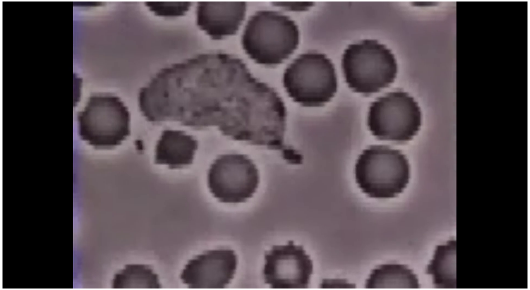
First Insights into PINNs for Accelerating Immune Response Models

Thiago E. Fernandes, Marcelo Lobosco, Rodrigo W. dos Santos

Graduate Program in Computational Modelling
Federal University of Juiz de Fora

July 25, 2024

Mathematical model



The mathematical model used to describe the pathophysiology of edema formation was proposed in a previous work [1, 2]. This model includes the inflammatory part which describes the interaction between a pathogen and the human immune system.

The pathogen is modelled by:

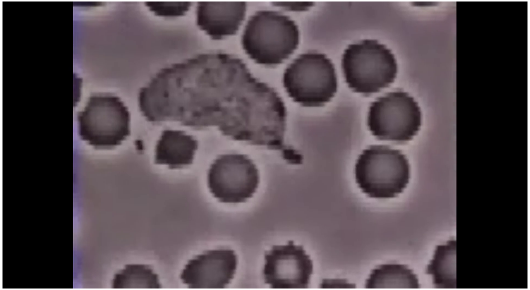
$$\begin{cases} \dfrac{d(\phi_f C_b)}{dt} = -r_b + q_b, \ t \in (0, 10] \\ C_b(0) = \delta_b, \end{cases} \tag{1}$$

where

$$q_b = c_b C_b \tag{2}$$

$$r_b = \lambda_{nb} C_n C_b \tag{3}$$

Mathematical model



The leukocyte differential model is represented by:

$$\begin{cases} \dfrac{d(\phi_f C_n)}{dt} = -r_n + q_n, \; t \in (0, 10] \\ C_b(0) = 0 \end{cases} \tag{4}$$

where

$$q_n = \gamma_n C_b (C_{n,max} - C_n) \tag{5}$$

$$r_n = \lambda_{bn} C_n C_b + \mu_n C_n \tag{6}$$

Mathematical model

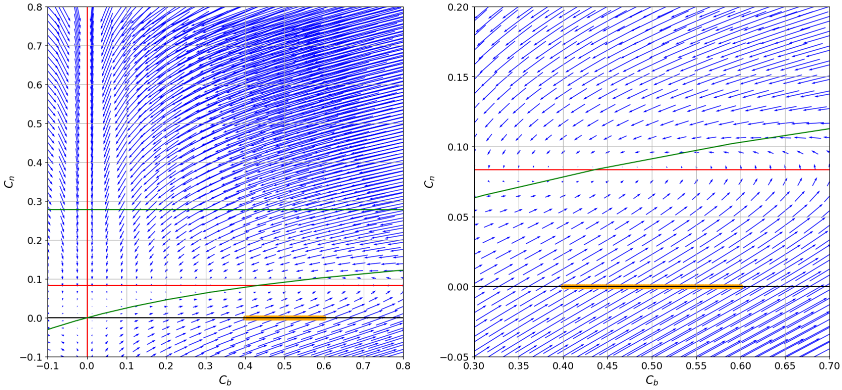


Figure 1: Phase plane for the system of equations.

Physics-Informed Neural Networks (PINN)

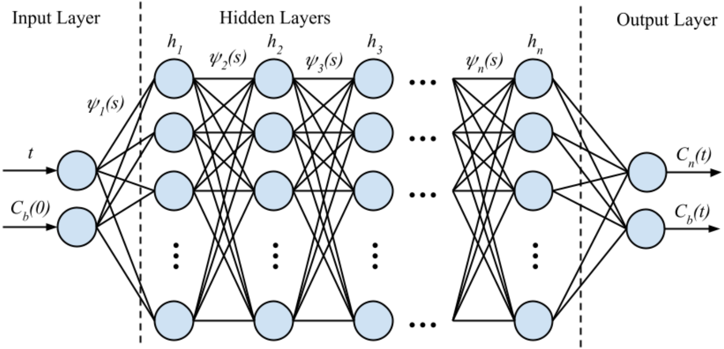


Figure 2: Neural Networks architecture.

