First Experiences on Exploiting Physics-Informed Neural Networks for Approximating Solutions of a Biological model

Attilio Di Vicino[1][0009−0009−8159−9118], Pasquale De Luca[1,2][0000−0001−7031−920X], and Livia Marcellino[1,2][0000−0003−2319−8008]

[1] Department of Science and Technology, Parthenope University of Naples, Centro Direzionale C4, Naples, I-80143
attilio.divicino001@studenti.uniparthenope.it
[2] UNESCO Chair "Environment, Resources and Sustainable Development", Department of Science and Technology, Parthenope University of Naples, Centro Direzionale, Isola C4, (80143) Naples, Italy
{pasquale.deluca,livia.marcellino}@uniparthenope.it

Abstract. Recent advances in artificial intelligence have changed the ability to study and model complex biological phenomena. Physics-Informed Neural Networks (PINNs) represent a novel approach that link deep learning techniques with fundamental physical principles in solving partial differential equations. This work proposes an implementation of PINNs for modeling tumor-induced angiogenesis through a system of coupled reaction-diffusion equations that track the interplay between different biological agents. We introduce a computational framework that combines neural network architectures with physics-based constraints, using an optimized loss function incorporating both empirical data and theoretical principles via strategic collocation points. Experimental results validate the reliability of our approach in predicting the intricate spatial and temporal patterns of blood vessel formation, showing the potential of PINNs as a robust computational tool for simulating complex biological processes.

Keywords: Physics-Informed Neural Networks · Machine learning algorithms · tumor angiogenesis · numerical computations

1 Introduction

Machine learning has emerged as a powerful tool for analyzing and understanding complex systems across various scientific disciplines [1]. Among machine learning approaches, deep learning has shown good capabilities in capturing intricate patterns and relationships in data through its layered neural network architecture [2]. This capability has led to innovation in many fields, from computer vision to scientific computing [3]. Particularly, a promising contribute of deep learning is given by data assimilation that is the process of combining mathematical models with observational data to improve predictions. Traditional data assimilation

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

techniques, while effective, often address computational challenges when dealing with complex systems. Deep learning methods offer new scenarios for improving these techniques [4–6]. These advances in machine learning and data assimilation are particularly relevant for dynamical systems described by partial differential equations (PDEs). In this work, we focus on a specific system of PDEs modeling tumor-induced angiogenesis. The governing equations capture the complex spatial and temporal dynamics of several components, including diffusion, chemical signaling, and tissue degradation. In order to solve the system of PDEs, we employ Physics-Informed Neural Networks (PINNs) [7]. This approach represent a novel integration of deep learning with physical modeling, where the neural network is trained not only on data but also constrained by the physical constraints of the PDEs. The main contribution of this work include the development of a PINN framework specifically adapted for coupled biological PDEs, the implementation of appropriate boundary and initial conditions within the neural network architecture. In PINNs, both sequential methods and parallel approaches can be leveraged to accelerate and simplify the training and inference processes [8–10]. The validation of the PINN solutions against established numerical methods is shown in experimental results.

The remaining part of this paper is organized as follows: Section 2 presents the mathematical background of both the biological model and PINN approach. Section 3 shows our numerical results and validates our approach through comprehensive experiments. Finally, Section 4 concludes the paper.

2 Mathematical Background and Model Description

In this section, we present a description of the mathematical model governing tumor-induced angiogenesis and introduce our computational approach using Physics-Informed Neural Networks [1, 3]. The model captures the complex interactions between various biological components involved in the angiogenic process. Let $\Omega = [0, L_f]$ be the spatial domain representing the tissue region where angiogenesis occurs, and let $\mathcal{T} = [0, T_f]$ be the temporal domain of interest. We consider a coupled system of reaction-diffusion partial differential equations that model the evolution of $m = 4$ key variables: $C(x,t)$ representing the density of endothelial cells, $P(x,t)$ denoting the concentration of proteases, $I(x,t)$ describing the density of inhibitors, and $F(x,t)$ representing the density of the extracellular matrix [12]. The governing system of PDEs is given by:

$$\begin{aligned}
\partial_t C &= d_C \partial_{xx} C + \partial_x(f_I \partial_x I) - \partial_x(f_F \partial_x F) - \partial_x(f_T \partial_x T) + k_1 C(1-C) \\
\partial_t P &= d_P \partial_{xx} P - k_3 P I + k_4 T C + k_5 T - k_6 P \\
\partial_t I &= d_I \partial_{xx} I - k_3 P I \\
\partial_t F &= -k_2 P F \qquad\qquad (x,t) \in \Omega \times \mathcal{T},
\end{aligned} \tag{1}$$

