1 TITLE

1) The effects of nondimentionalization in Physics-Informed Neural Networks applied to the Kinetics of Biological Reactions: A case study with Lactobacillus casei

2) Strategies of Physics-Informed Neural Networks model optimization applied to Biochemical Reactions: A case study with *Lactobacillus casei*

2 ABSTRACT

Many biological reactions have mathematical models available in scientific literature, but don’t have enough data to be simulated by a Neural Network achieving good enough accuracy of generalization. PINN (Physics-Informed Neural Network) is a relatively new technique capable of simulating physics-based systems using Neural Networks using mathematical models equations to generate the necessary ammount of data for Neural Network optimization. Biological reactions pose a new challenge for PINNs because bioreactions kinetics may include a system of equations for multiple reactions ocurrying at the same time, show multiple interdependence between the derivatives of substances of interest, and complex inhibitory terms that don’t support certain values of variables. This work employs PINNs to simulate the reaction kinetics for the production of LA from Lactose by *Lactobacillus casei* in batch, fed-batch and CSTR (in both transient and stationary states) reactors. Lactic Acid (LA) was chosen for being a molecule of great industrial and economical interest with plenty of experimental data and mathematical.models available. The solution is compared with a numeric method (Euler) and experimental data. PINNs were able to simulate with great accuracy the profiles of substrate, product and biomass concentrations over time.

3 INTRODUCTION

Lactic acid (LA) is a molecule of great industrial and economical interest. LA and its derivatives are used in pharmaceutical, cosmetics and food industries (NANCIB et al., 2015). Since it is an organic molecule, occurs naturally is and perceived as “green” or environmentally friendly by many consumers (DE OLIVEIRA et al., 2021), its perceived value comes not only from the molecule chemical or technological features, but also from a marketing standing point. Another point that drove interest in LA growing since the beginning of the 21th century was its application as a raw material for the production of PLA (poly-lactic acid), an environmentally friendly alternative to plastics and that has found many applications (KOMESU; MACIEL; FILHO, 2017; LI et al., 2006). The industry of Lactic Acid will grow to 160 kt by 2025 (LÓPEZ-GÓMEZ et al., 2019), with a revenue forecast of almost USD 9 billion (DIN et al., 2021).

LA has actually two optical active forms, which can vary greatly in application. The L(+) form is prefered for some cosmetical, food and pharmaceutical applications since D(-)-Lactic Acid can be a harmful enantiomer ((POHANKA, 2020)). The chemical synthesis yields a racemic mixture (BÜYÜKKILECI; HARSA, 2004), while many bacteria can produce mainly one of the two enantiomers depending on the process conditions. The great economical and scientific interest in LA is demonstrated in Academia, with many studies conducted on these kind of microorganisms and their LA production using different carbon sources were evaluated (ALTAF; NAVEENA; REDDY, 2007, 2007; ALTIOK; TOKATLI; HARSA, 2006; DEY; PAL, 2013; REZVANI; ARDESTANI; NAJAFPOUR, 2017; THAKUR; PANESAR; SAINI, 2019; VERA-PEÑA; HERNÁNDEZ-GARCÍA; VALENCIA-GARCÍA, 2022). Thus, the large amount of scientific literature available and relevance made Lactic Acid production by *Lactobacillus casei* (ALTIOK; TOKATLI; HARSA, 2006) the ideal case study for the application of Physics-Informed Neural Networks in biological reactions and reactor simulation.

Physics-Informed Neural Networks (PINN) are an approach to employ deep neural networks capability of universal function approximators to solve complex numerical problems, such as stochastic and high order partial differential equations

(PDEs) (HORNIK; STINCHCOMBE; WHITE, 1989; RAISSI; PERDIKARIS; KARNIADAKIS, 2019; SANTANA et al., 2022). This technique can be applied to many cases where the numeric solution is highly complex, but also for simpler cases where it provides a robust framework for simulating system with well defined mathematical models. One of the greatest advantages of PINNs is the possibility to use the mathematical models themselves instead of raw experimental data, so it is possible to work with *small data*. Biological reactions kinetics pose a natural challenge for mathematical solution because, while many models may have a relatively simple mathematical description, this descriptions often involves multiple derivatives referencing each other, limiting the methods of solving it, or even requiring greater computer processing power and more sophisticated integration methods.

This work aims to determine the relevance of the deepness or wideness of the neural network chosen and the influence of the nondimensionalization of equations in performance and computational cost required by the system to achieve enough precision. This is done while evaluating the application of PINNs in biological reaction kinetics, using as a case study the production of Lactic Acid by *Lactobacillus casei* as described in the literature (ALTIOK; TOKATLI; HARSA, 2006) in CSTR (Countinuous Stirred Tank), batch and fed-bacth reactors.

4 MATERIALS AND METHODS

4.1 Nondimensionalization

Nondimensionalization is a process in which variables are converted to nondimensional variables using nondimensional factors, in this work called nondimensional scalers. These scalers can substantially improve or worsen the error of the system of equations being solved, thus are an important for giving insights on how to improve the model (ALHAMA MANTECA; SOTO MECA; ALHAMA, 2012). The reduced dimension equation may need less computational resources, produce better results or be unable to attain a feasible solution. Since PINNs can be very sensitive to the value of the derivatives they are optimizing, the equations were nondimensionalized to evaluate the procedure impact on final results

We define each variable, represented by N, as an nondimensional variable. The variable is equals to the nondimensional variable multiplied by a coefficient:

$$N = N_A * N_S \tag{1}$$

where N is the variable itself, N_A is the nondimensional variable and N_S is the nondimensional scaler. The use of the subscript A represents the nondimensional variable version of N, and S represents the nondimensional scaler of N.

