UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
DEPARTAMENTO DE ENGENHERIA QUÍMICA
ENGENHARIA QUÍMICA INTEGRAL

HYBRID APPROACH BASED ON UNIVERSAL DIFFERENTIAL EQUATIONS FOR MODELING PARACETAMOL DISSOLUTION IN ETHANOL

AMYR CRISSAFF SILVA

RIO DE JANEIRO – RJ
2024

AMYR CRISSAFF SILVA

HYBRID APPROACH BASED ON UNIVERSAL DIFFERENTIAL EQUATIONS FOR MODELING PARACETAMOL DISSOLUTION IN ETHANOL

Trabalho de conclusão de curso de Engenharia Química da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Engenheiro Químico

Orientadores: Maurício Bezerra de Souza Jr., D.Sc.
Fernando Arrais Romero Dias Lima, M.Sc.
Marcellus Guedes Fernandes de Moraes, D.Sc.

RIO DE JANEIRO
2024

AMYR CRISSAFF SILVA

HYBRID APPROACH BASED ON UNIVERSAL DIFFERENTIAL EQUATIONS FOR MODELLING PARACETAMOL DISSOLUTION IN ETHANOL

Trabalho de conclusão de curso de Engenharia Química da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Engenheiro Químico

Aprovado em 06 de dezembro de 2024.

_____
Maurício Bezerra de Souza Jr., D.Sc., UFRJ

_____
Fernando Arrais Romero Dias Lima, M.Sc., UFRJ

_____
Marcellus Guedes Fernandes de Moraes, D.Sc., UERJ

_____
Gustavo Luis Caldas, M.Sc., UFRJ

_____
Bruno Didier Olivier Capron, D.Sc., UFRJ

RIO DE JANEIRO – RJ

2024

AGRADECIMENTOS

Agradeço a Deus por ter me abençoado com todas as vitórias que tornaram este momento possível; Aos meus pais Evandrea Crissaff e Rui Fernando Romariz pelo apoio incondicional, sempre me estimulando a continuar meus estudos e me apoiando em minhas escolhas; Aos meus avós Fernando Silva e Denizia Romariz pelo amor e carinho de sempre. À minha companheira Thaina Souza por todas as palavras de incentivo e conversas motivadoras.

Agradeço ao Prof. Maurício por ter confiado em mim e ter me oferecido a oportunidade de trabalhar com um tema tão inovador e em contato com pessoas brilhantes; Aos meus orientadores Fernando e Marcellus pelo gigantesco apoio em toda a jornada de desenvolvimento deste trabalho e orientação para conclusão. Com vocês pude aprender habilidades e senso crítico que vou levar para toda a minha jornada profissional.

Agradeço aos grandes amigos que fiz durante o curso, em especial ao Luiz Felipe Benevides e ao Lucas Jorge Spelta pela parceria inseparável que permaneceu desde o primeiro dia de graduação.

Agradeço à Universidade Federal do Rio de Janeiro por ter mudado a minha vida e a todos os meus colegas e professores que fizeram deste processo uma experiência, inesquecível e transformadora.

RESUMO

A cristalização é um processo fundamental na engenharia química, amplamente utilizado para separação e purificação em indústrias como a farmacêutica e a agroquímica. A modelagem precisa da cristalização é essencial para otimizar a qualidade do produto, incluindo o tamanho, a pureza e a morfologia dos cristais, além de enfrentar desafios como eficiência do processo e escalabilidade. A compreensão da cinética de dissolução de cristais é igualmente crítica, pois permite o controle preciso das taxas de dissolução, do manejo da supersaturação e da estabilidade geral do processo, elementos essenciais para operações eficientes e resultados de alta qualidade. Ferramentas baseadas em dados aplicadas à pesquisa científica têm ganhado destaque nos últimos anos e se mostrado ferramentas práticas para entender fenômenos complexos. Em campos científicos como a engenharia de processos químicos, a geração de dados confiáveis envolve custos elevados com experimentação, o que faz da modelagem híbrida uma solução viável para alcançar precisão com eficiência no uso de dados. Este trabalho é motivado pela oportunidade de contribuir para a aplicação correta dessa tecnologia e expandir as ferramentas disponíveis para modelagem de processos de cristalização. Este estudo utiliza uma abordagem de modelagem híbrida baseada em Equações Diferenciais Universais (UDE), integrando redes neurais como aproximadores universais dentro de equações de balanço populacional. O Método dos Momentos foi empregado, enquanto dados experimentais sobre a dissolução de paracetamol em etanol serviram como referência para o treinamento e a validação. Técnicas cuidadosas de treinamento, como o uso de “mini-batch” e parada brusca, foram implementadas para melhorar a robustez e a precisão preditiva. Duas redes neurais foram treinadas, e as escolhas de hiper parâmetros e as técnicas de regularização utilizadas foram discutidas. Análises das curvas de erro e dos perfis gerados pelos modelos permitiram insights valiosos, reforçando a importância das informações da fase sólida dos cristais. As conclusões ressaltam a importância de combinar métodos fenomenológicos e orientados por dados para avançar na modelagem de processos de cristalização.

Palavras-chave: modelagem híbrida; rede neural; Equações Diferenciais Universais; cristalização em batelada; paracetamol.

ABSTRACT

Crystallization is a fundamental process in chemical engineering, widely used for separation and purification in industries such as pharmaceuticals and agrochemicals. Accurate modeling of crystallization is crucial to optimize product quality, including crystal size, purity, and morphology, and to address challenges like process efficiency and scalability. Understanding crystal dissolution kinetics is equally critical, as it enables precise control over dissolution rates, supersaturation management, and overall process stability, which are essential for efficient operations and high-quality outcomes. Data-based tools applied to scientific research have been gaining attention through recent time and have been shown as practical tools for understanding complex phenomena. In some scientific fields such as chemical process engineering, reliable data is generated at cost of expensive experimentation, thus hybrid modelling emerges as a solution for achieving accuracy with data efficiency. This works is highly motivated by the opportunity to contribute to correct application of this technology and further expand the tools available for crystallization modelling. This study employs a hybrid modeling approach based on Universal Differential Equations (UDE), integrating neural networks as universal approximators within population balance equations. The Method of Moments is used, while experimental data on paracetamol dissolution in ethanol serves as a benchmark for training and validation. Careful training techniques, including mini-batch training and early stopping, were implemented to improve robustness and predictive accuracy. Two neural networks were trained and the hyper-parameter choice and regularization techniques used discussed. Insights were taken from analyzing both the error curve and the profiles obtained with each model. And the importance of crystal solid phase information was reinforced with the results. The results demonstrate the model's ability to accurately predict dissolution profiles and crystal size distribution across varying experimental conditions, showcasing its potential for process kinetic modelling applications.

Key words: hybrid modelling; neural network; Universal Differential Equations; Batch crystallization; Paracetamol.

TABLE OF CONTENTS

1 INTRODUCTION 8
2 BLIBIOGRAPHICAL REVIEW 11
2.1 CRYSTALLIZATION PROCESS MODELING 11
2.1.1 Crystal Size Distribution 11
2.1.2 Population Balance Modeling applied to Crystallization 13
2.1.3 Crystallization Kinetics 15
2.1.4 Method of Moments 19
2.2 HYBRID MODELLING 20
2.2.1 Artificial Neural Networks 20
2.2.2 UDE 22
2.2.3 Neural Network Training and Optimization 23
3 METHODOLOGY 26
3.1 PBM WITH METHOD OF MOMENTS 26
3.2 NEURAL NETWORK AND OPTIMIZATION 28
3.2.1 Neural Network Programing and UDE integration tools 28
3.2.2 IVP integration and data preparation 29
3.2.3 Error evaluation 32
3.2.4 Network initialization, forward pass and training 33
4 RESULTS AND DISCUSSION 38
4.1 MODELS DEVELOPED 38
4.1.1 Model 1: Test model 38
4.1.2 Model 2: Supersaturation constrained model 40
4.2 MODEL PERFORMANCE METRICS 45
5 CONCLUSION 48
BIBLIOGRAPHY 50

1 INTRODUCTION

Crystallization is a separation and purification process that may be defined as a phase change in which crystalline product is obtained from a solution (Myerson; Erdermir; Lee, 2019). It is a crucial unit operation and is commonly used for purification, separation and production. It is a practical method that produces pure, concentrated, appealing and convenient to handle chemical substances (Tavare, 1995). Thus, crystallization processes are present in many industries such as pharmaceuticals, food and agrochemicals. Many active ingredients in pharmaceutical industries are produced in the form of crystals, such as paracetamol (Kim et al.,2023), praziquantel (Moraes et al.,2023a) and amoxicillin (McDonald et al., 2019).

Crystallization processes are described by kinetic models and depend on supersaturation as driving force (Tavare, 1995). Supersaturation can be created on a solution by different means such as temperature change, solvent evaporation, chemical reaction, and changes in solvent composition. As solubility of most materials decline as temperature drops, cooling crystallization is one of the most common methods to induce supersaturation (Myerson; Erdermir; Lee, 2019).

A common issue revolving crystallization processes is the formation of small crystals, called fines. These may cause different problems such as filtration efficiency reduction and/or product specification difficulties (Myerson; Erdermir; Lee, 2019; Kim et al.,2023). Hence, initial crystallization research work recognized the importance of crystal size distribution (CSD) evaluation and established its interrelationship with crystallizers design criteria and operation conditions as a central theme to studies that were to come (Tavare, 1995). Many studies focused on one dimensional CSD modelling and the most applied phenomenological approach is the population balance model (PBM) coupled with constitutive equations that describe how size distributions are influenced by kinetic mechanisms such as nucleation, crystal growth, agglomeration, dissolution and breakage (Myerson; Erdermir; Lee, 2019; Lima et al., 2023).

More recently, new studies are developed focusing on the proposal of models with parameter estimation for crystallization and dissolution kinetics. (Myerson et al, 2019; McDonald et al., 2019; Kim et al.,2023). This concern comes from the necessity of formulating efficient models that can be used for crystallizations process control. Process control strategies applied to crystallization operations often include

temperature cycling and navigating operating conditions through different supersaturation states (Moraes, 2023; Kim et al., 2023; Trampuž et al., 2019; Szilágyi et al., 2022). In this context dissolution kinetics modelling appears as an important task for understanding crystal behavior in undersaturated conditions and creating model-based control approaches.

Recently data-driven tools such as Neural-Networks have been applied to predict kinetic rates for crystal nucleation, growth and dissolution for both organic and inorganic compounds. Combining such techniques with the conservative principles of the population balance modeling composes what has been called hybrid semi-parametric modeling (von Stosch et al., 2013; Lima et al., 2023; Moraes et al., 2023b).

Parametric or mechanistic models are determined based on knowledge about the process and have a fixed number of parameters which, most of them, have a physical or empirical interpretation (Thompson and Kramer, 1994). Nonparametric models are determined exclusively from data, the number of parameters and meaning are not determined in advance based on prior knowledge (Thompson and Kramer 1994). Hybrid Semi-parametric models are defined as models that combine parametric and nonparametric sub models. The main advantage of using such approach relies on the higher benefit/cost ratio to solve complex problems. Hybrid models are especially useful when there is no precise knowledge about specific terms of the model, but enough experimental data is available to infer unknown patterns of the phenomena. (von Stosch et al., 2013). The application to process modeling has evolved from neural networks field and was first reported in 1992 with uses in crystallization reported in 2000, 2003 and 2007 (Lauret et al., 2000; Georgieva et al., 2003; Georgieva & de Azevedo, 2007)

The advances in computing power allowed new studies on the use of neural networks and hybrid models in process engineering. Some examples are the use of the Universal Differential Equation (UDE) framework with artificial neural networks (ANNs), denoted by Rackaukas et al. (2020), composing a non-parametric technique for modeling terms that otherwise would require consistent thermodynamic correlations, heuristic rules or kinetic mechanisms description (Nogueira et al., 2022; Bangi; Kao; Kwong, 2022). Recently Lima et al. (2023) presented a novel hybrid model for crystallization using the UDE framework for the potassium sulfate crystals kinetic parameters. The results obtained revealed a less complex and still effective method that also reduces the influence of stochasticity of traditional hybrid-modeling

approaches. Models built with this method present smooth responses that avoid error mimicking and thus are more robust towards experimental variability and sensor noise.

The UDE approach is especially suitable because it uses functions as universal approximators that replace one or more terms in a differential equation. Using neural networks as universal approximators allows greater dimension inputs that account for unknown relationships in the process (Lima et al., 2023). To properly employ an UDE coupled with ANN's strategy to model chemical process the neural network's parameters need to be tuned. This is done by the application of an optimization strategy to find the parameters that minimize the difference between the predicted and experimental values (Lima et al., 2023). Different optimization strategies can be applied to tune neural networks in a UDE model. For example, Lima et al. (2023) used SLSQP to create a model for potassium sulfate crystallization; Bangi et al. (2022) used a 2-step method combining Adam algorithm for finding a minimum and a BFGS (Broyder-Fletcher-Goldfarb-Shanno algorithm) is applied to fine tune the parameters for beta-carotene fermentation UDE model.

Model nonlinearity makes objective function minimization difficult, and some considerations must be taken when choosing the best numerical method for optimization, such as the size of the parameter space, the existence of a local minimum, the continuity of an objective function and the sensitivity of the objective function to each model parameter. Thus, choosing the right optimization strategies and regularization techniques become an important topic of study as UDE framework is cemented as scientific machine learning tool for process modeling.