subject to Neumann boundary conditions, reflecting the no-flux at the domain boundaries:

$$\partial_x u(0,t) = \partial_x u(L_f, t) = 0, \quad \forall t \in \mathcal{T}, \quad u \in \{C, P, I, F\}, \tag{2}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

and following initial conditions:

$$C(x,0) = \begin{cases} C_0, & 0 \le x \le a \\ 0, & a < x \le L_f \end{cases} \qquad (3)$$
$$P(x,0) = \xi_1, \quad I(x,0) = \xi_2, \quad F(x,0) = \xi_3$$

where ξ_i represent small random perturbations that account for biological variability and a defines the initial width of the endothelial cell distribution. The first equation describes the evolution of endothelial cells, incorporating diffusion ($d_C \partial_{xx} C$), chemotaxis toward inhibitors ($\partial_x(f_I \partial_x I)$), haptotaxis in response to the extracellular matrix ($\partial_x(f_F \partial_x F)$), response to tumor angiogenic factors ($\partial_x(f_T \partial_x T)$), and logistic growth ($k_1 C(1-C)$). The protease equation includes diffusion, degradation through interaction with inhibitors ($-k_3 PI$), production by endothelial cells in response to tumor factors ($k_4 TC$), direct production due to tumor factors ($k_5 T$), and natural decay ($-k_6 P$). The inhibitor equation accounts for diffusion and consumption through interaction with proteases, while the extracellular matrix equation represents its degradation by proteases, as analyzed in [13]. The taxis coefficients, that modulate cell movement, are defined as:

$$\begin{aligned} f_F &= \alpha_1 C \quad \text{(haptotaxis)}, \\ f_I &= \alpha_2 C \quad \text{(chemotaxis)}, \\ f_T &= \frac{\alpha_3 C}{1 + \alpha_4 T} \quad \text{(tumor angiogenic factor response)}. \end{aligned}$$

The tumor angiogenic factor distribution is modeled as a static spatial profile:

$$T(x) = \exp(-\epsilon^{-1}(L_f - x)^2)$$

representing a diffusive profile emanating from the tumor location. The physical and mass properties of the system are discussed in [14].

2.1 Physics-Informed Neural Network Framework

In order to numerically solve the system (1), we employ a Physics-Informed Neural Network approach [15, 16]. The mathematical formulation of a deep neural network structure defined by means of its components, mapping from the space-time domain $\Omega \times T$ to the solution space $\mathbb{R}^m$ through the transformation $\mathcal{N}_\theta : \Omega \times T \to \mathbb{R}^m$, where $(x,t) \mapsto \mathbf{u}(x,t) = (C(x,t), P(x,t), I(x,t), F(x,t))^T$. The architecture includes an input layer with $n_0 = 2$ neurons corresponding to the space-time coordinates $\mathbf{x}_0 = [x,t]^T$, followed by multiple hidden layers with n_l neurons each, and concluding with an output layer of $n_L = m = 4$ neurons representing the solution components. The full network architecture can be defined as:

$$\mathcal{N}_\theta(x,t) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\dots \sigma(\mathbf{W}_1 [x,t]^T + \mathbf{b}_1) \dots) + \mathbf{b}_{L-1}) + \mathbf{b}_L. \qquad (4)$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

The Eq. (4) implements the network propagation through the following layer-wise computations:

$$\mathbf{x}_1 = \sigma(\mathbf{W}_1\mathbf{x}_0 + \mathbf{b}_1), \quad \mathbf{x}_l = \sigma(\mathbf{W}_l\mathbf{x}_{l-1} + \mathbf{b}_l), \quad \mathbf{x}_L = \mathbf{W}_L\mathbf{x}_{L-1} + \mathbf{b}_L = \mathcal{N}_\theta(x,t) \tag{5}$$

where the first equation represents the input layer transformation, the middle term describes the hidden layers propagation for $l = 2, \ldots, L-1$, and the last one provides the final output layer computation that produces the neural network output $\mathcal{N}_\theta(x,t)$. The weight matrices and bias vectors have dimensions $\mathbf{W}_l \in \mathbb{R}^{n_{l+1}\times n_l}$ and $\mathbf{b}_l \in \mathbb{R}^{n_{l+1}}$ respectively. We define the total loss functional as:

$$\begin{aligned}\mathcal{J}(\theta) = &\underbrace{\int_{\Omega\times\mathcal{T}} \|\mathcal{L}[\mathcal{N}_\theta]\|_{L^2}^2 \, dx\, dt}_{\text{PDE Loss}} \\ &+ \underbrace{\lambda_1 \int_{\mathcal{T}} \left(\|\frac{\partial \mathcal{N}_\theta}{\partial x}(0,t)\|_{L^2}^2 + \|\frac{\partial \mathcal{N}_\theta}{\partial x}(L_f,t)\|_{L^2}^2 \right) dt}_{\text{Boundary Condition Loss}} \\ &+ \underbrace{\lambda_2 \int_{\Omega} \|\mathcal{N}_\theta(x,0) - \mathbf{u}_0(x)\|_{L^2}^2 dx}_{\text{Initial Condition Loss}}.\end{aligned} \tag{6}$$