4.2 Reaction Kinetics Mathematical Model

The model of the production of Lactic Acid by *Lactobacillus casei* using *whey lactose* was proposed and validated by (ALTIOK; TOKATLI; HARSA, 2006) using experimental data of a batch reactor. The second experiment variant (with starting lactose concentration 21.4g/L and duration of 9 hours) is used in this work. *L. casei* prouction of Lactic Acid is known to be controlled by inhibitory effects and can be represented by the classical Monod equation adjusted to represent product inhibition effect (represented by (1 – P/P_M)) and biomass inhibition effect (represented by (1 – X/X_M)), where X is the biomass concentration, P is the product concentration, P_M is the inhibitory product concentration, and X_M is the inhibitory biomass concentration. Thus, the biomass reaction rate model includes product, biomass and substrate inhibitory effects. The biomass concentration derivative over time is given by:

$$r_X=\frac{\mu_{max}S}{K_S+S}X\left(1-\frac{X}{X_M}\right)^f\left(1-\frac{P}{P_M}\right)^h \tag{2}$$

where r_X is the reaction rate of biomass, μ_{max} is the maximum possible growth rate, S is the substrate (whey lactose) concentration, K_S is the Monod constant and f and h are factors that represent the toxicity or inhibitory potential adjustments.

The product formation rate is given by Luedeking-Piret kinetics, depending on the growth or decrease of biomass linearly (represented by dX/dt) and also on the biomass concentration itself:

$$r_P=\alpha r_X+\beta X \tag{3}$$

where α is the growth-associated product formation coefficient and β is the non-growth associated product formation coefficient.

The substrate consumption kinetics is given by a relationship including the substrate converted to product and the substrate used for biomass maintenance:

$$r_S=\frac{-1}{Y_{PS}}r_P-m_s X \tag{4}$$

where Y_{PS} is the product yield coefficient and m_s is the maintenance coefficient.

Equations 2 - 4 can be nondimensionalized as:

$$\frac{X_S}{t_S}r_{X_A}=\frac{\mu_{max}S_A S_S}{K_S+S_A S_S}X\left(1-\frac{X_A X_S}{X_M}\right)^f\left(1-\frac{P_A P_S}{P_M}\right)^h \tag{5}$$

$$\frac{P_S}{t_S} r_{P_A} = \alpha \frac{X_S}{t_S} r_{X_A} + \beta X_A X_S \qquad (6)$$

$$\frac{S_S}{t_S} r_{S_A} = \frac{-1}{Y_{PS}} \frac{P_S}{t_s} r_{P_A} - m_s X_A X_S \qquad (7)$$

4.3 Reactor Mathematical Model

The volume of liquid in the reactor is given by:

$$\frac{dV}{dt} = f_{\text{in}} - f_{\text{out}} \qquad (8)$$

Which can be nondimensionalized as:

$$\frac{V_S}{t_S} \frac{dV_A}{dt_A} = f_{\text{in}} - f_{\text{out}} \qquad (9)$$

The nondimensional concentration of a generic substance or biomass, N, can be given by:

$$V_A V_S \frac{N_S}{t_S} \frac{dN_A}{dt_A} = V_A V_S \frac{N_S}{t_S} r_{N_A} + f_{\text{in}} N_{\text{in}} - f_{\text{out}} N_A N_S \qquad (10)$$

where N is the concentration of each substance or biomass inside the reactor, V is the volume of liquid inside the reactor (L), t is the time (h), f_{in} is the inlet flow rate (L/h) in the reactor, N_{in} is the concentration of N in the inlet flow, f_{out} is the outlet flow rate (L/h) of the reactor, and N_{out} is the concentration of N in the outlet flow. The subscripts S and A were defined in section 4.1.

This model can be used for representing both batch, fed-batch and CSTR models. For the batch case, both f_{in} and f_{out} are zero. For the fed-batch case, only f_{out} is zero. For CSTR, f_{in} and f_{out} are greater than zero and $f_{in} = f_{out}$ at steady state.

4.4 Simulation using Physics Informed Neural networks

Physics-Informed Neural Network (PINN) was introduced to the scientific community in 2019 (RAISSI; PERDIKARIS; KARNIADAKIS, 2019). One of the most important points of the technique is to allow scientists to simulate physical, chemical or biological systems using differential equations as a source of data source for the optimization of Neural Networks (NN) created specifically to solve those problems. This is possible because NN are intrinsically universal approximators (SANTANA et al., 2022).