The objective of this work is to propose a model based on universal differential equations (UDEs) approach with a neural network as a universal approximator for the kinetic term of dissolution rate in a population balance model. The model will use a deterministic method for Neural Networks optimization using the work developed by Lima et al. (2023) as a benchmark for model structure and its errors will be evaluated against experimental data gathered from "Modeling of Nucleation, Growth and Dissolution of Paracetamol in Ethanol Solution for Unseeded Batch Cooling Crystallization with Temperature Cycling" (Kim et al., 2023). Further steps and improvements will be proposed based on results obtained.

The following pages include a bibliographical review of crystallization process modelling and hybrid modelling techniques, a methodology chapter explaining the model development and a results and discussion chapter followed by the conclusion.

2 BIBLIOGRAPHICAL REVIEW

2.1 CRYSTALLIZATION PROCESS MODELING

Crystallization can be seen as a self-assembly process where molecules, initially arranged randomly in a fluid, organize themselves into a structured three-dimensional array with a repeating, periodic pattern. (Myerson et al., 2019). Crystallization is a widespread unit operation in the chemical industry and is considered an important separation and purification method in food and pharmaceutical processing (Garside, 1985; Tavare, 1995; Paul et al., 2005). More than 200,000 various substances are crystallized daily in plants and laboratories all over the world and over 90 percent of all pharmaceutical products contain bioactive substances and excipients in the crystalline form for reasons of stability and ease of handle (Shekunov and York, 2000).

The main quality criteria of crystalline products are four: crystal size, purity, morphology, and crystal structure (Myerson; Erdermir; Lee, 2019). Many crystallization operations aim to produce larger crystals that can be separated easily through filtration or centrifugal steps. In some applications like inducing a faster dissolution rate and improved bioavailability a smaller crystal size is desirable (Shekunov and York, 2000; Carpenter and Wood, 2004). Hence crystal size modeling and control is needed for the development of new products and improvements on existing processes.

2.1.1 Crystal Size Distribution

Bulk crystallization processes invariably yield a distribution of characteristic crystals size and shape of the result solid-phase material (Tavare, 1995). Therefore, the product of a crystallizing process is not fully characterized by one single and linear dimension. Every specification of size by linear dimension is related to how the dimension is measured. Moreover, a single linear dimension does not carry the information about the shape of a crystal regardless of the determination method (Myerson; Erdermir; Lee, 2019). However, if the crystals of the same substance produced in a specific process all have roughly similar shape it is convenient to describe the material obtained by a one-dimensional particle distribution (Tavare, 1995). For that description, many methods have been used to measure and represent the particle size distribution.

CSD can be represented either as a mass-volume distribution or as number distribution. Number distribution is tied to methods of determination where crystals are examined one by one and can be counted (Myersonet al., 2019). Mass distributions are more related to industrial processes and to CSD determination by sieving (Nyvlt, 1985). Volume distribution can be useful when modelling crystal growth, agglomeration, and breakage in continuous or batch crystallizers (Tavare, 1995). Furthermore, the CSD is described by histogram charts or frequency distributions. In histogram representations each bar is a range of crystal size, and the height of the bar represents the mass or number of particles in that range. In frequency distribution the number is replaced by the number density (n), which is the number of particles divided per unit length of particle size. Defined by equation (1) and in its differentiated form in equation (2) where N is the number of crystals and L is the characteristic length of crystal. The number density (n) is also called population density (Myerson et al., 2019). Additionally, to histograms and frequency distributions, some statistics distribution models have also been used to describe CSD such as the normal, log-normal and gamma distributions (Tavare, 1995).

$$n(L_i) = \frac{N_i}{L_{i+1} - L_i} \quad (1)$$

$$n = \frac{dN}{dL} \quad (2)$$

Sieving is the classical method for characterizing size distributions, a known sample of suspension is weighted, separated from its mother liquor by filtering, washing and drying and then the dried crystals are sieved through different sieves with many apertures. The distribution is acquired by estimating the mean size of each fraction of the sieve by an arithmetic mean or a volumetric average. Still no information of the size distribution inside each sieve fraction is obtained hence uncertainty is inherent (Myerson; Erdermir; Lee, 2019).

Many Instruments can also be used to determine CSD making use of different physical principles such as laser diffraction (Adnan and Samad, 2023), electro-sensing zone technique (Caro et al.,2014), focused beam reflectance measurement (Gómez, 2014), and dynamic image analysis (Moraes et al.,2023b).

Proper crystallization modelling aims to predict CSD a priori based on well-defined crystal formation processes. This is possible through the solution of differential equations rather than by algebraic statistical distribution functions. Including process

parameters that affect CSD formation in the formulation of those differential equations allows proper description of the crystallization system and prediction of system change. PBMs are necessary for proper continuity depiction of a series of CSD that may form in crystallization processes (Tavare, 1995).

2.1.2 Population Balance Model applied to Crystallization

PBM is a framework for describing the evolution of a population of particles by the number density function for many industrial processes such as gas-liquid dispersions, liquid-liquid dispersion, aerosol engineering, crystallization systems and many others (Li et al., 2019). Population balance framework was first employed to model crystallization systems in two pioneer papers by Randolph and Larson (1962); and by Hulburt and Katz (1964). Since then, it has become a standard tool for crystallization processes description used in many research papers (Moraes et al.,2023b; Lima et al., 2023; Kim et al.,2023; Braatz, 2001).

Population balance models use number density as the dependent variable and in most cases a crystal one-dimensional size as its independent variable. Although other independent variables could also be used as particle volume or mass of suspension (Myerson et al., 2019). A population balance for an ensemble of crystals is written as equation (3) and (4):

$$Accumulation = Input - Output + Net\ generation \tag{3}$$

$$\frac{dn}{dt} + \nabla(v_e n) + \nabla(v_i n) + D - B = 0 \tag{4}$$

Where $\frac{dn}{dt}$ refers to the total rate of accumulation of number of crystals within time, while $\nabla(v_e n)$ and $\nabla(v_i n)$ are respectively the flow of crystals from internal and external phase space. The D term is the death term associated with crystal disappearance, often dissolution, and B is the birth term associated with crystal appearance, in most cases nucleation. This equation coupled with appropriate kinetic correlations for the rate processes, mass and energy balances, and boundary conditions can describe completely crystallization systems (Tavare, 1995). The complete solution of these equations can be difficult and most applications for modeling processes apply restricted forms of the original equation. The most used and

convenient form is the assumption of a well-mixed crystallization system with the particles characterized as a one linear dimension size as denoted in equation (5) (Myerson et al., 2019).

$$\frac{dn}{dt} + \frac{d(Gn)}{dL} + D(L) - B(L) + n\frac{dlnV}{dt} + \sum \frac{n_k Q_k}{V} = 0 \quad (5)$$

The term $\frac{d(Gn)}{dL}$ appears accounting for crystal growth rate and is related to population flux along the internal phase space coordinate, size (Myerson et al. 2019). Further restrictions could be made as neglecting $n\frac{dlnV}{dt}$ and $\sum \frac{n_k Q_k}{V}$ terms that account for variations in volume affecting the number density and inflow and outflow from the mixed volume and considering death and birth of crystals only in specific situations as boundary conditions (Moraes et al.,2023b; Lima et al., 2023; Kim et al.,2023; Braatz, 2001). Equation (6) is an example of a boundary condition of nucleation or birth of crystals at negligible size and equation (7) denotes a population balance model with size independent growth rate as used by Kim et al. (2023) in paracetamol crystallization modelling.

$$n^0 = n(L = 0) = \frac{B_0}{G(L = 0)} \quad (6)$$

$$\frac{dn}{dt} + G\frac{dn}{dL} = 0 \quad (7)$$

Alongside equations to describe changes in particle size distribution, PBM also needs equations that describe the rate of change in solute concentration. Solute concentration is a key parameter to estimate supersaturation, therefore dictating the kinetics of the process. This is called supersaturation balance and is a common mass balance that may vary its formulation as the definition of supersaturation of each model (Myerson et al., 2019). A general form of supersaturation mass balance is exposed in equation (8) where the concentration rate $\frac{d\Delta c}{dt}$ is function of spatial $((Qc)_{in} - (Qc)_{out})$ and kinetics generation and consumption terms $V(N_G - N_c)$. Nucleation and growth terms are described by kinetic correlations dependent on supersaturation, and thermodynamic correlations are often used to describe supersaturation generation based on specific mechanisms studied in each process and substance (Lima et al.,

2023; Moraes et al.,2023b). For cooling crystallization, only temperature dependency is evaluated, and parameters may vary with system composition.

$$V\ \frac{d\Delta c}{dt} = (Qc)_{in} - (Qc)_{out} + V(N_G - N_c) \quad (8)$$

The population balance provides robust equations for crystallization processes description. However, limitations and possibilities are well known. Providing industrial relevant models for crystallization kinetics that describe birth, death and growth terms are essential for real process modelling and thus it is the biggest limitation of PBM's (Myerson et al., 2019).

2.1.3 Crystallization Kinetics

Once a population balance model is structured, the modelling narrows down to a parameter estimation problem. In crystallization, kinetic parameters for crystal growth, formation and depletion are the most important for proper system description. (Braatz., 2001). In crystallizers modeling, power-law expressions are often used to describe crystal growth, primary nucleation and secondary nucleation (Braatz, 2001). These expressions are functions of supersaturation that are defined in two interchangeable forms, as the difference or ratio between concentration and solubility (Myerson et al., 2019). Below general power-law expressions for primary nucleation (9), secondary nucleation (10), crystal growth (11) where k_{np}, k_s and k_g denotes specific velocities for kinetic processes and $\Delta c = (c - c^*)$ is the absolute supersaturation comprised by the difference of concentration and equilibrium concentration for a given state.

$$B_p = k_{np}\Delta c^{np} \quad (9)$$

$$B_s = k_s M_T^m N_S^h \Delta c^{ns} \quad (10)$$

$$G = k_g \Delta c^g \quad (11)$$

Supersaturation is often expressed as concentration difference (12) or as ratio of concentrations (13) where c^* is the equilibrium concentration:

$$\Delta c = c - c^* \quad (12)$$

$$S = \frac{c}{c^*} \quad (13)$$

Nucleation and crystal growth compose the dominant rate processes in a crystallization system, although several other events are identifiable and can be considered in this operation (Tavare, 1995). Some recent models include crystal

dissolution as an important mechanism, especially when the modeling goal is crystal size distribution control, as shown in models elaborated by Moraes et al. (2023b) and Kim et al. (2023). Crystal dissolution modelling is useful for creating model-based control schemes depending on supersaturation manipulation and thus understanding the crystal kinetics in undersaturated conditions becomes imperative.

Nucleation is the birth of nuclei in a supersaturated solution and is the process responsible for determining properties of the resulting solid phase such as purity, crystal structure, and particle size. Nuclei are the product of an aggregation process that act as the center of a crystallization (Myerson et al., 2019). There are two kinds of nucleation: primary and secondary. Primary nucleation occurs directly from the solution at high supersaturation values. Secondary nucleation is the emergence of small crystals in the presence of bigger parent crystals as the result of many different effects such as attrition and breakage (Braatz, 2001).

Primary nucleation can be homogenous, product of solute concentration fluctuations in absence of solid interfaces, and heterogenous, in presence of foreign interfaces that function as centers for nucleation (Myerson et al., 2019). Many physical models are available for primary nucleation. The simplest and most well-known is the classical nucleation theory that describe homogenous nucleation with equations (14) that were originally derived from an analogy to a vapor condensation process (Tavare, 1995).

$$J = Aexp\left[\frac{16\pi\sigma^3 v^2}{3k^3T^3(lnS)^2}\right] = Aexp[-K(lnS)^{-2} \tag{14}$$

Although primary nucleation can be well described by CNT as homogenous nucleation (Kim et al.,2023), in real processes this rarely occurs as the presence of impurities can act as foreign surfaces for heterogenous nucleation (Myerson et al., 2019). Therefore, some models still recur to power-law equations for nucleation modeling and many mix primary and secondary nucleation in the same expression or do not consider primary nucleation at all (Moraes et al.,2023b; Lima et al., 2023; Morris et al.,2015; Halfwerk et al.,2023; Moraes et al., 2021).

In industry, most of the crystallizing operations are seeded and nucleation is mainly governed by particle-particle and particle-impeller collisions, thus secondary nucleation has a significant importance in industrial process description (Braatz, 2001). Physical models for secondary nucleation have been developed accounting for crystal collision with the impeller and internal stress distribution at the time of each collision to

predict the effects on crystal nucleation, such as the study from Gahn and Mersmann (1999). However, secondary nucleation is a complex phenomenon still not understood completely. There is no general theory for rate prediction and still many models use power-law expressions for kinetic approximation as equation (15) (Myerson et al., 2019).

$$B = k'_N W^i M_T^j (\Delta C^n) \tag{15}$$

Additional terms are often used in those expressions to account for variables that are believed to influence the process, as suspension density (mass of crystals per volume of solution) and agitation rate when relevant (Moraes et al.,2023b; Lima et al., 2023; Morris et al.,2015; Halfwerk et al.,2023; Moraes et al., 2021).