The Eq. (6) is governed by: the nonlinear differential operator

$$\mathcal{L} : H^1(\Omega\times\mathcal{T}) \to L^2(\Omega\times\mathcal{T}),$$

which incorporates the physical laws; Neumann boundary conditions defined on $\partial\Omega = \{0, L_f\}$; and initial conditions $\mathbf{u}_0(x) \in H^1(\Omega)$. The hyperparameters $\lambda_1, \lambda_2 \in \mathbb{R}^+$ balance these interacting objectives in the $L^2(\Omega)$ topology, ensuring well-posedness of the variational formulation through appropriate Sobolev space regularization [7]. For simplicity, we discretize the loss functional using collocation points. Let $\{(x_i,t_i)\}_{i=1}^N \subset \Omega\times\mathcal{T}$ be a set of interior collocation points, $\{(0,t_j^b),(L_f,t_j^b)\}_{j=1}^{N_b}$ be boundary points, and $\{x_k\}_{k=1}^{N_0} \subset \Omega$ be initial points. The discrete counterpart of (6) yields to:

$$\begin{aligned}\mathcal{J}_h(\theta) = &\sum_{i=1}^{N} w_i \|\mathcal{L}[\mathcal{N}_\theta](x_i,t_i)\|_{L^2}^2 \\ &+ \lambda_1 \sum_{j=1}^{N_b} \left(\|\frac{\partial \mathcal{N}_\theta}{\partial x}(0,t_j^b)\|_{L^2}^2 + \|\frac{\partial \mathcal{N}_\theta}{\partial x}(L_f,t_j^b)\|_{L^2}^2 \right) \\ &+ \lambda_2 \sum_{k=1}^{N_0} \|\mathcal{N}_\theta(x_k,0) - \mathbf{u}_0(x_k)\|_{L^2}^2,\end{aligned} \tag{7}$$

where w_i are quadrature weights. The required derivatives are computed using automatic differentiation, which provides exact derivative calculations

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

through the neural network. The training process involves minimizing the discrete loss function using stochastic gradient descent, specifically the Adam optimizer with an adaptive learning rate strategy. Let us define the optimal solution $\mathbf{u}^* = (C^*, P^*, I^*, F^*)^T$ as the vector-valued function that satisfies PDEs system (1) with the given initial and boundary conditions, obtained through the PINN optimization process. More formally:

Definition 1 (Optimal Solution). *Let* $\mathcal{V} = \{\mathbf{u} \in [H^1(\Omega \times \mathcal{T})]^m : \partial_x \mathbf{u}|_{\partial\Omega} = 0, \|\mathbf{u}\|_{H^1} < \infty\}$ *be the space of admissible solutions. The optimal solution* $\mathbf{u}^*$ *is defined as:*

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathcal{V}} \mathcal{J}_h(\mathbf{u})$$

where $\mathcal{J}_h(\mathbf{u})$ *is the discrete functional:*

$$\mathcal{J}_h(\mathbf{u}) = \sum_{j=1}^{N} \|\mathcal{L}[\mathbf{u}]\|^2_{L^2(\Omega_j \times \mathcal{T}_j)} + BC_h[\mathbf{u}] + IC_h[\mathbf{u}]$$

with discretized boundary and initial conditions terms:

$$BC_h[\mathbf{u}] = \|\partial_x \mathbf{u}(0,\cdot)\|^2_{L^2(\mathcal{T}_h)} + \|\partial_x \mathbf{u}(L_f,\cdot)\|^2_{L^2(\mathcal{T}_h)}$$
$$IC_h[\mathbf{u}] = \|\mathbf{u}(\cdot,0) - \mathbf{u}_0\|^2_{L^2(\Omega_h)}.$$

where $\Omega_h = \{x_i\}_{i=1}^{N_x}$ *and* $\mathcal{T}_h = \{t_j\}_{j=1}^{N_t}$ *represent the spatial and time discretization points respectively, with* N_x *and* N_t *denoting the number of collocation points in each domain.*