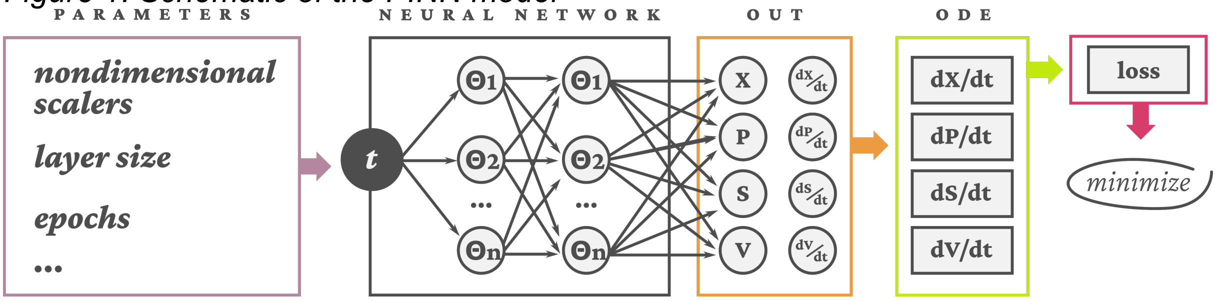
The overall loss function is defined as the weighted sum of the L^2 residuals (LU et al., 2021):

$$L=\sum w_N \left\| \left(\left.\frac{dN}{dt}\right|_{calculated} - \left.\frac{dN}{dt}\right|_{predicted} \right) \right\|^2 \tag{11}$$

where N represents X, S, P or V, and w_N is the weight of loss for N.

To test many different PINN configurations, a custom Grid Search and repetition loop was created. The PINN evaluation (without the repetition loop) scheme adopted in shown in Figure 1. Both hyperparameters and equations parameters are fixed before each loop. Then, the Neural Network is fed with the data from the equation system. The outputs (labeled as OUT in Figure 1) are the variables of interest (X, P, S and V) and their respective derivatives according to time. Then, the loss function is calculated comparing the given derivatives at step “n” compared with what their actual value should be. This can be achieved simply by making the difference (subtraction) of the derivative calculated (as show in equations 5-10) and the ones predicted by the NN.

Figure 1: Schematic of the PINN model



In the original paper that introduced PINN (RAISSI; PERDIKARIS; KARNIADAKIS, 2019) questions suchs as how deep and wide the neural network must be to appropriately represent the mathematical models, the relevance of normalization and/or nondimensionalization and how the weights of loss functions can impact on the model loss and accuracy were raised. Because this subject still being debated and more studies need to be carried out (SANTANA et al., 2022), these points will be evaluated in the next sections. This work compares traditional numeric methods (Euler Forward Method), experimental (ALTIOK; TOKATLI; HARSA, 2006) and PINN generated solutions for the simulation of biological reactions, in specific the production of LA by *L. casei*, and strategies to improve performance and reduce loss in different reactors regiments (batch, fed-batch and CSTR). Therefore, is out of the scope of this work to discuss meticulously the implementation of PINNs.

4.5 Simulation configurations

The main questions to be answered are:

- Is it possible to approximate simple, ordinary differential equations (ODEs), with PINNs using lower numbers of neurons and layers? The simple Burgers equation required at least 20 neurons per layer and a few layers in another study (LU et al., 2021);
- Do the weights of the loss function of each variable impact significantly on the accuracy and number of steps required for acceptable results?
- What impacts does the independent variable (time) nondimensionalization cause to the loss?
- How does the dependent variables (X, P, S, V) nondimensionalization affect the system? Are some of them more relevant than the others?
- Since the same mathematical model of reactor is being used for the three cases (batch, fed-batch and CSTR), can the best configuration (group of parameters and hyperparameters, such as number of neurons per layer and

nondimensional scalers) for the batch reactor also represent accurately the other two?

- Is it possible to find a pattern of relationship between hyperparameters, loss weights and nondimensionalization scalers in order to improve the traditional Random Grid Search, replacing it with a more rational and predictable approach?

Batch reactor simulation starts with 5 L of solution and Fed-batch and CSTR starts with 1 L, all with initial concentrations of Biomass, LA and whey lactose as the same used in the experiment (case 2 of (ALTIOK; TOKATLI; HARSA, 2006)). CSTR is simulated on a greater time period (96 h) while batch and fed-batch are both simulated for 10.6 h (slightly above the experimental time). The batch reactor, by definition, has closed inlets and outlets. The fed-batch reactor volume is allowed to increase indefinitely. CSTR starts in transient state and is expected to converge to stationary state after some time, with its volume being regulated by a function in order to open the outlet as the volume capacity (5L) is approached. The inlet flow of fed-batch is set to 2L/h, while the inlet flow of CSTR is set to 1 L/h. The outlet function for the CSTR is given by:

$$f_{\mathrm{in}} * (V/V_{max})^7 \tag{12}$$

where V_{max} is the maximum liquid volume in the reactor, set as the same of the maximum volume of the reactor used in the original experiment (5 L).

All simulations were run with all weights of loss function (w_V, w_X, w_P, w_S) = 1, all nondimensionalization scalers (t_S, X_S, P_S, S_S, V_S) = 1, number of training points = 800, number of testing points = 1000, layer size = 22 neurons in 3 layers, epochs (iterations of Adam algorithm (KINGMA; BA, 2017)) = 30,000, learning rate = 1×10^{-3}, except when explicitly stated otherwise. The PINN model was solved using DeepXDE (LU et al., 2021), and hyperbolic tangent activation was used to to enhance convergence speed.