Crystal growth is often described by linear growth rate, which is the change in a given dimension of the crystal with time. It has dimensions as length per unit time. It is often used to describe the increase in a characteristic length that could be easily translated to surface area and crystal volume using a shape factor (Myerson et al.2019). A general definition of growth is expressed in equation (16).

$$G = \frac{dL}{dt} \tag{16}$$

Crystal growth can also be measured by specific rate of mass deposition rather than linear crystal size growth, as shown by Tavare (1995). As discussed before, with the right shape factors (accounting for volume and shape features) the expression below (17) could be used to convert linear growth rate to mass deposition rate (Myerson et al., 2019).

$$R_G = \frac{1}{A}\frac{dm}{dt} = 3\frac{\alpha}{\beta}\rho G = 3\frac{\alpha}{\beta}\rho\frac{dL}{dt} \tag{17}$$

Where R_G denotes the increase of mass per unit time per unit surface area; A is the surface area of the crystal; α and β are volume and area shape features respectively, ρ is the crystal density and L is the crystal characteristic dimension.

In industrial applications, classical kinetic theories assume that crystal growth involves two steps: transportation of solute in solution, called bulk diffusion (18); and a surface reaction or particle integration step, called surface integration (19). Crystal dissolution otherwise is assumed to be determined solely by bulk diffusion. Experiments have been designed to describe kinetics of both steps and compose a

general expression for crystal growth rate as shown below on equation (20) (Tavare, 1995).

$$R = k_d(c - c_i)^d \tag{18}$$

$$R = k_r(c_i - c^*)^r \tag{19}$$

$$R = k_r\left[\Delta c - (R/k_d)^{\frac{1}{d}}\right]^r \tag{20}$$

More recent approaches usually start from the general power-law expression for kinetic growth that was shown before and fit the k_g constant into an Arrhenius expression, as denoted in equation (21) and (22), to describe its temperature dependency (Myerson et al., 2019). The use of the Arrhenius equation is particularly useful as the activation energy contains information whether the rate controlling step is bulk-diffusion or surface integration (Lefever 1971; Nyvlt et al. 1985).

$$k_g = Aexp\left(\frac{-E_G}{RT}\right) \tag{21}$$

$$G = Aexp\left(\frac{-E_G}{RT}\right)\Delta C^g \tag{22}$$

This equation is sufficient for size-independent crystal growth modeling. However, size-dependent growth rates may be needed in some cases. A solution to this problem was presented by Moraes et al. (2023b), using a size-dependent growth constant and the general power-law and Arrhenius formulation.

As discussed, many parameters are needed for modeling crystallization kinetics, thus methods for correct estimation have been studied throughout the years. A common approach is the use of a MSMPR (mixed-suspension mixed-product removal) crystallizer at a static steady state, monitoring the crystal size and its population density, when experimenting through different operation conditions parameter estimation for the power-law expressions presented can be performed (Braatz, 2001).

Another common approach is the solution of an optimization problem minimizing an objective function related to the squared error between experimental and predicted data. A non-linear optimization method is necessary, and many approaches have been applied to estimate parameters on a completely phenomenological model (Caro et al., 2014; Morris et al., 2015; Kim et al., 2023). With the improvements in non-parametric techniques, such as ANNs, hybrid models are also being used for estimation of kinetic

rates of crystallization, and for the optimization problem, it becomes a neural network parameter estimation problem and not a kinetic parameters estimation anymore (Moraes et al.,2023b; Lima et al., 2023; Lauret et al. 2000; Georgieva et al., 2003).

2.1.4 Method of Moments

A common solution of the PBE (population balance equation) in many engineering applications is the use of the method of moments or moments transformations (Moraes et al., 2023b). In this framework, rather than knowing the complete crystal size distribution, the estimation of a representative average or total attribute of a distribution is calculated in the form of moments or ration of moments (Tavare, 1995).

Parting from the restricted population balance equation (23), the j^{th} moment of a crystal size distribution can be defined as expression (24):

$$\frac{dn}{dt} + \nabla(v_e) + \frac{d(Gn)}{dL} - B + D = 0 \tag{23}$$

$$\mu_j(x_e, t) = \int_0^{\infty} nL^j dL \tag{24}$$

Applying equation (23) to the moment definition (24), the moment transformation of the population balance model is carried over and results in the following set of equations (25) used in many works under different assumptions (Moraes et al.,2023b; Lima et al., 2023; Kim et al.,2023).

$$\frac{d\mu_j}{dt} + \nabla\left(v_e\mu_j\right) = 0^j B_0 - jG\mu_{j-1} + \overline{B} - \overline{D} \qquad j = 0,1,2,\ldots \tag{25}$$

Disregarding the term v_e that accounts for flow of crystals across the control volume and averaging the birth and death functions over a range of interest the population balance partial differential equations are reduced suitably to a set of ordinary differential equations converting a spatial-time dependent problem to time-only dependent problem as stated by Kim et al. (2023).

By this method, a reduction in dimensionality is obtained by converting a single four-dimension equation to an infinite set of three-dimension equations which is usually truncated to a finite set usually of three or four equations. The moments transformation often offers great advantages in numerical solutions of PBM problems.

2.2 HYBRID MODELLING

2.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are models that were developed to mimic the human brain. ANNs are also known as learning algorithms and are part of the representation learning field, which is a machine learning approach that aims to map features of interest for a specific task execution (Goodfellow et al., 2016).

The typical example of an ANN is the feedforward networks or multi-layer perceptron (MLP). These are models that approximate a specific function through mapping an input to an output by determining a specific set of parameters that results in the best approximation. In MLPs there are no feedback connections and information flows from input through all the intermediate computations used to define the function. When the ANN presents feedback connections, it is called recurrent neural network (Goodfellow et al., 2016).

MLPs are structured on a fixed number of basis functions determined in advance. MLPs different parameters values that may be adapted during a training phase in order to reduce a cost function or error function (Bishop, 2006). MLP can be described as a series of nonlinear regression models on top of each other, with the final layer being either a logistic regression or a linear regression depending on whether the problem is a classification or regression problem (Murphy, 2012).

$$y(x, w) = f\left(\sum_{j=1}^{M} w_j \phi_j(x)\right) \tag{26}$$

A typical output layer of a neural network can be modeled by the expression (26) with $\phi_j(x)$ being a nonlinear basis function and w_j are the weights of a linear regression and $f(*)$ is the activation function in the case of a classification or identity in the case of a regression. Neural networks use basis functions in the form of function (26). This way, each basis function works as a nonlinear transformation of a linear combination of inputs where the parameters of the linear combination are adaptive (Bishop, 2006). The adaptability of weight parameters of the linear combination grants the training capability of the model. To establish the nonlinearity of the model, two examples of activation functions $f(*)$ are the hyperbolic tangent and the sigmoidal function. The first one is preferred for nonlinearity in the hidden nodes as it maps the response R to [−1, +1], the last one is preferred for nonlinearity for binary nodes at the

output layer as it maps R to [0, 1] (Murphy, 2012). The application of other nonlinear functions have been reported as the rectified linear unit (ReLU) and the leaky rectified linear unit (Leaky ReLU), as well as many others.

Each repetition of the structure presented in equation (26) is denominated as a layer and together they compose a chain structure with the layers in between the inputs and the output of the model being called hidden layers. The overall length of this chain structure gives the neural network depth, hence the names deep neural networks or deep learning are often used (Goodfellow et al., 2016).

Using the structure shown before, according to Bishop (2006), the basic model for a neural network is described with the following expressions that are sequentially combined to form the network layers.

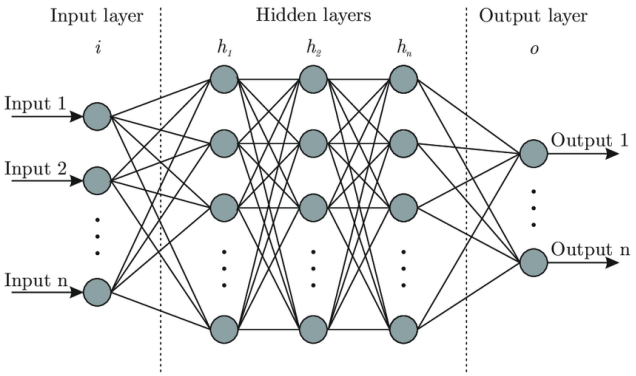
$$a_j = \sum_{i=1}^{N} w_{ji}^{(1)} x_i + b_{j0}^{(1)} \tag{27}$$

$$z_j = h(a_j) \tag{28}$$

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \tag{29}$$

Where N is the number of input variables of any given model, j is the number of linear combinations in single unit and the superscript (1) and (2) indicates the corresponding layer of the neural network. Equation (27) denotes the activation function, and equation (28) denotes the post activation function that must be differentiable and non-linear to grant the model a nonlinearity. The third equation denotes another layer where the process of activation and post-activation is looped through all the predefined layers of the model. By convention, the parameters w_{ji} and b_{j0} are known as weights and bias respectively. This basic structure can be represented in the form of a neural network diagram (Figure 1), where every line represents an activation, and every circle represents the value obtained after applying the post activation function. The model can be extended both in width and depth by changing the number of post activations in each layer and the number of layers respectively.

Figure 1 - General Structure of a feed-forward neural network



Source: (Lavanya Shukla, 2019)

2.2.2 UDE

Universal differential equations (UDE) are a scientific machine learning framework denoted by Rackauckas et al. (2021) that aims to bridge the gap between mechanistic models and data driven techniques, such as ANNs and other machine learning structures. The importance of such approach relies on the fact that in some fields of study, such as Chemical Engineering, the expense of scientific experiments impair the generation of large amounts of data necessary to build models based solely on data driven techniques. Because of the cost of such experiments, mechanistic models are still largely used in many fields of engineering. However, the data driven approach can enable more flexibility by allowing one to drop simplifications needed to derive mechanistic models and by that avoiding the necessity of prior structural knowledge of the phenomena (Rackaukas et al., 2021).

Physics-informed neural networks (PINNs) have shown the advantages of combining machine learning with differential equations and have become popular since its reported use by Raissi et al. (2019). PINNS framework uses partial differential equations and physics laws in cost functions of neural networks to induce scientific knowledge in its training process. While it has been shown to be data efficient by using very small data sets to achieve good accuracy (Wang et al.,2020 and Zhong et al., 2023), as stated by Rackaukas et al. (2021), PINNs frame the solution process as a large optimization problem and do not incorporate the numerical techniques that have led to stable and efficient solvers the majority of scientific models.

In contrast the formalism denoted as UDE comprises differential equations which are defined in full or part by a universal approximator, a parametric object capable of representing any function given a certain number of parameters. Neural

networks are common high dimensional universal approximators (Park et al., 2020). The universal differential equation in its most generalized form is mathematically a forced stochastic delay partial differential equation defined with embedded universal approximators with the form expressed below:

$$N\left[u(t), u\left(\alpha(t)\right), W(t), U_\theta\left(u, \beta(t)\right)\right] \tag{30}$$

Where $\alpha(t)$is a delay function and $W(t)$ is a Wiener process (Rackaukas et al., 2021)

As examples of UDEs applied to process engineering we can cite the work done by Lima et al., (2023) which heavily inspired this work. Neural networks were used to predict the kinetic terms of Nucleation and Growth of a PBM with moments transformation. Another recent application is the one performed by Faria et al. (2024) in which an UDE framework is used with neural networks for estimating a reservoir oil and gas flowrate in a Gas-Lift oil well operation. Nogueira et al. (2022) have used UDE framework applied to multicomponent adsorption modeling to solve a PDE system. In this case, the neural networks are used to eliminate the need of Sherwood, Schmidt and Reynolds correlations related to mass transfer coefficient estimation. Therefore, UDEs have shown to be widely useful in the context of process engineering by providing flexible techniques for parameter estimation and modeling and effectively reducing experimental costs.

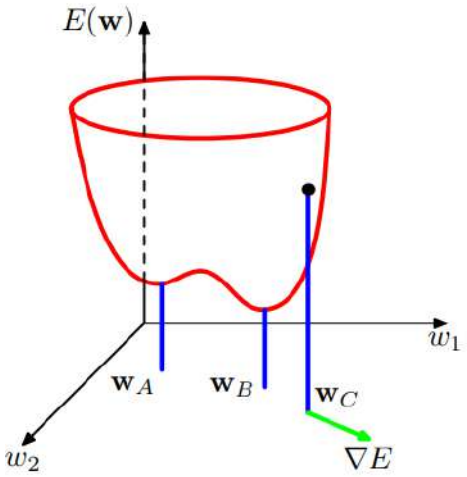
2.2.3 Neural Network Training and Optimization

The process of training and optimization of a neural network comprises of finding the right parameters that minimize an error function. The error function can be defined by the sum of squares error between the expected value t_n and the output of the model $y(x_n, w)$ as shown in equation (31):

$$E(W) = \frac{1}{2}\sum_{n=1}^{N} \|y(x_n, w) - t_n\|^2 \tag{31}$$

The error function and its parameter dependency can be geometrically represented by a surface sitting over a weight space with local (w_A) and global (w_B) minima and a gradient vector ∇E at a given parameter dependent state (w_C) (Bishop, 2006).

Figure 2 – Geometric representation of an optimization problem



Source: (Bishop, 2006)

Assuming the error function is smooth and continuous in all the space of the weights' domain, by this representation the vector of weights and bias that give the smallest error will be available at a point where the gradient of the error function equals zero. As the ANN structure has a highly nonlinear dependence on its parameters, there will be many points where the gradient of the error function decreases to very small number or zero varying from local minima, global minima and saddle points (Bishop, 2006).

