by our research group. Each line represents a distinct combination of weight initialisation and activation function employed in the NN model to solve the Bio-Heat equation in a 2D domain. The x-axis depicts time in seconds, while the y-axis represents CPU utilisation in percentage.

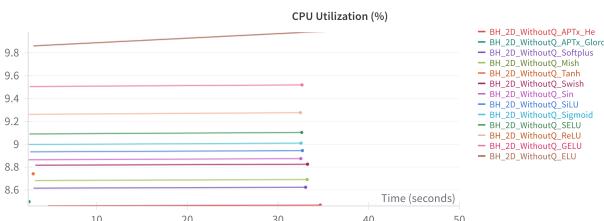


Fig. 15: CPU Utilization (%)

The graph's results suggest that the selection of activation functions and initialization methods doesn't significantly impact the CPU load during the simulation of the Bio-Heat equation in 2D. The variations observed are minimal and indicate that all configurations are computationally feasible. However, "BH_2D_WithoutQ_APtx_He" could potentially be less optimal due to its slightly higher CPU utilization. If the focus is on minimizing computational resources, configurations with the lowest CPU utilization, like "BH_2D_WithoutQ_APtx_Glorot" and "BH_2D_WithoutQ_ELU," might be preferable. However, if the quality of the solution is more critical than computational efficiency, these choices should be balanced with the accuracy and convergence of the neural network, which would require further analysis beyond CPU utilization. The next graph shows the disk utilization over time (in GB) for various configurations of a 2D problem using different initialization methods and activation functions.

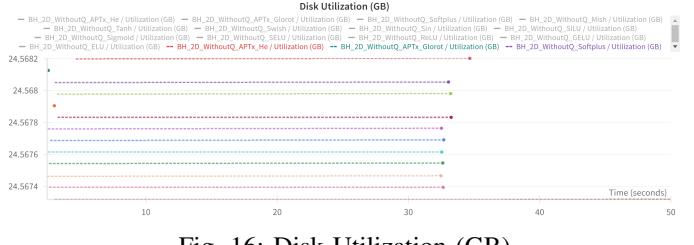


Fig. 16: Disk Utilization (GB)

Given the very close values for disk utilization across the different configurations, the choice of activation function and initialization scheme seems to have an insignificant effect on disk usage. This implies that other factors, such as model accuracy, convergence speed, or CPU utilization, should be prioritized when selecting the optimal configuration. The subsequent graph illustrates the system memory utilisation (expressed as a percentage) over time for a variety of configurations pertaining to the 2D Bio-Heat equation simulation. The choice of configuration has a noticeable

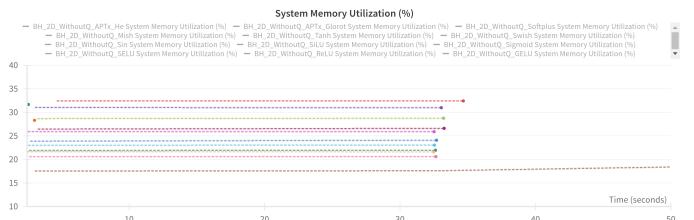


Fig. 17: System Memory Utilization (%)

impact on system memory utilization in these simulations. While all configurations are stable, they exhibit different levels of memory usage. Depending on the availability of system resources, selecting a configuration with lower memory usage could be beneficial. If memory efficiency is a priority, configurations like "BH_2D_WithoutQ_APtx_Glorot" should be preferred, as they use the least amount of system memory.

XIII. 2D CONSUMES WITH Q

The following section presents an exhaustive examination of the system's resource consumption across diverse configurations employed in the simulation of the 2D Bio-Heat equation, with a particular emphasis on three pivotal metrics: CPU utilisation, disk utilisation, and system memory utilisation. The analysis elucidates the manner in which distinct activation functions and initialization methods influence these metrics.