5 RESULTS

In order to try to find patterns between many of the hyperparameters and process parameters, tests were conducted in parallel instead of conducting a traditional Random Grid Search, were a myriad of hyperparameters is set, every possible combination is iterated and the result with the least loss chosen. Table 1 summaries the tests.

Table 1 - Conclusions for each test

Test Name	Objectives	Conclusion
t_S	Evaluate the effect of nondimensionalization scaler of time (t_S)	In general, the greater t_S the greater the loss
non_dim	Evaluate the effect of nondimensionalization scalers of X_S, P_S, S_S, and V_S	Batch, fed-batch and CSTR optimizations vary greatly, and there is no general rule
layer_size	1) Evaluate the effect of layer size in performance 2) Check if it is possible to rationalize the ratio neuron/layers (N/L) in function of the loss	1) Complex NNs (more than 4 layers or more than 60 neurons per layer) took much more epochs to minimize loss, but it was still not sufficient for achieving acceptable simulation results, while some simple NN (3 layers with 32 neurons and 4 layers with 22 neurons) were generally acceptable for batch reactor simulation; 2) It was not possible to find any clear relationship between loss, N/L and total number of neurons
weight	Evaluate the impact of loss weight in all cases studied	There is not set of weights that reduce the loss for all cases. Each case has a specific set of weight that minimizes loss. CSTR
multiple_config	Evaluate if the same configuration can appropriately represent batch, fed-batch and CSTR	Almost all configurations tested can produce aggreable results simulating the batch reactor, and almost none can represent fed-batch and CSTR cases

pinn_numeric_xp	Validate the proposed batch model, comparing with the solution using Euler Method and Experimental Data	The model was successfully validated

Nondimensionalization effects can greatly improve CSTR and fed-batch performance, but easily made batch reactor simulation yield greater losses. The variation of t_S did not only not improve the simulation of batch reactor error, but also resulted in greater loss (Figure 2 - a). From all the values tested for the independent variable (t) nondimensionalization, the best t_S was found to be one. In other words, the best model is found to be the one which does not nondimensionalize the time. The nondimensionalization of time, as a rule, increase the loss of the model and, in many cases, made the NN stagnate at points next to loss = 1. Thus, the nondimensionalization of time was ruled out as noneffective. It must be noted, however, that the 6 cases tested for time were greater or equal 1. Since the loss increased with the increase of t_S, values of $t_S <=1$ may improve model performance in systems with multiple dependency (P and S depend directly on X, and X, P and S depend on V), such as the presented.

The nondimensionalization (test “non_dim”) of X, P, S and V was conducted in sequence, with 6 different combinations. The layer size was tested as 5 layers with 32 neurons and 3 layers with 22 neurons. CSTR loss was reduced greatly using the smaller layer. The best cases for CSTR were n1 and n6. The best cases for fed-batch was n4, with loss < 10^{-2}. Despite of that, none of these models were capable of representing adequately any of the reactors, even with more than 80,000 epochs.

Table 2 - Test Parameters

Test Name	**Default Parameters**	**Subtest Identifier**	**Parameters**
t_S	3 layers with 22 neurons 30,000 epochs	case t_1	t_S = experiment time (9 h)
		case t_2	$t_S = 1 / (\mu_{max} * S_o / (K_S + S_o)$
		case t_3	$t_S = \alpha * S_o * (K_S + S_o) / \mu_{max}$
		case t_4	$t_S = (1/Y_{PS}) * \alpha * (K_S + S_o) / \mu_{max}$
		case t_5	$t_S = 1$
		case t_6	$t_S = 1 / \mu_{max}$
non_dim	3 layers with 22	n1	$V_S = 1$, $X_S = 1$, $P_S = 1$, $S_S = 1$

	neurons 80,000 epochs	n2	$V_S = f_{in}*1h$, $X_S = 1$, $P_S = 1$, $S_S = 1$
		n3	$V_S = V_{max}$ (5 L), $X_S = X_o$, $P_S = P_o$, $S_S = S_o$
		n4	$V_S = V_{max}$ (5 L), $X_S = X_M$, $P_S = P_M$, $S_S = S_o$
		n5	$V_S = 1$, $X_S = 1$, $P_S = P_M$, $S_S = S_o$
		n6	$V_S = V_{max}$ (5 L), $X_S = X_M$, $P_S = 1$, $S_S = 1$
layer_size	$t_s = 1$ 30,000 epochs	t_lay1	6 layers with 8 neurons
		t_lay2	10 layers with 8 neurons
		t_lay3	14 layers with 8 neurons
		t_lay4	3 layers with 16 neurons
		t_lay5	4 layers with 16 neurons
		t_lay6	6 layers with 16 neurons
		t_lay7	3 layers with 22 neurons
		t_lay8	4 layers with 22 neurons
		t_lay9	5 layers with 22 neurons
		t_lay10	3 layers with 32 neurons
		t_lay11	4 layers with 32 neurons
		t_lay12	5 layers with 32 neurons
weight	5 layers with 32 neurons 45,000 epochs w_V, w_X, w_P, $w_S = 1$	W1	w_V, w_X, w_P, $w_S = 1$
		W2	$w_X = 3$
		W3	$w_P = 3$
		W4	$w_S = 3$
		W5	$w_V = 3$
		W6	$w_X = 3$, $w_V = 3$
		W7	$w_P = 3$, $w_S = 3$
		W8	$w_X = 2$, $w_S = 3$, $w_V = 5$
		W9	$w_X = 5$, $w_V = 10$
multiple_config	5 layers with 32 neurons 80,000 epochs Pre and post L-BFG-S optimization	Batch	-
		Fed-Batch	$V_S = V_{max}$ (5 L), $X_S = X_M$, $P_S = P_M$, $S_S = S_o$
		CSTR	$w_V = 3$ $V_S = V_{max}$ (5 L), $X_S = X_M$
pinn_numeric_xp	Not applicable	Experimental Data	Not applicable