Analytically finding a solution to the error function minimization problem may be a hard task and numerical algorithms developed to the solution of nonlinear problems are used in order to tune and optimize a neural network. Popular ones are based on gradient descent approach where gradient information on a given state of the model is used to update the weights and bias in its negative direction. Updates are made in small steps where its magnitude is given by a number often called learning rate (Bishop, 2006). The expression below (32) synthesizes the weight update based on learning rate for a gradient descent optimization algorithm.

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E(w^{(\tau)}) \tag{32}$$

Where η is the learning rate that indicates the magnitude of each optimization step, ∇E is the gradient of the error function in respect to each parameter and $w^{(\tau)}$ is the model weights and bias matrix of optimization iteration τ.

As stated before, an important part of most neural network training routines involves calculating the derivatives related to each parameter of the hidden layers. For that the efficient technique used is called error backpropagation and it involves a local

message passing scheme in which information is sent alternately forwards and backwards through a neural network (Bishop, 2006).

In general, the backpropagation algorithm is comprised of four steps. As presented by Bishop (2006) the first one is often called forward step, where it is performed the application of an input vector to the network and forward propagation through its hidden units and calculation of the activations and outputs units. In the next step the error needs to be evaluated for the output units using the expression (33).

$$\delta_k = y_k - t_k \tag{33}$$

The backpropagating step occurs when error δ_k, is used to calculate the respective errors of each hidden unit in before in the network, this is done by using the general backpropagation formula (34) derived from the chain rule of derivatives applied to error derivatives (35).

$$\delta_k = h'(a_j) \sum_k w_{kj}\, \delta_k \tag{34}$$

$$\delta_j \equiv \frac{dE_n}{da_j} = \sum_k \frac{dE_n}{da_k} \frac{da_k}{da_j} \tag{35}$$

Where a is the activation for layer k and j. The derivatives in respect to each weight are finally estimated as expressed in equation (36).The fourth step comprises of taking the derivatives estimated and updating the parameters of the neural network using an optimization algorithm.

$$\frac{dE_n}{dw_{ji}} = \delta_j z_i \tag{36}$$

For UDEs, the backpropagation of error needs to be evaluated through a numerical integrator of differential equation system. For this task, methods called adjoint are efficiently used (Allaire, 2015; Johnson, 2012). These methods present asymptotic computational costs that do not grow with the number of state variables and parameters and as numerical forward sensitivity analysis and therefore are considered a robust option on large parameters models (Rackaukas, 2021).

Nowadays tools for easy implementation, training of neural network, integration and backpropagation through differential equations methods have been developed in high level programming languages and for that enabling the execution of this work (Paszke, et al., 2019; Poli, M., et al., 2020).

3 METHODOLOGY

In this section the methodology used to derive and build the model studied will be described. Details about the tools used, the training routine, framework for neural networks implementation, error function used, and optimization strategy adopted will be discussed.

3.1 PBM WITH METHOD OF MOMENTS

As stated before, the objective of this work is to implement a universal differential equation approach to model paracetamol dissolution in ethanol solution using Kim et al. (2023) as a benchmark and a neural network as a universal approximator for the kinetic term related to dissolution of crystals.

Following Kim's (2023) work a one-dimensional population balance model can be expressed by the yet presented equation (7).

$$\frac{dn}{dt} + G\frac{dn}{dL} = 0 \tag{7}$$

Where the $\frac{dn}{dt}$ term denotes the rate of change of a number density of crystals within time [#/(cm s kg of solvent)] , G denotes the growth or dissolution rate of crystals [cm/s], and the $\frac{dn}{dL}$ term is the rate of change in the number density of crystals in relation to a characteristic size L [#/cm² kg].

The assumptions needed to derive expression (7) from the general PBM expression (23) presented before are the following: 1) Crystals have uniform shape across different sizes. 2) The growth/dissolution rate does not depend on crystal size. 3) Growth rate does not have dispersion. 4) Crystal density is constant 5) Agglomeration and breakage of crystals can be neglected.

As this model aims to simulate only dissolution experiments, the rate of appearance of new crystals will not be considered, also for the same reasons Kim et al. (2023) neglected the disappearance of crystals, the same assumption will be adopted. Hence the following boundary conditions (37) and (38) can be applied:

$$n(t = 0, L) = n_0 \tag{37}$$

$$n(t, L = 0) = 0, \quad S < 1 \tag{38}$$

For solving the PBM, the moments transformation general expression (24) was applied to transform the PDE problem into a set of ordinary differential equations of

moments. The expression for the set of ordinary differential equations (39) is shown below:

$$\mu_j(x_e, t) = \int_0^\infty nL^j dL \tag{24}$$

$$\frac{d\mu_j}{dt} = 0^j B_0 + jG\mu_{j-1}, \ \ j = 0, 1, 2, \ldots \tag{39}$$

Considering no variation in total number of crystals the zeroth moment is converted to (40):

$$\frac{d\mu_j}{dt} = 0^j B_0 + jG\mu_{j-1}, \ \ j = 0, 1, 2, \ldots \tag{40}$$

Where the B (birth rate, or nucleation rate) equals to zero.

As denoted by Kim et al. (2023), a mass balance based on the third moment and its relation to the total volume of crystals (41) was employed to estimate the concentration at a given time. Differentiating the mass balance equation, the rate of change of concentration depending on the second moment and the dissolution rate is obtained as described in (42).

$$C(t) = C_0 - k_v \rho \mu_3(t) \tag{41}$$

$$\frac{dC}{dt} = -k_v \rho \frac{d\mu_3}{dt} = -3k_v \rho G \mu_2 \tag{42}$$

The last variable that needed to be defined is the relative supersaturation that was denoted by equation (13) and for that a correlation for the solubility of paracetamol in ethanol solution as function of temperature was used. This correlation was proposed by Kim et al. (2023) and is described in equation (43).

$$C_s = -8.707 + 9.699 \cdot 10^{-2} T[K] - 3.610 \cdot 10^{-4} T[K]^2 + 4.590 \cdot 10^{-7} T[K]^3 \tag{43}$$

With all the assumptions and equations presented we have the following ODE system:

$$\frac{d\mu_0}{dt} = 0$$

$$\frac{d\mu_1}{dt} = G\mu_0$$

$$\frac{d\mu_2}{dt} = 2G\mu_1$$

$$\frac{d\mu_3}{dt} = 3G\mu_2$$

$$\frac{dC}{dt} = -3k_v \rho G \mu_2$$

$$S = \frac{C}{C_s}$$

$$C_s\,[g/g] = -8.707 + 9.699 \cdot 10^{-2} T[K] - 3.610 \cdot 10^{-4} T[K]^2 + 4.590 \cdot 10^{-7} T[K]^3$$

3.2 NEURAL NETWORK AND OPTIMIZATION

3.2.1 Neural Network Programing and UDE integration tools

Inspired by Lima et al. (2023), an UDE framework was applied to the population balance model using a neural network as an universal approximator for the dissolution rate G. The neural network model was developed in Python using the deep learning library, Pytorch (Paszke, A. et al. 2019) and the neural differential equations library TorchDyn (Poli, M. 2020).

TorchDyn library is primarily focused on using neural networks to solve differential equation systems by applying neural network as solvers. However it also contains tools for hybrid systems and the great advantage is the ease of use for developing training loops that require backpropagation through the numerical integrator step. The library was developed to work in communication with Torch's deep learning resources and for that it became a convenient tool for the execution of this work.

Applying the neural network to the kinetic term for crystal dissolution rate expression (44) is obtained.

$$G = NN_D(S(T), C, \theta_{NN}) \tag{44}$$

$$\frac{d\mu_0}{dt} = 0$$

$$\frac{d\mu_1}{dt} = NN_D(S(T,C), C, \theta_{NN}) \cdot \mu_0$$

$$\frac{d\mu_2}{dt} = 2 \cdot NN_D(S(T,C), C, \theta_{NN}) \cdot \mu_1$$

$$\frac{d\mu_3}{dt} = 3 \cdot NN_D(S(T,C), C, \theta_{NN}) \cdot \mu_2$$

$$\frac{dC}{dt} = -3k_v\rho \cdot NN_D(S(T,C), C, \theta_{NN}) \cdot \mu_2$$

$$S = \frac{C}{C_s}$$

$$C_s\,[g/g] = -8.707 + 9.699 \cdot 10^{-2} T[K] - 3.610 \cdot 10^{-4} T[K]^2 + 4.590 \cdot 10^{-7} T[K]^3$$

Where $NN_D(*)$ is the neural network with inputs supersaturation $S(T,C)$, concentration C [g solute/g solvent], and parameters weights and bias θ_{NN}.

The variables chosen needed to have the same dependent variables of the phenomenological model in order to enhance its physical meaning. However, as the temperature measured had a considerable amount of noise in the experimental data set, it was preferred to use the supersaturation calculated by the given correlation as it carried over the temperature information in it. The input variables of the neural network were scaled using a min-max scale aiming to eliminate the effects of variables order of magnitude over the result. The expression used for inputs normalization can be found below (45).

$$x_{norm} = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \tag{45}$$

3.2.2 Initial value problem integration and data preparation

The training routine developed was focused on optimizing the neural network such as it predicts the kinetic coefficient that reduces the error between the integrator output and the expected values for each time point of the experiments made by Kim et al. (2023). Initially the plan was to use the same error function used by Kim et al. (2023) for parameter estimation, but later the expected and predicted moments were also included in the error function to better fit the model.

The differential equations system of the PBM was solved by a numerical integrator embedded in Torchdyn (torchdyn.numerics.odeint). The method of integration used was the Runge Kutta of 4th order and for the initial value problem the value of each moment was calculated based on experimental conditions for dissolution experiments provided by Table 1 in Kim et al. (2023).

The moments' initial values were calculated using mainly 3 variables: 1) the initial average length of crystal, 2) the mass of seed for dissolution, 3) the number of crystals obtained by estimating each crystal as a spherical volume with diameter equal to the average length of crystals. The initial concentration for each experiment was also provided in the article.

The expressions used to calculate each moment are denoted on the following equations (46 - 50) and are derived from the definition of the moment transformation. The initial crystal length is the average of widest and narrowest mesh of sieves from

the section where the seed crystal was taken. The number of crystals is estimated assuming each crystal has an spherical volume with diameter equals to the average crystal length. Each moment is related to a certain property of the size distribution per mass of solution. For that the first moment is the number of particles, the second moment is total length of particles, the third is the total area of particles and the fourth is the total volume of particles all of them calculated from the initial length of particles and divided by the mass of solution. All experiments were performed with 100 g of solution

$$L_{init\,[\mu m]} = \frac{(L^{+} + L^{-})}{2} \tag{46}$$

$$n_{c\,[\#]} = \frac{\left(\frac{m_{seed}}{\rho}\right)}{\frac{4}{3}\pi\left(\frac{L_{init}\cdot 10^{-4}}{2}\right)^{3}} \tag{47}$$

$$\mu_{0\,init\,[\#/g]} = \frac{n_c}{100} \tag{48}$$

$$\mu_{1\,init\,[cm/g]} = \frac{n_c \cdot L_{init} \cdot 10^{-4}}{100} \tag{49}$$

$$\mu_{2\,init\,[cm^2/g]} = \frac{n_c \cdot (L_{init} \cdot 10^{-4})^2}{100} \tag{50}$$

$$\mu_{3\,init\,[cm^3/g]} = \frac{m_{seed}}{100 \cdot \rho \cdot k_v} \tag{51}$$

Where L_{init} is the mean crystal size used in each experiment and given by Kim et al. (2023). n_c is the number of crystals estimated by the mass seed, crystal density and estimating spherical crystals with length equals to L_{init} .

Each experimental condition is listed in Table 1 and Table 2 summarizes the initial values used:

Table 1 – Experimental initial conditions collected from (Kim et al., 2023)

Experiment	Experimental initial conditions from Kim et al. (2023)				
	$L_{init}[\mu m]$	$m_{seed}[g]$	$T[°C]$	n_C	$C_{init}[g/g]$
1	462.5	4.8	10	7.34E+04	0.1425
2	462.5	4.85	15	7.41E+04	0.1539
3	327.5	4.9	20	2.11E+05	0.1672
4	650	4.9	20	2.70E+04	0.1672
5	462.5	5.1	30	7.80E+04	0.2018
6	650	4.8	10	2.64E+04	0.1425
7	462.5	4.9	20	7.49E+04	0.1672

Source: (Own Authorship with data from Kim et al., 2023)

Table 2 – Initial values applied in numerical integrator

Experiment	Initial values for integration				
	$\mu_0[\#/g]$	$\mu_1[cm/g]$	$\mu_2[cm^2/g]$	$\mu_3[cm^3/g]$	$C_{init}[g/g]$
1	733.6756	33.9325	1.5694	0.0477	0.1410
2	741.3180	34.2860	1.5857	0.0482	0.1535
3	2,109.4061	69.0830	2.2625	0.0487	0.1683
4	269.8075	17.5375	1.1399	0.0487	0.1672
5	779.5303	36.0533	1.6675	0.0507	0.2018
6	264.3012	17.1796	1.1167	0.0477	0.1425
7	748.9605	34.6394	1.6021	0.0487	0.1672

Source: (Own Authorship with data from Kim et al., 2023)

For the error evaluation, the moments for each time step were calculated using the phenomenological model denoted by Kim et al. (2023), with the symbolic integration library CasADi (Andersson et al., 2018) also coded in Python. The concentration profiles for each experiment were obtained directly from experimental

data retrieved from each experiment spectra obtained through the attenuated total reflectance-Fourier transform infra-red technique (ATR-FTIR).