Physics-Informed Neural Networks (PINN)

In alignment with the original formulation of Raissi et al. [3], we generally consider PDEs taking the form

$$\frac{\partial u}{\partial t} + \Phi[u] = 0,\ t \in [0,T],\ x \in \Omega \tag{7}$$

subject to the initial and boundary conditions

$$u(0,x) = g(x),\ x \in \Omega \tag{8}$$

$$\beta[u] = 0,\ t \in [0,T],\ x \in \partial\Omega \tag{9}$$

This formulation enables us to define the PDE residuals as follows:

$$\zeta_\theta(t,x) = \frac{\partial u_\theta}{\partial t}(t,x) + \Phi[u_\theta](t,x) \tag{10}$$

PINN architecture grid-search

Consequently, a physics-informed model is trained by minimising the following composite loss function:

$$L(\theta) = L_{ic}(\theta) + L_{bc}(\theta) + L_r(\theta) + L_{dt}(\theta) \tag{11}$$

$$L_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \sqrt{(u_\theta(0, x_{ic}^i) - g(x_{ic}^i))^2} \tag{12}$$

$$L_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \sqrt{\beta[u_\theta](t_{bc}^i, x_{bc}^i)^2} \tag{13}$$

$$L_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \sqrt{(\zeta_\theta(t_r^i, x_r^i)^2} \tag{14}$$

$$L_{dt}(\theta) = \frac{1}{N_{dt}} \sum_{i=1}^{N_{dt}} \sqrt{(u_\theta(t_{dt}^i, x_{dt}^i) - u(t_{dt}^i, x_{dt}^i))^2} \tag{15}$$

PINN architecture grid-search

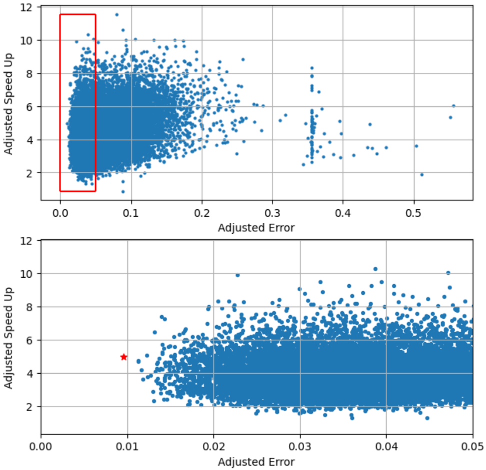


Figure 3: Adjusted error and speed up of the tested architectures.

The adjusted error and speedup are defined respectively by Equations 16 and 17.

$$E_{aj} = RMSE + RSE_{max} \tag{16}$$

$$S_{aj} = \langle S \rangle_{av} - \sigma_s \tag{17}$$

where *RMSE* stands for Root Mean Squared Error and *S* is the speed up.

PINN architecture grid-search

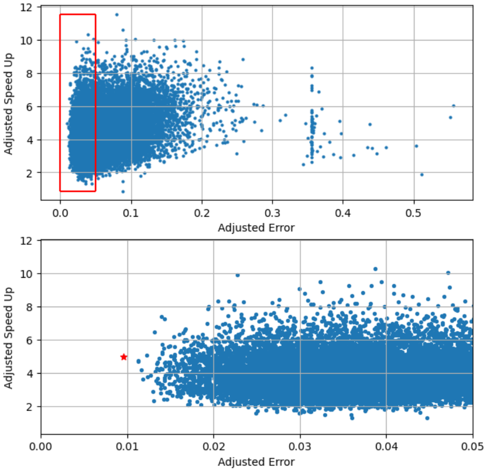


Figure 4: Adjusted error and speed up of the tested architectures.

Considering the 22465 architectures studied, the chosen one is composed by 4 hidden layers with 16, 8, 32, and 16 neurons respectively. Considering the PyTorch documentation, [4], the activation functions are defined as:

$$\psi_1 = \frac{sinh(x)}{cosh(x)} \tag{18}$$

$$\psi_2 = x \cdot \sigma(x) \tag{19}$$

$$\psi_3 = \psi_4 = max(0, x) \tag{20}$$

where x represents the values from the previous layers and $\sigma(x)$ is the logistic sigmoid function.

PINN and NN comparison

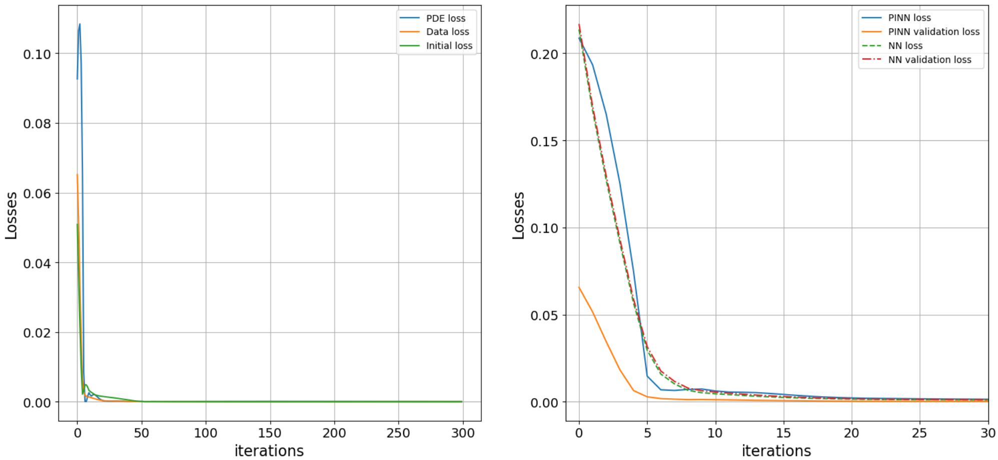


Figure 5: Learning curves for the NN and PINN training.