		Euler	Euler simulation with 240 discretization points
		PINN	5 layers with 32 neurons 30,000 epochs

for all tests were the values are not specified, loss weights (w_V, w_X, w_P, w_S) and nondimensionalization scalers (V_S, X_S, P_S, S_S) were set to 1.

The layer size test was performed solely for the batch reactor case and is show in 2 - b. It was tested 12 different layer sizes. With a constant number of neurons of 22, experiments t_lay7, t_lay8 and t_lay9 varied only the number of layers to 3, 4 and 5, respectively. While there is a great improvement from 3 to 4 layers, the loss of 5 layers is about the same, for 22 more neurons to optimize. We can infer that, while increasing the number of layers may improve loss reduction, it can also need many more epochs to optimize, reducing the loss for a same given number of epochs. Experiments t_lay4, t_lay5 and t_lay6 were done for 3, 4 and 8 layers respectively, and show that when using smaller number of neurons more deepness (number o layers) is necessary to reduce loss and get out of local minimums. From t_lay 1 to 3, we can infer that ratios of neuron/layers (N/L) < 1 can yield better results than ratios > 1 for this range. From t_lay 4 to 6, we can infer that an N/L > 4 can yield better results increasing the number of layers, with N/L varying from 5.33 to 2.67. From t_lay 7 to 9, the loss reduces as N/L goes from 7.33 to 4.4. The loss is also smaller comparing t_lay11 (N/L = 10.67) to t_lay12 (N/L = 6.4). While there is a general trend in this data that N/L = 3 or less can generate better models, it was not possible to determine a general rule for the given system of equations. The best performance (smaller loss) was achieved in t_lay12, the model with more neurons per layer (32 neurons in 5 layers, N/L = 6.4).

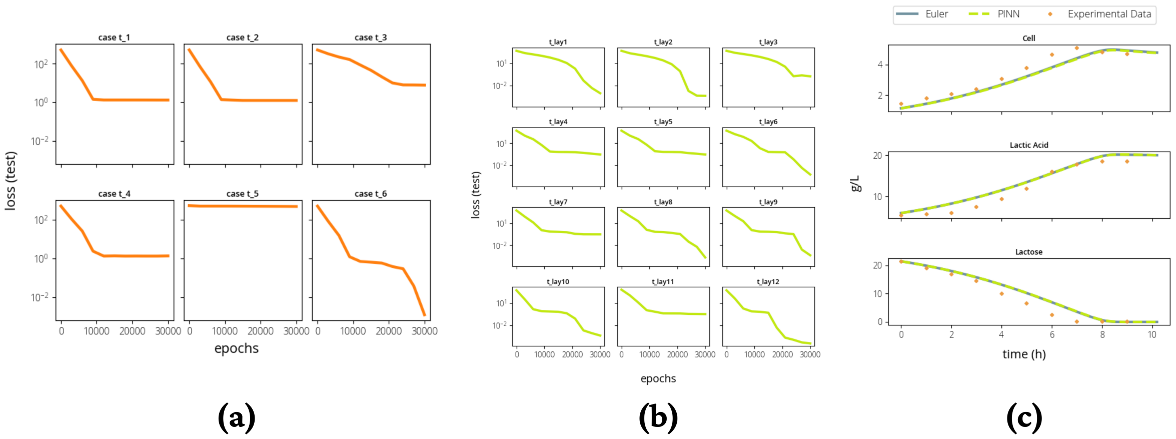


Figure 2: Nondimensional test, layer test and comparison between experimental data and PINN results

The plots shown are described on Tables 1 and 2, and are: (a) – t_S test; (b) – layer_size; (c) – pinn_numeric_xp.