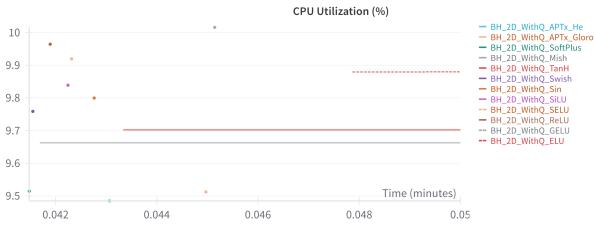


Fig. 18: CPU Utilization (%)

Figure 18 illustrates that CPU utilisation remains relatively consistent across the majority of configurations. The observed variations are minor, indicating that all configurations are computationally feasible. It is, however, noteworthy that the configuration "BH_2D_WithQ_Tanh" exhibits a slightly higher CPU utilisation than the others, which may indicate a less optimal configuration in terms of CPU efficiency. Conversely, configurations such as "BH_2D_WithQ_APTx_He" and "BH_2D_WithQ_ELU" exhibit reduced CPU utilisation, which may be advantageous when the objective is to minimise CPU load.

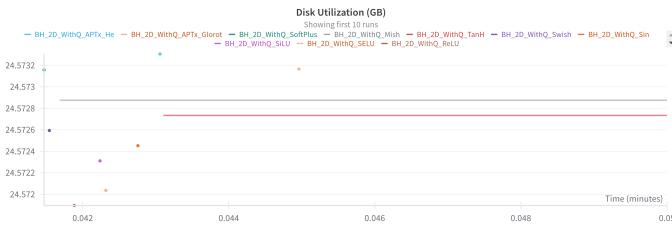


Fig. 19: Disk Utilization (GB)

Figure 19 illustrates the disk utilisation across different configurations. The results demonstrate that the choice of activation function and initialisation method has a negligible impact on disk usage, with minimal differences observed across the various configurations. Consequently, other factors such as model accuracy and runtime should be given greater consideration when selecting the optimal configuration.

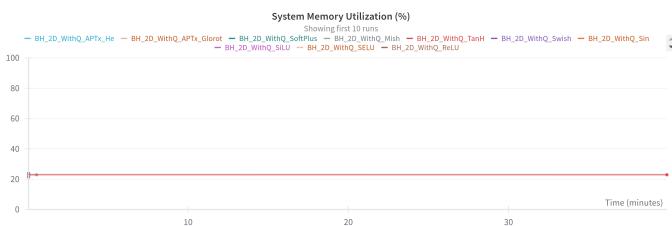


Fig. 20: System Memory Utilization (%)

Figure 20 illustrates the temporal evolution of system memory utilisation. The graph demonstrates that the configurations have a discernible impact on memory usage. While all configurations demonstrate consistent memory utilisation, "BH_2D_WithQ_Tanh" exhibits slightly elevated memory consumption in comparison to the

other configurations. Conversely, configurations such as "BH_2D_WithQ_APTx_He" and "BH_2D_WithQ_SoftPlus" exhibit reduced memory utilisation, making them optimal choices if memory efficiency is a critical consideration.

XIV. CONCLUSIONS

The study successfully demonstrates the feasibility of using Physics-Informed Neural Networks (PINNs) to solve the Pennes Bio-Heat equation for temperature estimation in hyperthermia treatments. By training the PINNs with data generated from a Mathematica model, the network can predict temperature distribution with high accuracy, even in complex 1D and 2D cases. The comparison of activation functions and optimization techniques, such as Adam and L-BFGS, reveals that ELU and SiLU activation functions, combined with He normal initialization, yield the best results. Additionally, the 2D case with heat generation (Q) provides insights into optimizing the network's performance while managing computational resources efficiently. However, the approximations made about the tissue and the machine used inside our analysis lead to worse result with respect to what we have obtained in the Two dimensional case without Q. The main reasons could be found in the definition of the properties of the Breast tissue, which has a complicated structure, and in the definition of the properties of the machine used, since most of them are not public. A further potential avenue for enhancement may be the modification of the definition of q_p , given that this value should exhibit a temporal dependency with respect to the heating procedure. This research suggests that PINNs are a powerful tool for real-time applications in medical thermal treatments, offering a balance between computational efficiency and prediction accuracy. Future work may explore the integration of different types of tissues, with the possibility to use real world data to improve the model's capabilities. Other possible improvements could be found in the definition of additional boundary conditions to enhance the generalizability of the model.