The complete PINN-based solution process for the angiogenesis model (1) is compactly re-written as follows:

$$\mathbf{u}^* = \arg\min_{\theta\in\Theta} \mathcal{J}_h(\theta)$$
$$= \mathcal{N}_{\theta^*}(x,t) : \begin{cases} \theta^* = \arg\min_{\theta\in\Theta} \left\{ \|\mathcal{L}[\mathbf{u}]\|^2_{L^2(\Omega\times\mathcal{T})} + BC[\mathbf{u}] + IC[\mathbf{u}] \right\} \\ \text{subject to:} \\ \mathcal{L}[\mathbf{u}] = 0 \quad \text{in} \quad \Omega \times \mathcal{T} \\ \partial_x \mathbf{u} = 0 \quad \text{on} \quad \partial\Omega \times \mathcal{T} \\ \mathbf{u}(x,0) = \mathbf{u}_0(x) \quad \text{in} \quad \Omega \end{cases}$$

where θ^* are the optimal network parameters deriving from the minimization of the discrete loss functional $\mathcal{J}_h$. This formulation provides an effective framework for approximating the solution of the PDE system, as will be shown by numerical experiments in Section 3. The previous procedures of the framework can be summarized in the following Algorithm:

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

Algorithm 1 PINN-based Solution for Angiogenesis Model

Input: Domain parameters Ω, $\mathcal{T}$, model parameters $\{d_i, k_i, \alpha_i\}$
Numerical Model:
  Select collocation points $\{(x_i, t_i)\}_{i=1}^{N}$, $\{(0, t_j^b), (L_f, t_j^b)\}_{j=1}^{N_b}$, $\{x_k\}_{k=1}^{N_0}$
  Define neural architecture $\mathcal{N}_\theta$ and loss function $\mathcal{J}_h$
Training Process:
  Initialize $\theta^{(0)}$ randomly
while not converged **do**
    Compute PDE residuals $\mathcal{L}[\mathcal{N}_\theta]$
    Evaluate boundary and initial conditions
    Update θ via Adam optimizer with adaptive weights
end while
Output: Trained network $\mathcal{N}_{\theta^*}$ approximating solution $\mathbf{u}$

3 Results and discussion

In this section we show several experiments that confirm the reliability of proposed approach. We performed the tests on a super-computer machine, offered by CINECA [17], with following technical specifications: 1 CPU Intel Xeon 8358 32 cores, 2.6 GHz (8×64) GB RAM , NVIDIA A100-SXM-64GB. The results are achieved by a mean on 10 executions of the Algorithm 1. We consider the spatial domain $\Omega = [0, 1]$ and temporal domain $T = [0, 50]$. In order to avoid numerical instabilities and, according to [14], the model parameters are set as follows: diffusion coefficients: $d_C = d_P = d_I = 10^{-3}$, reaction rates: $k_i = 0.1$ for $i = 1, \ldots, 6$, and, taxis coefficients: $\alpha_i = 0.1$ for $i = 1, \ldots, 4$. The implementation of Algorithm 1 has been performed by using PyTorch[1].

Test 1. Numerical validation. In order to evaluate the accuracy of our PINN solutions, we compare them with reference solutions ($\mathbf{u}_{\text{FE}}$) obtained using a schema, defined in [14], at specific time points $t \in \{5, 10, 15, 25, 50\}$. Table 1 shows the relative L^2 errors computed as:

$$E(t) = \frac{\|\mathcal{N}_{\theta^*}(\cdot, t) - \mathbf{u}_{\text{FE}}(\cdot, t)\|_{L^2(\Omega)}}{\|\mathbf{u}_{\text{FE}}(\cdot, t)\|_{L^2(\Omega)}}. \tag{8}$$

The relative L^2 errors in Table 1 highlight distinct patterns across components

Table 1. Relative L^2 errors at different time points

Component	$t = 5$	$t = 10$	$t = 15$	$t = 25$	$t = 50$
C	3.82e-3	5.67e-3	8.91e-3	1.24e-2	1.89e-2
P	2.13e-3	2.89e-3	3.45e-3	4.12e-3	5.34e-3
I	2.45e-3	2.98e-3	3.56e-3	4.23e-3	5.12e-3
F	1.92e-3	2.34e-3	2.87e-3	3.45e-3	4.23e-3

and time points. The component (C) shows the highest errors due to its complex

[1] https://pytorch.org

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

nonlinear dynamics. The protease concentration, (P), and inhibitor density (I) demonstrate more stable behavior. The extracellular matrix density (F) exhibits the most stable error progression. The hierarchical error pattern ($C > P \approx I > F$) correlates with each component of equation complexity, while maintaining overall accuracy below 2% throughout the simulation, validating the robustness of our PINN approach.