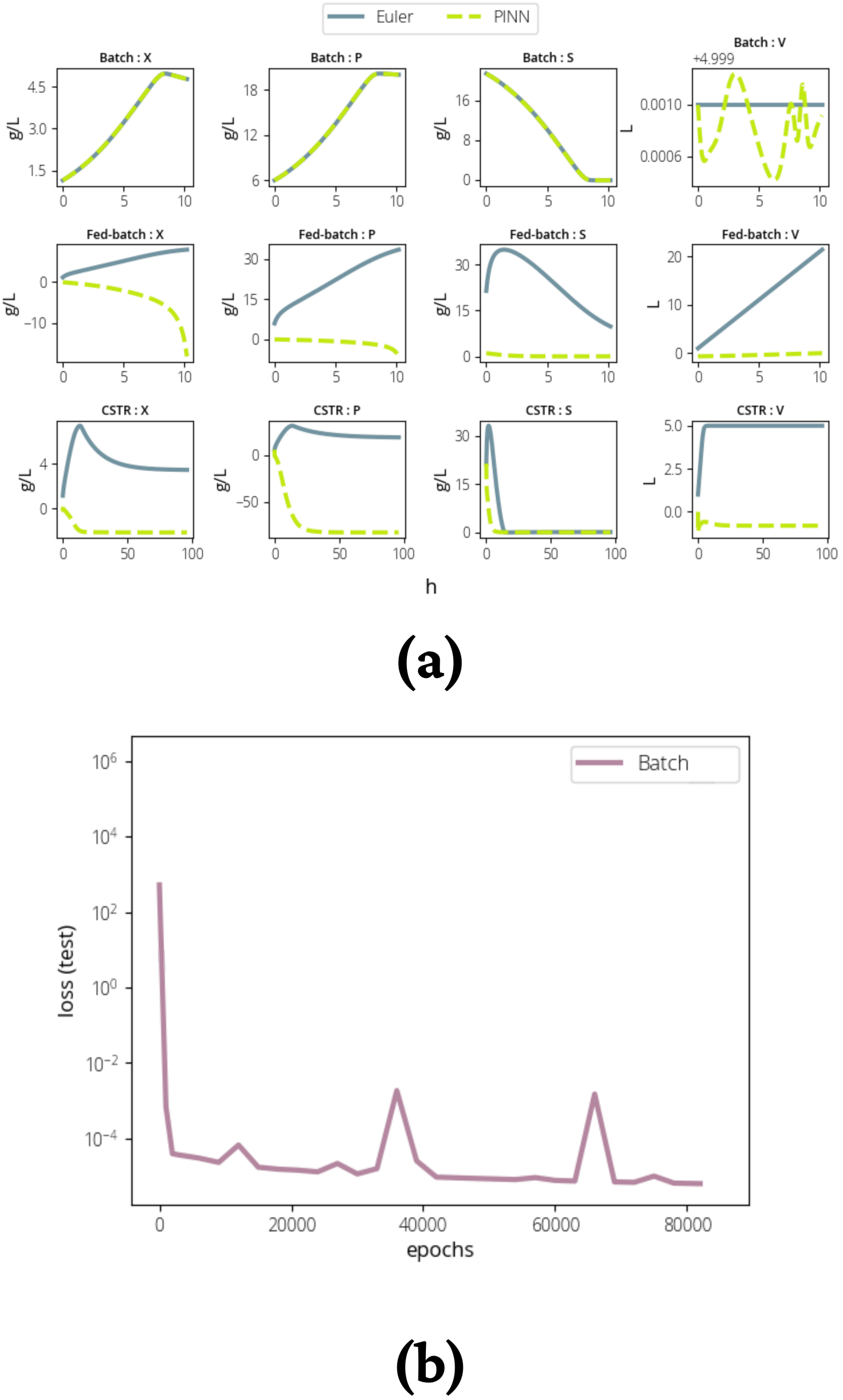
Loss weights tests were performed in different layer sizes (6 layers with 8 neurons, 3 layers with 22 neurons, 5 layers with 32 neurons, and 3 layers with 90 neurons) (data not published). It is clear that the simulation for the batch reactor showed a good performance for any group of parameters, with the exception of W3, where w_P = 3. Fed-batch is the model with the worst performance in almost every group of weights, but performs better at W2, with w_X = 3. Surprisingly, fed-batch loss was smaller in the deeper and narrower layer size (8 neurons in 6 layers), with an error of. 8.03×10^{-1} after 120,000 epochs. Batch performed better in the most equilibrated layer size (32 neurons in 5 layers), with an error of 1.37×10^{-5} after 45,000 epochs. The only configuration were CSTR performed well (loss $<10^{-2}$) was also the only one were it got smaller losses than the other two reactors, with layer size of 90 neurons in 3 layers and 120,000 epochs. This can't be determined as an optimal point, though, since both batch performed slightly worse than the other variations and fed-batch loss was greater than 10. This indicates that the CSTR model needs more effort to be solved, probably due to one more variable (f_{out}) that the others don't have, as their outlet is closed. The batch and fed-batch models, though, not only benefit from simpler models because of less computational cost, but also can achieve slower losses in fewer epochs in simpler Neural Networks. Using 5 layers with 32 neurons in combination with case W3 was the only configuration were CSTR loss was smaller than the one of the batch simulation. Excluding this case, almost every combination of layer deepness and wideness and weight factors resulted in acceptable results for batch (loss < 10^{-3}), a greater loss for CSTR and an even greater loss for fed-batch.