The four moments estimated by phenomenological model and the experimental concentration data were concatenated on a target value matrix that was later used for error evaluation. Kim et al. (2023) uses only concentration information over the experiments as metric to evaluate the optimization error. However, following Lima et al. (2023) the four moments were included in the neural network training error evaluation for the best model fit. Throughout the training sessions made with different hyperparameter, evidence was found that using the four moments on error evaluation could significantly improve the network training.

During training tests, it was noticed that some of the pair temperature and concentration would output a supersaturated state (S > 1), which should produce an inversed signal dissolution rate. Because of these states a boundary condition was introduced to the model where whenever a supersaturated conditions was met the dissolution rate would become 0. This boundary condition was employed in order to keep the model physical meaning and simplify the training problem.

3.2.3 Error evaluation

For the training routine, backpropagation and parameters update at each optimization step, the error function used was defined by equation (52). Being the mean root of relative squared error over all points of the output matrix in comparison to expected values matrix.

$$Error = \frac{1}{K \cdot N}\left(\sum_{k=0}^{K}\sum_{n=0}^{N}\sqrt{\frac{(y_{nk} - \bar{y}_{nk})^2}{\bar{y}_{nk}{}^2}}\right) \tag{52}$$

Where y_{nk} and $\bar{y}_{nk}$ represents the estimated and expected value for the time step n and variable k. N is the length of evaluated matrix and K is the total number of variables used to compute error. On the first model training routine only concentration was computed in error function, in later models training the four moments were added, thus K became assumed the value of 5, the four moments and concentration variable.

This custom loss function was particularly useful as each dependent variable (the four moments and concentration) had a different order of magnitude and thus it was necessary to avoid bias of reducing the error of greater magnitude variables in prejudice of lesser magnitude ones. Using this custom function, the error calculation

works as a normalization step as it has the magnitude similar to the relative error calculation.

The error was calculated comparing each point of the whole matrix of outputs against expected values matrix and the mean of all error points was then used as total error for the batch experiment evaluated.

3.2.4 Network initialization, forward pass and training

As deep learning training algorithms are iterative, an initial condition for the model needs to be given to start algorithm optimization. As stated by Goodfellow (2016), training a deep learning model is a sufficiently difficult task that most algorithms are strongly affected by choice of initialization. However, the common understanding about neural network parameters initialization is still very primitive and thus most initialization techniques come from heuristic rules. A common heuristic is that the initial values “must break symmetry” and for that one neuron must not have two inputs functions with equal parameters as the gradient will be computed equally at each backpropagation step (Goodfellow, 2016). Therefore, commonly used initialization involves random numbers generation. The one used in this work is based on the Glorot initialization (Glorot and Bengio, 2010) where the parameters initialized are randomly generated from a Gaussian distribution domain with 0 mean and standard deviation (σ) described by equation (54):

$$\sigma = \sqrt{\frac{2}{m+n}} \tag{53}$$

Where m is the number of inputs for the neural network and n is the number of outputs. As the neural network intention is to mimic the behavior of the kinetic rates of dissolution, the inputs of the network used for this model were the supersaturation and the concentration at each step. Thus, the standard deviation suggested for the gaussian distribution was 0.81. Larger weights for neural network initialization help “symmetry breaking” and avoid losing signal trough the network but can also cause big enough gradients that cannot be computed (exploding gradients) and thus ruining an optimization experiment (Goodfellow, 2016). To avoid the exploding gradients problem, tests were made with different standard deviations for the gaussian initialization and for the model presented in this work a smaller magnitude standard deviation was chosen, as it made the convergence and gradient descent faster.

Parameters related to the training conditions are called hyperparameters according to Goodfellow (2016), these are settings defined outside the training loop that are used to control algorithms behavior. The values of the hyperparameters are not adapted by the learning algorithm but rather impose the conditions which the training will occur. To tune the parameters, training data must be split into 3 different sets, training samples, validation samples and test samples. The first one is used to update the network parameters, the second set is used to evaluate the generalization of the model during or after training and is not observed by the optimization algorithm, but it is used to guide the hyperparameter selection. Finally, the test set is used to evaluate model generalization error, and that set has no influence on the optimization and training routines.

For the experiments conducted in this work, the main hyperparameters chosen for training routine were the initial learning rate, the optimizer, the number of training iterations (epochs), a random number generator seed and the network structure. Data used to train the model was split following Kim Y. et al. (2023) selection. Experiments 1 to 5 were used as training sets, with experiment 1 being used as the validation set and for the test sets experiments 6 and 7 were used for model evaluation.

To adjust training hyperparameters and aiming to achieve best performance, several tests were conducted varying each hyperparameter from the training optimization strategy implemented. The changes in hyperparameters initially began following a trial-and-error procedure and were conducted as such for long as an understanding on how each hyperparameter affected the overall result. The end of such a procedure culminated on a test model in which the performance on training data was evaluated and considerations were taken based on knowledge acquired through test routines.

Examples of changes made on hyperparameters with trial-and-error experimentation and the observations that were noticed in each test include: 1) Experimentation with different optimizers as SGD, Adam and L-BFGS showed that in general gradient descent with adaptative moments optimizers as Adam and its variations yields better results with stable loss decay and robust performance that resisted to chaotic parameters update, exploding/vanishing gradient problems and local minima stagnation; 2) Changes in starting learning rate denoted the necessity of regular updates within training epochs by decreasing its magnitude and thus increasing precision of optimization steps that helped to avoid oscillating errors; 3) Different

initialization techniques exposed that reducing the magnitude of initial parameters also produced smaller error on initial epochs but whenever initial parameters approached to 0 the model became more susceptible to local minima and vanishing gradients; 4) Varying total epochs number exposed that not always increasing the number of epochs also increases model performance, therefore indicating the need of implementing data segmentation and an early stopping algorithm that stopped training whenever a number of epochs with no improvement in validation error exceeded a criteria called patience. This criteria is considered a hyperparameter that affects the training routine and needs to be tuned (Goodfellow, 2016).

With the knowledge acquired a test model was developed and the results evaluated revealed further improvements for the training algorithm. For the test model the training routine was set to iterate over 30000 epochs (18 hours of processing) and used AdamW (Loshchilov and Hutter, 2019) for optimization steps with starting learning rate of 1×10^{-4} and learning rate updates scheduled on epochs 10, 1000, 4000 and 10000 with update coefficient as 1×10^{-1}. The learning rate update schedule was set to avoid training stagnation, fine tune the parameters through the training routine and avoid chaotic parameters updates as the training proceeded. Patience, the early stopping criteria that denotes the maximum number of epochs without improvement in validation error, was set to 50 epochs. Whenever the model exceeded 50 epochs without lowering validation error, the training was terminated and the model that yielded the best results was saved.

Training session lasted over all the 30000 epochs and the estimated profiles for each training experiment can be verified on the following chapter as also the training and validation error curves plotted for the run.

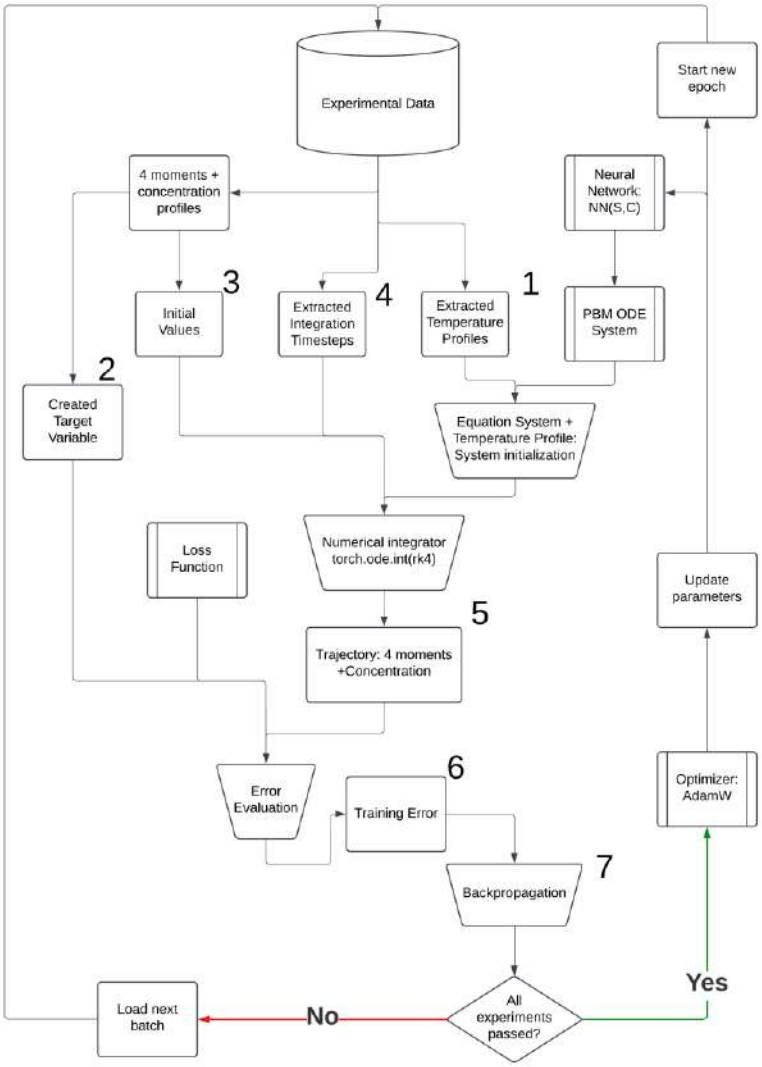
The training loop consisted of two distinct steps, the first is a forward and backward pass and the second is the optimization step. The forward and backward step consists in using the model to predict each moment and concentration following by evaluation of error for all the training and validation sets and backpropagating the training error. The optimization step consists of using the computed gradients to update neural network parameters and start a new training iteration. Each training iteration where the model passes through all the data points is called an epoch. An epoch is one hyperparameter that measures the training time and should be tuned for best model fit.

For each experimental batch the forward and backward pass was structured following the steps listed in Figure 3. For better understanding these steps will be explained as follows: 1) Batch temperature profile and the hybrid ordinary differential equations system for the PBM was initialized; 2) The four moments and the concentration at each time step were stored on a target variable; 3) Initial values were taken from the first time step stored on initial values variable; 4) Integration time span was acquired from the length of batch vector; 5) A variable called trajectory was created and stored the results for the torchy.numerics.odeint function with arguments of initial values vector obtained in step 3 and integration time vector obtained in step 4; 6) Loss was evaluated comparing target variable and trajectory variable; 7) the backward pass was called for training error and gradients for parameters update were stored until optimization step. The sum of training errors and the sum of validation errors of each epoch was stored for comparison afterwards. Optimizer step was taken only after gradients from all experiments were computed and stored.

For better regularization of the neural network (i.e. reduction of the validation error without reducing the training error) an early stopping routine was also implemented where the training algorithm would be terminated, saving the best model, the one which presented the minimum recorded error over a predetermined number of epochs with no improvements on validation error. This number of epochs is often called patience and is considered another hyperparameter to be tuned (Goodfellow, 2016).

Another technique used for neural networks regularization was the mini-batch data training procedure. As denoted in the results section, examining the training sets results made with different hyperparameter and training routines, suggestion was found that introducing smaller sets for error evaluation could potentially benefit the results of optimization. For that reason training and validation data sets were splitted in 5 different parts composing four parts of 24 data points each and 25 data points for the last one and error was evaluated independently and gradients for all the segments were stored in backpropagation step before optimization step was taken.

Figure 3 – Flowchart of experimental forward and backward pass



Source: (Own Authorship)

4 RESULTS AND DISCUSSION

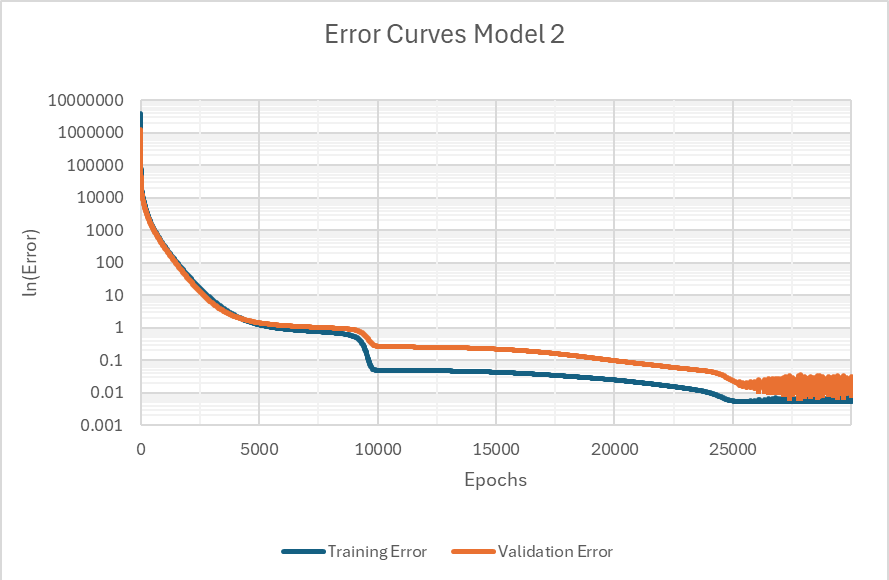
4.1 MODELS DEVELOPED

After multiple training sessions, two final models were adjusted and evaluated. The results and discussion will be presented in this section as well as recommendation on what to improve in further investigations. The test model results will be presented as a conducting line of thought on how decisions were made to achieve end-results.