Test 2. Computational performance analysis. We conducted a comparison between CPU and GPU execution times.

Table 2. Computational performance comparison between CPU and GPU implementations

Dataset size	CPU Time (s)	GPU Time (s)
5625	27.71	10.70
10000	52.64	10.96
15625	75.34	17.69

Table 2 presents the execution times for both CPU and GPU implementations across different dataset sizes. The results exhibit interesting patterns in the computational performance. The significant performance gain appears with larger datasets, where the GPU implementation achieves a remarkable 9.80× speedup over the CPU counterpart.

4 Conclusions

In this work, we have presented a Physics-Informed Neural Network approach for solving a complex biological system modeling tumor-induced angiogenesis. The presented framework integrates deep learning capabilities with physical constraints through an optimized loss function that incorporates PDE residuals, boundary, and initial conditions. The numerical validation demonstrated the reliability of our approach, with relative L^2 errors consistently below 2% across all system components, even for extended time horizons. The computational performance analysis revealed significant speedup (up to 9.80×) when utilizing GPU acceleration, particularly beneficial for larger datasets. Future developments could include extending this framework to handle multi-scale environmental processes, incorporating data assimilation techniques for real-time environmental monitoring, and adapting the architecture for coupled atmosphere-ocean systems.

Acknowledgment

This work is funded and supported by project “DSTE372– Risoluzione numerica di problemi differenziali anomali (RNPDA)". P. De Luca and L. Marcellino are member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2

References

1. Karniadakis, G. E., et al. (2024). Physics-informed machine learning. Nature Reviews Physics, 1-25.
2. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
3. Wang, S., Wang, H., & Perdikaris, P. (2023). Learning the effective dynamics of complex systems using physics-informed neural networks. Nature Computational Science, 3(1), 27-37.
4. Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2022). Machine learning for fluid dynamics. Annual Review of Fluid Mechanics, 54, 315-346.
5. Valentino, C., Pagano, G., Conte, D., Paternoster, B., Colace, F., & Casillo, M. (2025). Step-by-step time discrete Physics-Informed Neural Networks with application to a sustainability PDE model. Mathematics and Computers in Simulation, 230, 541-558.
6. Colace, F., Conte, D., Pagano, G., Paternoster, B., & Valentino, C. (2024). Physics-informed neural networks for a Lithium-ion batteries model: A case of study. Advances in Computational Science and Engineering, 2(4), 354-367.
7. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686-707.
8. Pan, X., & Xiao, D. (2024). Domain decomposition for physics-data combined neural network based parametric reduced order modelling. Journal of Computational Physics, 519, 113452.
9. De Luca, P., Galletti, A., Ghehsareh, H. R., Marcellino, L., & Raei, M. (2020). A GPU-CUDA framework for solving a two-dimensional inverse anomalous diffusion problem. In Parallel Computing: Technology Trends (pp. 311-320). IOS Press.
10. Fiscale, S., De Luca, P., Inno, L., Marcellino, L., Galletti, A., Rotundi, A., ... & Quintana, E. (2021, June). A GPU algorithm for outliers detection in TESS light curves. In International Conference on Computational Science (pp. 420-432). Cham: Springer International Publishing.
11. De Luca, P., Galletti, A., & Marcellino, L. (2019, November). A Gaussian recursive filter parallel implementation with overlapping. In 2019 15th international conference on signal-image technology & internet-based systems (SITIS) (pp. 641-648). IEEE.
12. Kevrekidis, P.G., Whitaker, N., Good, D.J., Towards a reduced model for angiogenesis: a hybrid approach, Math. Comput. Model. 41 (2005) 987–996.
13. Anderson, A. R., & Chaplain, M. A. J. (2005). Mathematical modeling of tumor-induced angiogenesis. Annual Review of Biomedical Engineering, 7, 233-257.
14. De Luca, P.; Marcellino, L. Conservation Law Analysis in Numerical Schema for a Tumor Angiogenesis PDE System. Mathematics 2025, 13, 28. https://doi.org/10.3390/math13010028
15. Kissas, G., et al. (2023). Robust training of physics-informed neural networks. Journal of Computational Physics, 474, 111759.
16. Jagtap, A. D., & Karniadakis, G. E. (2023). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Communications in Computational Physics, 33(2), 477-514.
17. CINECA Supercomputing Centre, SuperComputing Applications and Innovation Department. (2024). "LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI applications.", Journal of large-scale research facilities, 8, A186.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97567-7_2