For validation of the model proposed, the system of equations was solved for the batch reactor case using the traditional Euler Method (with 240 discretization points in time) and the result is plotted, alongside PINN (60,000 epochs and layer size of 32 neurons and 5 layers) and experimental data (from (ALTIOK; TOKATLI; HARSA, 2006)) in Figure 2 - c. We can conclude that the given PINN configuration can represent with great accuracy the biological reaction. The graphic doesn't answer, however, if the same model is capable of solve both the reaction and the volume variation, since in batch dV/dt = 0. The data (not published) show that the most probable culprit to the deviation when simulating CSTR and Fed-batch models is the volume. Despite having one inlet and one outlet, the CSTR eventually approaches a constant internal liquid volume, while fed-batch increases infinitely. In weight test, it was observed a trend where the existence of volume variation made the solution very unstable and of difficulty convergence. Since X depends on V and P and S depend on both X and V, they were significantly affected when the model could not predict appropriately the values of V and dV/dt. X, P and S values always present high losses when the volume predict loss is too high. Thus, it was expected that a greater w_V would make the solver more sensitive to errors in the volume variable and, consequently, be able to produce more precise results. Since this could not be validated in the loss weight tests, it is possible that this specific problem requires a greater amount of neurons (>90) and layers (>5) than the ones that were tested.

None of the combination of above configurations (layer size, number o neurons per layer, number of iterations/epochs, loss weights or nondimensional scalers) was capable of representing the fed-batch and CSTR reactors appropriately. A final test was tried, applying alongside the Adam algorithm, L-BFGS (BYRD et al., 1995) as a previous and post-processing step, combined with loss weights and nondimensionalization found to be the best fit on previous attempts. The result ins show in Figure 3 - a. The name of each subplot is constructed as "Name of the reactor : Name of the variable". So "Batch : V" for instance, represents the volume in the batch reactor over time. While the batch reactor model presents a persistent fluctuation of volume prediction in comparison to the correct value, it is irrelevant since more than 3 orders of magnitude smaller than the total volume. In another study with multiple equations, PINNs behaved similarly, being able to predict accurately two variables while one showed noise or considerable deviation (MAO; JAGTAP; KARNIADAKIS, 2020). The CSTR model results are alike the ones for many other configurations and the same reactor: only one of the four variables values could be more or less acceptably predicted, while the others are clearly deviated. Fed-batch has the greatest loss of the three models in almost any configuration. Also, both fed-batch and CSTR errors are clearly stagnated in values of order of magnitude of about 10^0, as show in Figure 3 - b.

Another relevant point is that, as shown in the inhibitory terms of equation 5, there are expressions raised to exponents. While ideally X and P would never be higher than X_m and P_m, respectively, this can't be assumed from a Neural Network that is learning and updating itself based on errors. Because of this, fluctuation of errors on volume or biomass (which were determined to be the most sensible features, probably because they are directly involved in the equations of P and S) can rapidly destabilize the system, predict values of X and P that are greater than their respective maximum values and generate negative numbers raised to 0.5 for example. This was the cause of numerous errors, so many tests were not finished: they were prematurely interrupted, since the error was infinite, and the system was incapable of recover from these fluctuations. While this may induce us to think that the losses values being too high were entirely related to the biomass and product inhibitory effects, the batch model shows clearly that they are not the cause – at least not the only cause. The volume presents some fluctuations even though X, P and S are appropriately in accordance with the solution obtained using Euler Method. Therefore, we conclude that the main challenges faced by the system while trying to minimize the loss function were the volumetric balance over time, cells mass balance over time, or both. The mere use of loss weights and nondimensionalization didn't prove effective for reducing these problems and enabling the same configuration to represent both batch, fed-batch and CSTR reactor models.

Figure 3: PINN comparison for Bacth, Fed-batch and CSTR reactors



(a) – The profile of biomass (X), product (P), substrate (S) and volume (V) for each reactor simulated. (b) – The total loss for the simulation of the batch reactor over the number of epochs/iterations.

While PINNs are very interesting and open the doors of Machine Learning to many applications in fields of science that don’t have a multitude of data available, we believe a better approach would be to use PINNs to model only the reaction given a starting point, and to solve the physics-involved variables, which are much more straightforward, by a traditional numeric method. The system would be, then, a hybrid of PINN and traditional numeric integrator or solver system. We also could not find a clear relation between hyperparameters, nondimensionalization and loss weights that could serve as a guide for future works and substitute or rationalize the Grid Search other than the loss increasing with t_S increasing.

6 CONCLUSIONS

Simpler Neural Networks, with less neurons and layers, were capable of producing acceptable losses for the reaction kinetics (batch reactor) only, but performed poorly on balancing the reactions and the volume variation simultaneously. Almost all configurations produced high losses when dV/dt was not expected to be zero. Since the PINN performance for the batch reactor was exceptionally good and both CSTR and fed-batch models required a large amount of epochs to yield a still high loss, in cases where both the volume of the reactor is varying considerably and there are multiple reactions that reference each other, a better approach may be combine the PINN for solving the biological or chemical set of equations (group of reactions), which is intrisically more complex, and use a regular numeric method, like Euler or Runge-Kutta, to solve the physics part of the system (volumetric and mass balance) which is usually more well-known and established.

Physics-Informed Neural Networks were shown to represent with adequate accuracy biological reactions with complex inhibition terms and multiple derivative dependency. However, all combinations tested failed to represent adequately both the concentrations of X, S, and P and liquid volume variation in the reactor. We recommend future works to try employing nondimensionalization scalers for time (t_S) of values less than 1, even if the proposed t_s must be arbitrary and not based on actual constants from the equations. To explore new strategies to deal with these kind of physical and biochemical complex systems and to propose another ways to rationalize the process of creating a PINN model, we also recommend paying special attention to the independent (in this case, t_S) and quasi-independent (V) variables, since they can influence all other variables and be responsible for the success of failure of the model.