4.1.1 Model 1: Test model

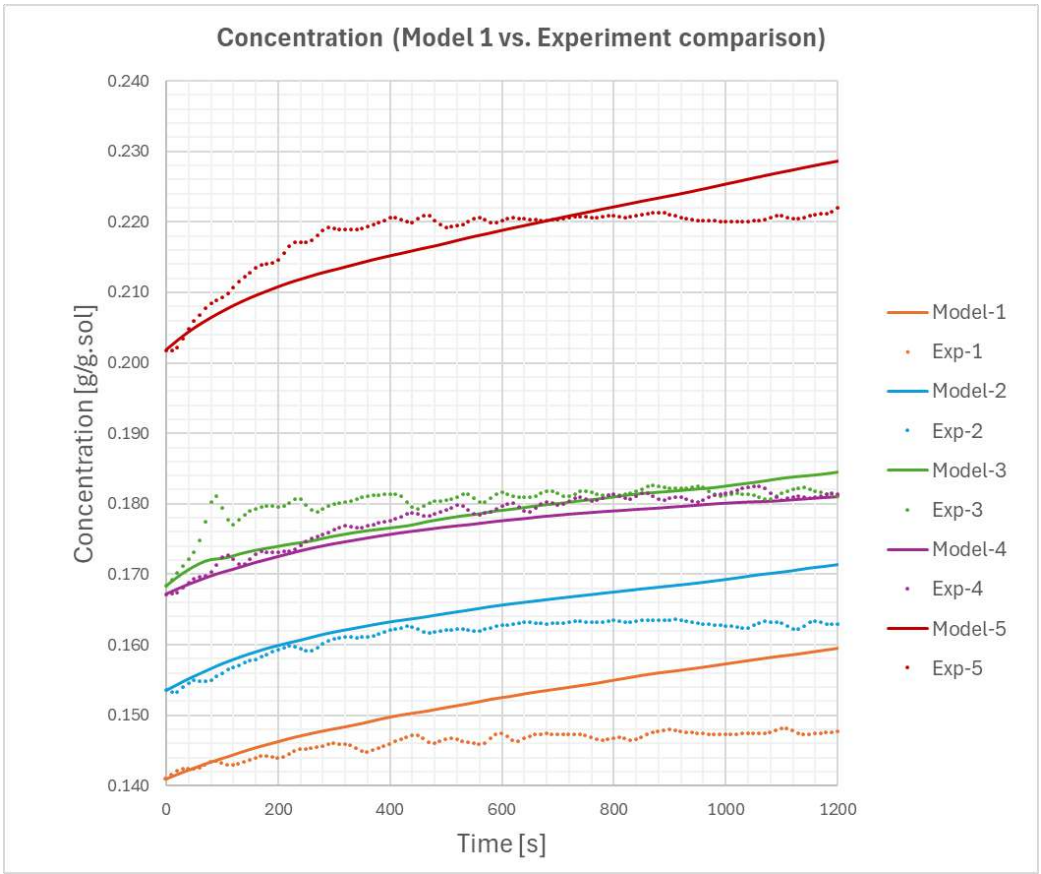
The first model was the test model used to optimize training routine and adjust hyper parameters. This model was training using a simple sum of squared errors as loss function in the training routine and the optimization strategy did not include the four moments in error calculation. The training strategy did not use data segmentation besides the usual training and validation sets. The neural network trained was composed of two hidden layers with 20 neurons each and used the hyperbolic tangent as an activation function. The model parameters were initialized with a gaussian distribution with 0 mean and standard deviation of 0.0055, a seed number for random number generation was used to have reproducibility. The results can be assessed in Figures 4, 5 and 6.

Figure 4 – Error learning curves for model 1



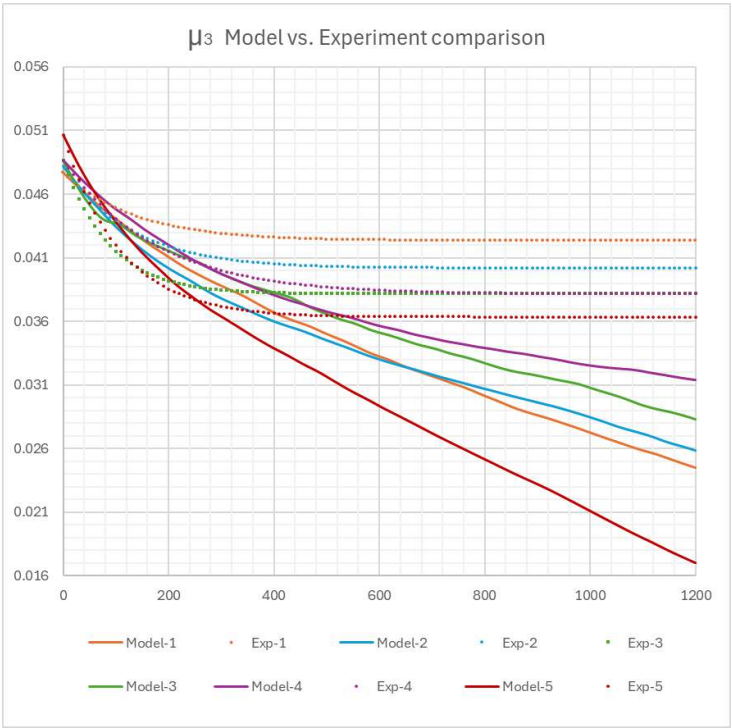
Source: (Own Authorship)

Figure 5 – Estimated concentration profiles for training sets by model 1



Source: (Own Authorship)

Figure 6 - Estimated third moment profile for training sets by model 1



Source: (Own Authorship)

From the results obtained, two important considerations were taken. The first one was that the model tended to drift away from experimental data with each time step. This could be related to the model learning algorithm computing error and calculating the gradient for the entire time set instead of learning and computing error with different time zones and experiment maturity frames. The second consideration was that although the model has taken several epochs to achieve its best fit, error at the end of each experiment was still big and this error behavior on later time-steps could be observed as well on the four moments estimated. Hence indicating that neglecting the four moments on error calculation could be compromising the training routine.

Another important observation was that although experiments 3 and 4 seemed to fit well with experimental data all the other experiments apparently had a worst fit including the validation set that presented the worst performance of all 5 training experiments. Due to the quality of the model observed on training sets the test sets were not evaluated for this model.

4.1.2 Model 2: Supersaturation constrained model

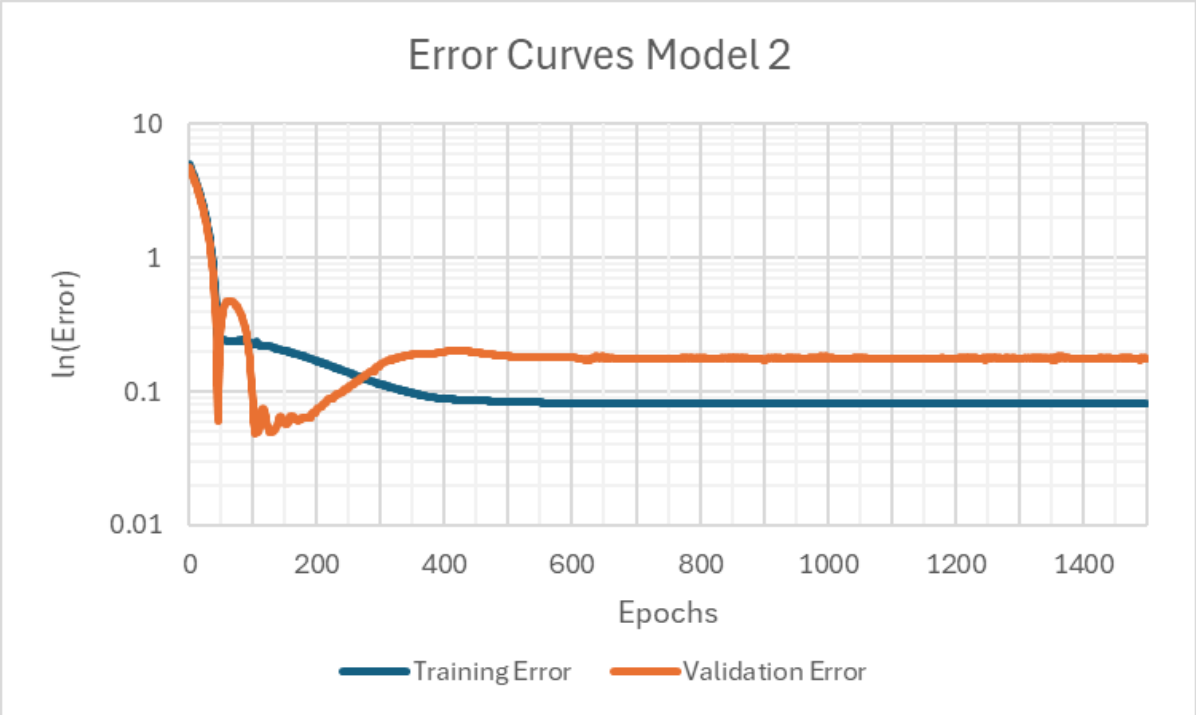
Based on these observations, the training routine was modified and model 2 was developed employing a data segmentation of each experimental set in five parts of equal length for error evaluation and backpropagation. The loss function was modified to include the four moments and calculate the average of relative squared error. A simpler neural network structure was proposed using one hidden layer with 30 neurons aiming for faster training as a smaller number of parameters should be updated and evaluated at each iteration. Aiming to further reduce the training time, the standard deviation for the initializing parameters was further reduced to 7.5×10^{-4}. The initial learning rate was also reduced to 1×10^{-5} and the learning rate update schedule was changed by removing the initial update at 10 epochs. No changes were made to the optimizer algorithm or the number of total epochs nor the patience for early stopping.

The last change in comparison with the previous model was due to the further examination of the input data set, at given points specially at the end of each experimental run, it was noticed that the experimental concentration was above equilibrium concentration denoted by equation (43), thus indicating a supersaturated state. To avoid such states on the input variable and constraint the problem to

undersaturated conditions a constraint was applied to the PBM denote kinetic term of dissolution equal to zero whenever relative supersaturations was greater than 1.02. A 0.02 tolerance over the supersaturation boundary conditions was adopted as concentration experimental data presented an error, inherited from Infra-Red spectra processing, that would be propagated to supersaturation and thus for better profiles fit and simplicity of model this error was considered on boundary condition.

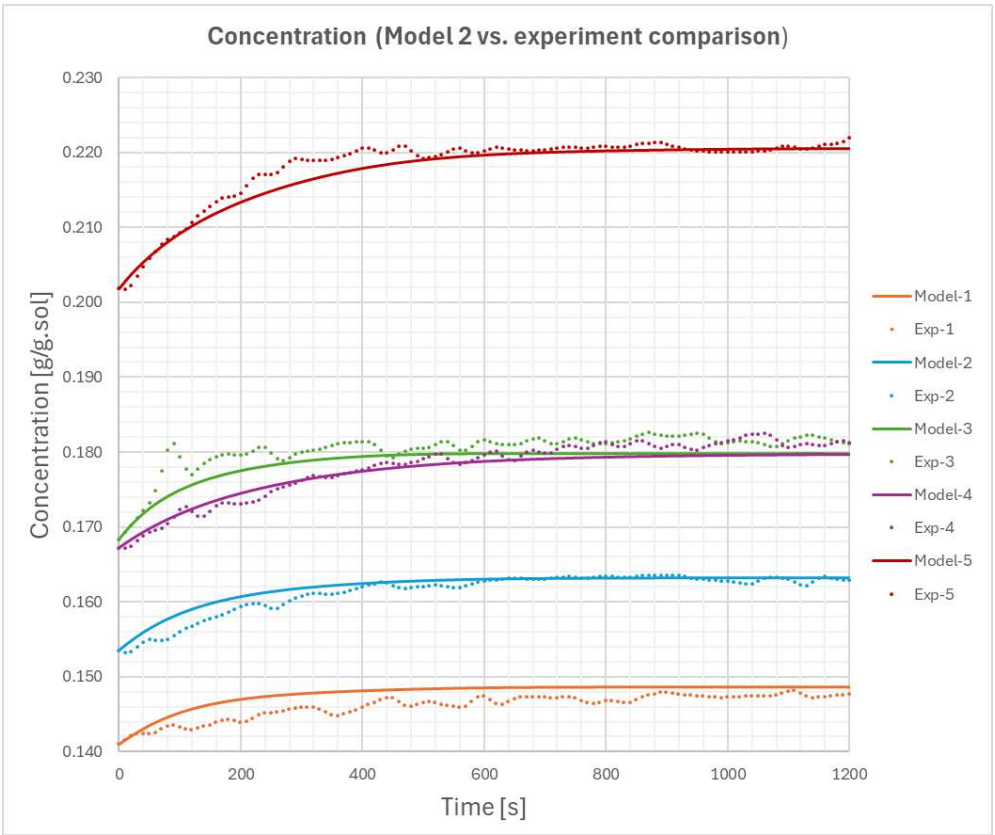
With all these modifications, great improvements were noticed in model performance and training routine. The model reached its best solution at 1500 epochs approximately 90 minutes from training initialization. The loss curves and comparisons between training sets are presented in Figures 7 and 8. Good fits were obtained for all training sets.