REFERENCES

- [1] A. Cristofaro, G. Cappellini, E. Staffetti, G. Trappolini, and M. Vendittelli, "Adaptive Estimation of the Pennes Bio-Heat Equation - I: Observer Design," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 1931–1936.
- [2] G. Cappellini, G. Trappolini, E. Staffetti, A. Cristofaro, and M. Vendittelli, "Adaptive Estimation of the Pennes Bio-Heat Equation - II: A NN-Based Implementation for Real-Time Applications," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. Marina Bay Sands, Singapore: IEEE, Dec. 2023.
- [3] S. Wang, X. Yu, and P. Perdikaris, "When and Why PINNs Fail to Train: A Neural Tangent Kernel Perspective," *arXiv preprint arXiv:2007.14527*, July 2020. [Online]. Available: <https://arxiv.org/abs/2007.14527>
- [4] E. H. Wissler, "Pennes' 1948 Paper Revisited," *Journal of Applied Physiology*, vol. 85, no. 1, pp. 35–41, 1998, pMID: 9655751. [Online]. Available: <https://doi.org/10.1152/jappl.1998.85.1.35>
- [5] H. H. Pennes, "Analysis of Tissue and Arterial Blood Temperatures in the Resting Human Forearm," *Journal of Applied Physiology*, vol. 1, no. 2, pp. 93–122, 1948. [Online]. Available: <https://journals.physiology.org/doi/pdf/10.1152/jappl.1998.85.1.35>
- [6] A. Bashir, S. Khan, S. Bashmal, N. Iqbal, S. Ullah, and L. Ali, "Designing Highly Efficient Temperature Controller for Nanoparticles Hyperthermia," *Nanomaterials*, vol. 12, no. 19, p. 3539, 2022.

-
- [7] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, “DeepXDE: A Deep Learning Library for Solving Differential Equations,” *arXiv preprint arXiv:1907.04502*, 2019.
 - [8] J. Lee, K. Park, and W. Jung, “Physics-Informed Neural Networks for Cantilever Dynamics and Fluid-Induced Excitation,” *Applied Sciences*, vol. 14, no. 16, p. 7002, 2024. [Online]. Available: <https://www.mdpi.com/2076-3417/14/16/7002>
 - [9] MathWorks, *Solve Partial Differential Equations with L-BFGS Method and Deep Learning*, MathWorks, 2024, available at <https://it.mathworks.com/help/deeplearning/ug/solve-partial-differential-equations-with-lbfgs-method-and-deep-learning.html>.
 - [10] Z. Cheng, W. Zhang, L. Liu, and S. Li, “Constrained by Empirical Blood Perfusion Rate, Tissue-Specific Dielectric Properties, and Average of Specific Heat Capacity Values,” *PubMed Central*, 2020. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7308290/>
 - [11] *Brochure ALBA ON4000D Hyperthermia System*, ALBA Hyperthermia, n.d. [Online]. Available: https://www.albahyperthermia.com/brochure_ALBAON4000D.pdf
 - [12] L. C. Junqueira and J. Carneiro, Eds., *Basic Histology: Text & Atlas*. National Center for Biotechnology Information (NCBI), 2003. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK13291/>
 - [13] M. Hussain, S. Jamil, and A. Tahir, “Nanotechnology-Based Thermal Therapy for Cancer Treatment,” *PubMed Central*, 2022. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9143271/>