7 REFERÊNCIAS

ALHAMA MANTECA, I.; SOTO MECA, A.; ALHAMA, F. Mathematical characterization of scenarios of fluid flow and solute transport in porous media by discriminated nondimensionalization. **International Journal of Engineering Science**, v. 50, n. 1, p. 1–9, jan. 2012.

ALTAF, MD.; NAVEENA, B. J.; REDDY, G. Use of inexpensive nitrogen sources and starch for l(+) lactic acid production in anaerobic submerged fermentation. **Bioresource Technology**, v. 98, n. 3, p. 498–503, fev. 2007.

ALTIOK, D.; TOKATLI, F.; HARSA, Ş. Kinetic modelling of lactic acid production from whey byLactobacillus casei (NRRL B-441). **Journal of Chemical Technology & Biotechnology**, v. 81, n. 7, p. 1190–1197, jul. 2006.

BÜYÜKKILECI, A. O.; HARSA, S. Batch production ofL(+) lactic acid from whey by Lactobacillus casei(NRRL B-441). **Journal of Chemical Technology & Biotechnology**, v. 79, n. 9, p. 1036–1040, set. 2004.

BYRD, R. H. et al. A Limited Memory Algorithm for Bound Constrained Optimization. **SIAM Journal on Scientific Computing**, v. 16, n. 5, p. 1190–1208, set. 1995.

DE OLIVEIRA, P. M. et al. Production of L (+) Lactic Acid by Lactobacillus casei Ke11: Fed Batch Fermentation Strategies. **Fermentation**, v. 7, n. 3, p. 151, 13 ago. 2021.

DEY, P.; PAL, P. Modelling and simulation of continuous L (+) lactic acid production from sugarcane juice in membrane integrated hybrid-reactor system. **Biochemical Engineering Journal**, v. 79, p. 15–24, out. 2013.

DIN, N. A. S. et al. Lactic acid separation and recovery from fermentation broth by ion-exchange resin: A review. **Bioresources and Bioprocessing**, v. 8, n. 1, p. 31, dez. 2021.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, jan. 1989.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. arXiv, , 29 jan. 2017. Disponível em: <http://arxiv.org/abs/1412.6980>. Acesso em: 23 jan. 2023

KOMESU, A.; MACIEL, M. R. W.; FILHO, R. M. Lactic Acid Production to Purification: A Review. p. 20, 2017.

LI, Z. et al. L-Lactic acid production byLactobacillus casei fermentation with corn steep liquor-supplemented acid-hydrolysate of soybean meal. **Biotechnology Journal**, v. 1, n. 12, p. 1453–1458, dez. 2006.

LÓPEZ-GÓMEZ, J. P. et al. A review on the current developments in continuous lactic acid fermentations and case studies utilising inexpensive raw materials.

Process Biochemistry, v. 79, p. 1–10, abr. 2019.

LU, L. et al. DeepXDE: A Deep Learning Library for Solving Differential Equations. **SIAM Review**, v. 63, n. 1, p. 208–228, jan. 2021.

MAO, Z.; JAGTAP, A. D.; KARNIADAKIS, G. E. Physics-informed neural networks for high-speed flows. **Computer Methods in Applied Mechanics and Engineering**, v. 360, p. 112789, mar. 2020.

NANCIB, A. et al. The use of date waste for lactic acid production by a fed-batch culture using Lactobacillus casei subsp. rhamnosus. **Brazilian Journal of Microbiology**, v. 46, n. 3, p. 893–902, set. 2015.

POHANKA, M. D-Lactic Acid as a Metabolite: Toxicology, Diagnosis, and Detection. **BioMed Research International**, v. 2020, p. 1–9, 18 jun. 2020.

RAISSI, M.; PERDIKARIS, P.; KARNIADAKIS, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. **Journal of Computational Physics**, v. 378, p. 686–707, fev. 2019.

REZVANI, F.; ARDESTANI, F.; NAJAFPOUR, G. Growth kinetic models of five species of Lactobacilli and lactose consumption in batch submerged culture. **Brazilian Journal of Microbiology**, v. 48, n. 2, p. 251–258, abr. 2017.

SANTANA, V. V. et al. A First Approach towards Adsorption-Oriented Physics-Informed Neural Networks: Monoclonal Antibody Adsorption Performance on an Ion-Exchange Column as a Case Study. **ChemEngineering**, v. 6, n. 2, p. 21, 1 mar. 2022.

THAKUR, A.; PANESAR, P. S.; SAINI, M. S. Optimization of process parameters and estimation of kinetic parameters for lactic acid production by Lactobacillus casei MTCC 1423. **Biomass Conversion and Biorefinery**, v. 9, n. 2, p. 253–266, jun. 2019.

VERA-PEÑA, M. Y.; HERNÁNDEZ-GARCÍA, H.; VALENCIA-GARCÍA, F. E. Kinetic modeling of lactic acid production, co-substrate consumptions and growth in Lactiplantibacillus plantarum 60-1. **Revista DYNA**, v. 89, n. 224, p. 51–59, 2022.