PINN and NN comparison

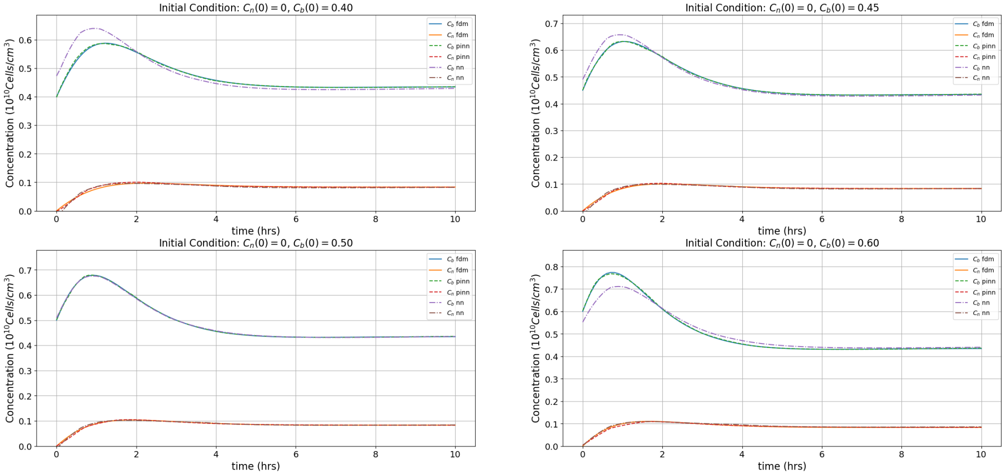


Figure 6: Concentration curves for the different computational models

PINN and NN comparison

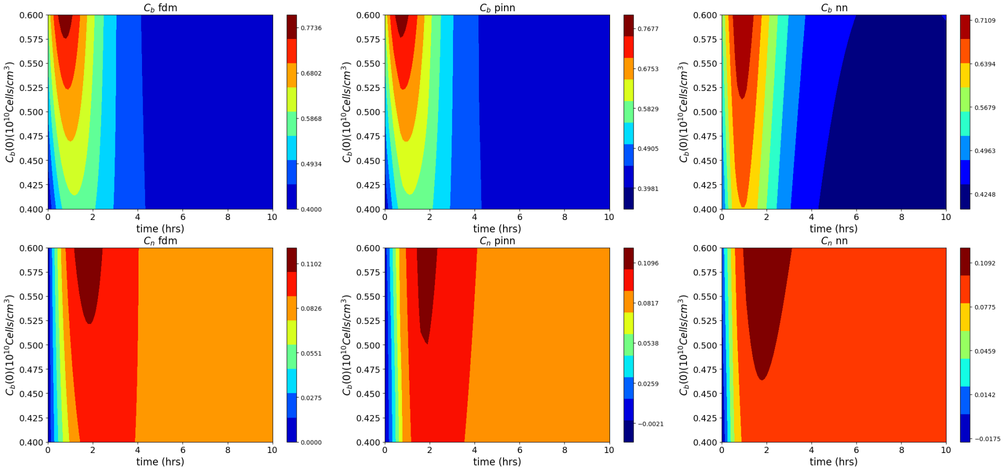


Figure 7: Concentration heat-map for the different computational models

PINN and NN comparison

Table 1 presents the results for the PINN and NN.

Table 1: Experimental results of PINN constrains implementation

Model	RMSE	RSE_{max}	Mean speed up	Speed up Standard Deviation
PINN	0.001530657	0.0080879815	5.93	0.97
NN	0.11618373	0.14936338		

[1] R. F. Reis, R. W. dos Santos, B. M. Rocha, and M. Lobosco, "On the mathematical modeling of inflammatory edema formation," *Computers & Mathematics with Applications*, vol. 78, no. 9, pp. 2994–3006, 2019.

[2] R. F. Reis, J. L. Fernandes, T. R. Schmal, B. M. Rocha, R. W. Dos Santos, and M. Lobosco, "A personalized computational model of edema formation in myocarditis based on long-axis biventricular mri images," *BMC bioinformatics*, vol. 20, pp. 1–11, 2019.

[3] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[4] *PyTorch documentation*, PyTorch Foundation, 2023. [Online]. Available: https://pytorch.org/docs/stable/index.html

Thank You