Figure 7 – Error learning curves for model 2



Source: (Own Authorship)

Figure 8 - Estimated concentration profile for training sets by model 2

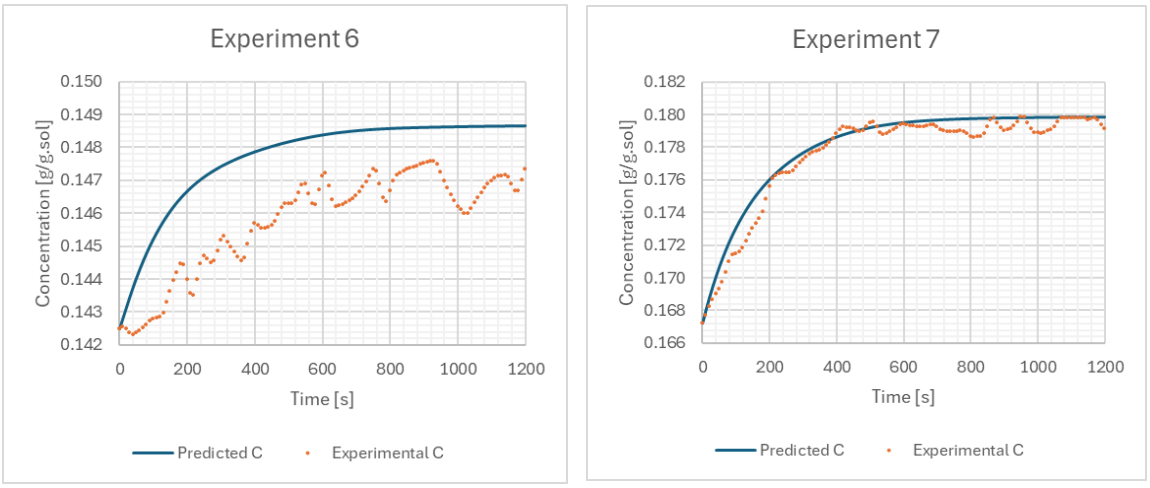


Source: (Own Authorship)

Model's performance on all experiments was improved and the training procedure became less expensive with hyper-parameters changes made.

Evidence of good generalization were found for the other 4 experiments, model 2 performance was evaluated in comparison to test sets. The results can be found in figure 9:

Figure 9 – Estimated concentration profiles for test sets by model 2

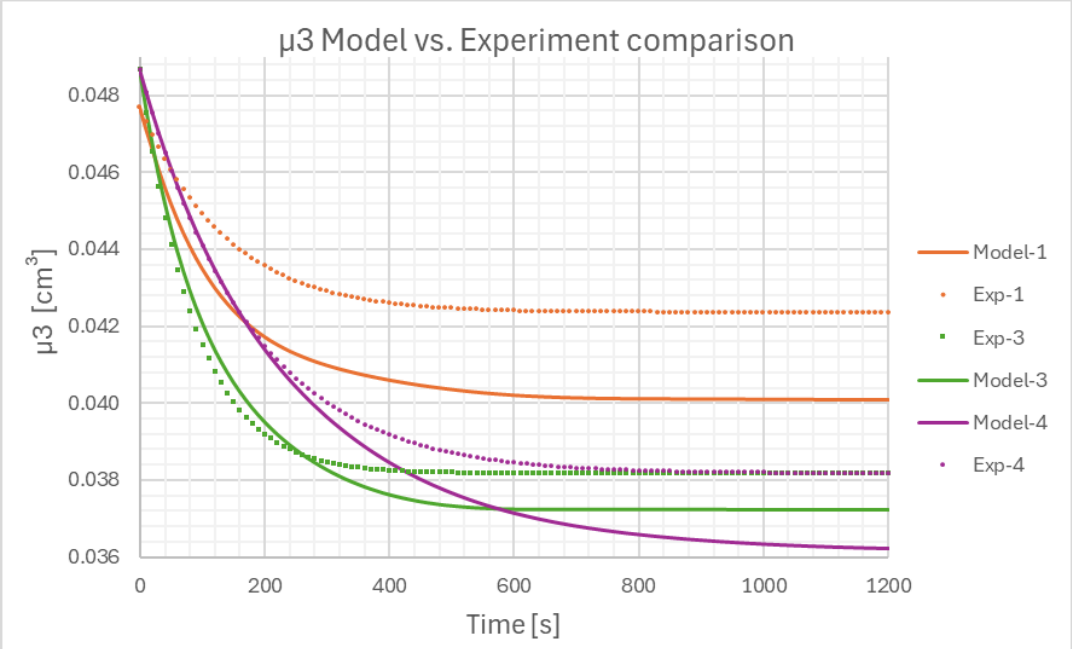


Source: (Own Authorship)

As denoted, the model follows the experimental profile for the test sets showing a good generalization capability on a data set which the neural network was not exposed before. By graph observation experiments apparently presents greater error. This could be explained by the expected final concentration being on the edge of a possible confidence interval. Experiment 1, that presents final concentration and profile similar to experiment 6, was used for validation and thus the model could have developed a worst generalization for such conditions.

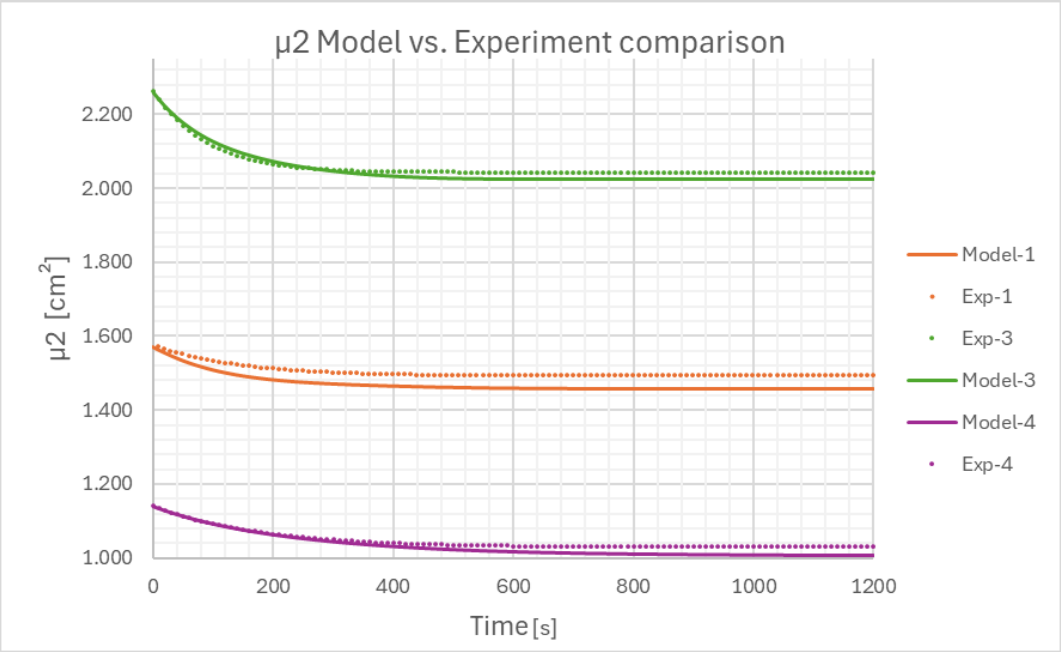
The four moments profiles also followed experimental data behavior. However, a considerable amount of error was still present on final third moment estimation. Other moments as second and first showed good correlation between its estimated and expected values hence reinforcing the importance of using crystal information on parameters optimization for crystallization models.

Figure 10 – Estimated third moment profiles for test sets by model 2



Source: (Own Authorship)

Figure 11 – Estimated second moment profiles for test sets by model 2



Source: (Own Authorship)

Figure 12 – Estimated first moment profiles for test sets by model 2

Source: (Own Authorship)

4.2 MODEL PERFORMANCE METRICS

Model 2 visually presented the best fit to experimental data. However, no performance metric was presented, in this section the total error will be evaluated as also the coefficient of determination for each experiment. The average relative squared error was calculated following equation (53). The error for each experimental set and each estimated variable is presented on the following Table:

Table 3 – Error between experimental and estimated values by model 2

Experiment	Mean Relative Squared Error				
	$\mu_0[\#/g]$	$\mu_1[cm/g]$	$\mu_2[cm^2/g]$	$\mu_3[cm^3/g]$	$C_{init}[g/g]$
1	1.06E-04	4.19E-04	2.34E-03	1.60E-04	1.06E-04
2	5.21E-05	2.07E-04	1.22E-03	4.48E-05	5.21E-05
3	1.74E-05	6.94E-05	4.30E-04	1.14E-04	1.74E-05
4	4.90E-05	1.94E-04	1.20E-03	4.52E-05	4.90E-05
5	2.87E-04	1.12E-03	7.38E-03	3.50E-05	2.87E-04
6	1.19E-05	4.72E-05	2.65E-04	2.00E-04	1.19E-05
7	7.11E-05	2.81E-04	1.74E-03	1.82E-05	7.11E-05

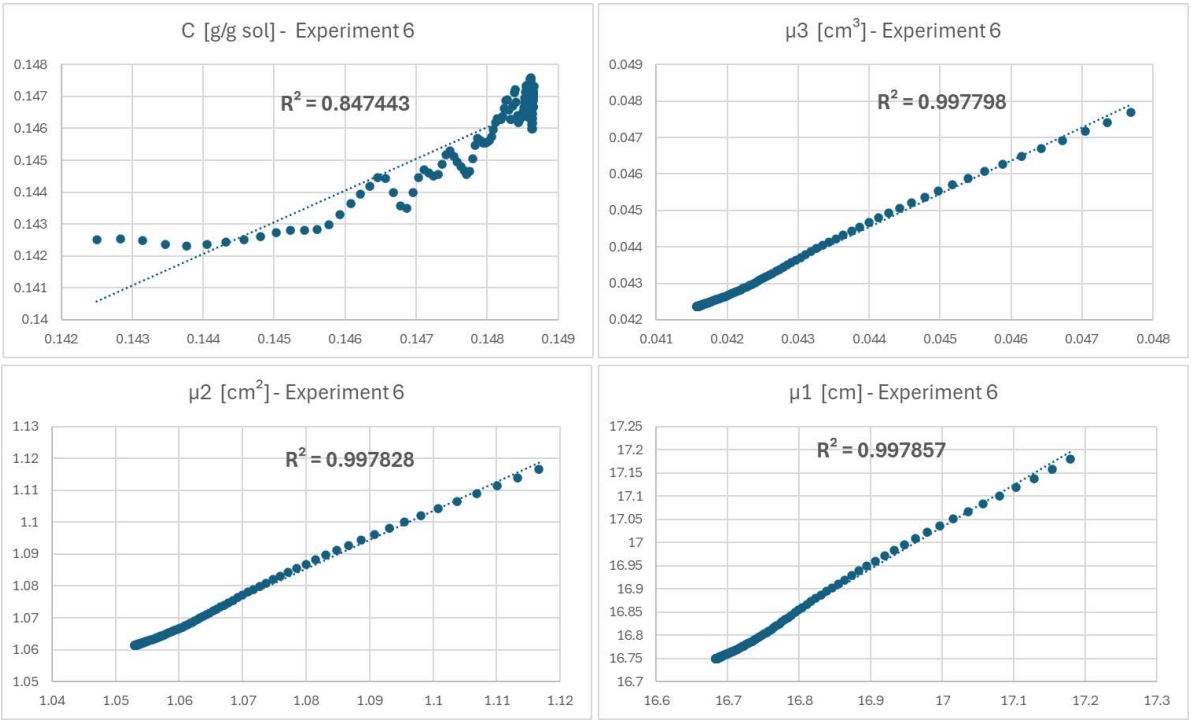
Source: (Own Authorship)

Evaluating the mean relative squared error for all experiments estimated values one can notice that for concentration estimates for experiments 1, 3 and 6 were far worse from the others. For experiments 1 and 6, this error can be explained as part of the training procedure conducting which the data select for error evaluation and gradient evaluation weakened generalization for low temperature plateaus and low mass dissolution experiments. For experiment 3, the error is due to experimental data having an abnormal early profile that could be related to experimental conducting.

Further observing the errors of experiment 5, one can infer that the model does not perform well for crystal information prediction on higher concentrations and temperatures, even though good regressions are obtained for concentration estimates.

Determination coefficients were estimated between experimental and predicted results for each estimated variable of the test sets. The results can be assessed in Figures 13:

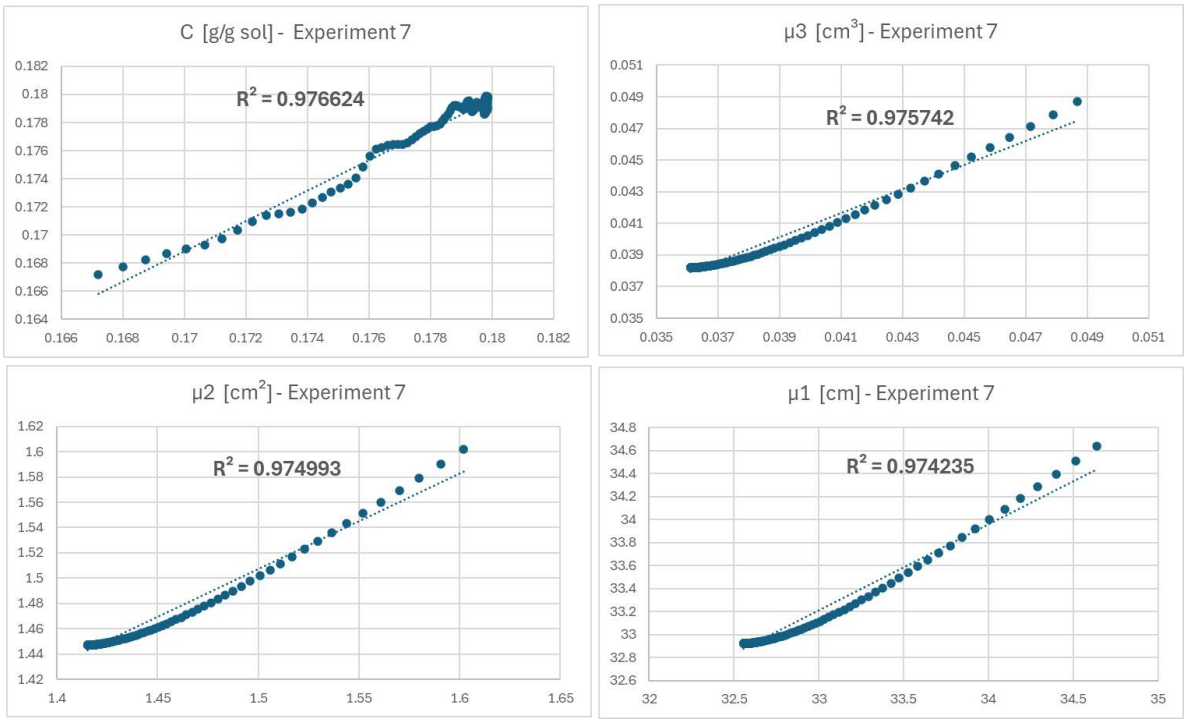
Figure 13 - Determination coefficients for experiment 6



Source: (Own Authorship)

For experiment 6, the model presented strong correlation, with the determination coefficients on all estimated moments, above 0.998 which indicates that the model was coherent with solid phase information. However, the determination coefficient for concentration estimates had the worst result 0.847, reinforcing the expectation of lower performance based on graphical analysis. It is expected that the model should perform better on moments estimation as the data used originated from a phenomenological model. The concentration, however, was originally retrieved from experimental data and therefore had characteristic noise which should generate increased model deviations. Therefore, aligned with expectations, the model presented a lower coefficient of determination for estimating concentration.

Figure 14 - Determination coefficients for experiment 7



Source: (Own Authorship)

Differently from experiment 6, experiment 7 presented better correlations for concentration and the determination coefficient indicated better results for all four moments. This can be due to the presence of less noise on the experimental set, and as similar conditions were exposed to the model by introducing experiments 3 and 4, in the training routines. As for experiment 6 no similar conditions were used on training but rather only used as validation, with experiment 1.

Experiment 7 concentration results demonstrate the best performance of the developed neural network on starting conditions that were not fed to the model before. Experiment 6 shows the good generalization capabilities as it presented a strong correlation with data and experimental conditions that the model was not trained on and also hinting for the limits of model extrapolation capabilities.

5 CONCLUSION

The aim of this work was to review the hybrid modelling framework denoted as Universal Differential Equations and apply it to a population balance model for paracetamol dissolution in ethanol using a neural network as universal estimator and experimental data from Kim et al. (2023) as benchmark for model training and optimization.

Two Neural Networks were developed for simulating the kinetic parameter of dissolution on differential equation models. Deep learning optimization strategies were studied and proposed for improvements on training and inference performance between training sessions.

A model with good generalization capacity was achieved and major improvements were made to training performance. The final model presented an accepTable correlation to most experimental data. Major deviations were due to the selection of training sets being overpopulated on experiments that had 20 ºC temperature plateau and thus the performance on experiments with conditions similar to validation set (Temperature 10 ºC) were worse.

When compared to test data, experiment 6 presented the worst model performance, as in the majority of experimental run the model had concentrations below the ones predicted by the model. The same behavior can be observed in experiment 1 used as the validation dataset as it had similar starting conditions (same temperature plateau and same initial concentration).

Even though some experiments presented greater deviation than others, in general the benefits of employing regularization techniques such as mini-batch and early stopping were noticed on greatly improved optimization procedure. By making the model simpler neural network convergence and optimum state was achieved faster with only 1500 iterations corresponding to approximately 90 minutes on available hardware.

Test model and final model comparison also reinforce the importance of crystal information on parameter estimation of crystallization systems. Good generalization was only possible when using both concentration and moments data as targets for error calculation. As for the first model, using only the concentration on the solution the error on moments estimation was one of the major causes for model's poor performance.

Different regularization techniques were used for model improvement and insights into neural networks training could be acquired. A simpler neural network performed better on training and inference steps This could be related to fewer number of parameters, easing the optimization task and still retaining regression capacity. Early stopping saved the model with the best regularization performance, avoiding overfitting issues. Neural network initialization inspired by Glorot and Bengio (2010) also provided low initial errors that did not cause vanishing gradients problem when generating parameters with standard deviations 1000 times smaller than recommended for initialization. Mini-batch training procedure seemed beneficial to the performance of trained models specially on later time-steps.

Overall model development was a success considering that constraint was made to achieve the result. Good regularization was achieved in both test and validation sets and average training loss remained low even in the worst performance experiment. For further studies, benefits of each regularization technique applied to hybrid-modeling should be evaluated to further cement the toolbox of techniques for hybrid modeling development.

BIBLIOGRAPHY

ALLAIRE, G. A review of adjoint methods for sensitivity analysis, uncertainty quantification and optimization in numerical codes. **Ingénieurs de l'Automobile**, v. 836, p. 33–36, 2015.

ANDERSSON, J. A. E. et al. CasADi: a software framework for nonlinear optimization and optimal control. **Mathematical Programming Computation**, v. 11, n. 1, p. 1–36, 11 jul. 2018.

BISHOP, C. **Pattern Recognition and Machine Learning**. Disponível em: <https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/?msockid=0db937648b57654a1b7323bf8a006450>. Acesso em: 11 nov. 2024.

BRAATZ, R. D.; HASEBE, S. Particle size and shape control in crystallization processes. **ResearchGate**, v. 98, n. 326, 2001.

BRANDÃO, A. L. T. et al. Modeling and parameter estimation of step-growth polymerization of poly(ethylene-2,5-furandicarboxylate). **Polymer Engineering and Science**, v. 58, n. 5, p. 729–741, 19 abr. 2017.

CARO, J. A.; WOLDEHAIMANOT, M.; RASMUSON, Å. C. Semibatch reaction crystallization of salicylic acid. **Chemical Engineering Research and Design**, v. 92, n. 3, p. 522–533, 1 mar. 2014.

CARPENTER, K. J.; WOOD, W. M. L. Industrial crystallization for fine chemicals. **Advanced Powder Technology**, v. 15, n. 6, p. 657–672, 2004.

FARIA, R. R. et al. Enhanced Hybrid Model for GasLifted Oil Production. **12th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2024**, v. 58, n. 14, p. 7–12, 2024.

GAHN, C.; MERSMANN, A. Brittle fracture in crystallization processes Part B. Growth of fragments and scale-up of suspension crystallizers. **Chemical Engineering Science**, v. 54, n. 9, p. 1283–1292, 12 abr. 1999.

GARSIDE, J. Industrial crystallization from solution. **Chemical Engineering Science**, v. 40, n. 1, p. 3–26, 1985.

GEORGIEVA, P.; DE, F. Neural NetworkBased Control Strategies Applied to a FedBatch Crystallization Process. **INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE**, v. 3, n. 3, 1 jan. 2006.

GEORGIEVA, P.; MEIRELES, M. J.; FEYO DE AZEVEDO, S. Knowledge-based hybrid modelling of a batch crystallisation when accounting for nucleation, growth and agglomeration phenomena. **Chemical Engineering Science**, v. 58, n. 16, p. 3699–3713, ago. 2003.

GLOROT, X.; BENGIO, Y. **Understanding the difficulty of training deep feedforward neural networks**. Disponível em: <https://proceedings.mlr.press/v9/glorot10a.html>.

GÓMEZ, J. et al. Crystal growth analysis in a membrane crystallization process using focused beam reflectance measurements (FBRM). **Desalination**, v. 573, n. 2023, p. 117201–117201, 1 dez. 2023.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Disponível em: <https://www.deeplearningbook.org/>.

GUEDES, M. et al. Polymorphism of Praziquantel: Role of Cooling Crystallization in Access to Solid Forms and Discovery of New Polymorphs. **Crystal Growth & Design**, v. 23, n. 2, p. 1247–1258, 2023.

HALFWERK, R. et al. Crystallization kinetics of lactose recovered at sub-zero temperatures: A population balance model combining mutarotation, nucleation and crystal growth. **Journal of Food Engineering**, v. 345, n. 2023, p. 111412–111412, 1 maio 2023.

HULBURT, H. M.; KATZ, S. Some problems in particle technology: A statistical mechanical formulation. **Chemical Engineering Science**, v. 19, n. 8, p. 555–574, 1 ago. 1964.

JOHNSON, S. G. **Notes on Adjoint Methods for 18.335**. Disponível em: <https://www.semanticscholar.org/paper/Notes-on-Adjoint-Methods-for-18.335-Johnson/75c033c779d9b357af2dab11f438fcc7b689e310>. Acesso em: 17 nov. 2024.

KIM, Y. et al. Modeling of Nucleation, Growth, and Dissolution of Paracetamol in Ethanol Solution for Unseeded Batch Cooling Crystallization with TemperatureCycling Strategy. **Ind. Eng. Chem. Res.**, v. 62, n. 6, p. 2866–2881, 2023.

LAURET, P.; BOYER, H.; GATINA, J. C. Hybrid modelling of a sugar boiling process. **Control Engineering Practice**, v. 8, n. 3, p. 299–310, 1 mar. 2000.

LAVANYA SHUKLA. **Designing Your Neural Networks**. Disponível em: <https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed>. Acesso em: 16 dez. 2019.

LEFEVER, R. A.; NASSAU, K. *Solid State Materials: Preparation and Properties, Vol. 1: Aspects of Crystal Growth*. **Physics Today**, v. 25, n. 11, p. 59–59, 1 nov. 1972.

LI, D.; LI, Z.; GAO, Z. Quadrature-based moment methods for the population balance equation: An algorithm review. **Chinese Journal of Chemical Engineering**, v. 27, n. 3, p. 483–500, mar. 2019.

LIMA, F. A. R. D. et al. Improved modeling of crystallization processes by Universal Differential Equations. **Chemical Engineering Research and Design**, v. 200, n. 1, p. 538–549, 2023a.

LIMA, F. A. R. D. et al. Statistical Analyses of a Population Balance Model of a Batch Crystallization Process. **Crystal Growth & Design**, v. 24, n. 1, p. 308–324, 12 dez. 2023b.

LOSHCHILOV, I.; HUTTER, F. Decoupled Weight Decay Regularization. **arxiv.org**, 14 nov. 2017.

MCDONALD, M. A. et al. Continuous reactive crystallization of βlactam antibiotics catalyzed by penicillin G acylase. Part I: Model development. **Computers & Chemical Engineering**, v. 123, n. 2019, p. 331–343, 2019a.

MCDONALD, M. A. et al. Continuous reactive crystallization of β-lactam antibiotics catalyzed by penicillin G acylase. Part I: Model development. **Computers & Chemical Engineering**, v. 123, n. 2019, p. 331–343, 6 abr. 2019b.

MORAES, M. G. F. D. *et al*. COOLING CRYSTALLIZATION: FROM MODELING AND CONTROL TO EXPLORING NEW POLYMORPHIC STRUCTURES. 1. ed. Rio de Janeiro: UFRJ/ COPPE/ Programa de Engenharia Química, 2023. p. 1-2.

MORAES, M. G. F. D. *et al*. Polymorphism of Praziquantel: Role of Cooling Crystallization in Access to Solid Forms and Discovery of New Polymorphs. Crystal Growth & Design, Rio de Janeiro, v. 23, n. 2, p. 1247-1258, jan./2023.

MORAES, M. G. F. et al. Optimal Control of Crystal Size and Shape in Batch Crystallization Using a Bivariate Population Balance Modeling. **IFAC-PapersOnLine**, v. 54, n. 3, p. 653–660, 2021.

MORITZ VON STOSCH et al. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. **Computers & Chemical Engineering**, v. 60, n. 2014, p. 86–101, 10 jan. 2014.

MORRIS, G. et al. Estimation of Nucleation and Growth Kinetics of Benzoic Acid by Population Balance Modeling of a Continuous Cooling Mixed Suspension, Mixed Product Removal Crystallizer. **Organic Process Research & Development**, v. 19, n. 12, p. 1891–1902, 9 dez. 2015.

MURPHY, K. P. **Machine learning : a probabilistic perspective**. Cambridge (Ma): Mit Press, 2012.

NOGUEIRA, I. B. R. et al. Using scientific machine learning to develop universal differential equation for multicomponent adsorption separation systems. **Canadian journal of chemical engineering/˜ The œCanadian journal of chemical engineering**, v. 100, n. 9, p. 2279–2290, 10 jul. 2022.

NÝVLT, J. **The Kinetics of Industrial Crystallization**. [s.l.] Elsevier, 1985.

PARK, S. et al. Minimum Width for Universal Approximation. **arxiv.org**, 16 jun. 2020.

PASZKE, A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. **arXiv.org**, 2019.

PAUL, E. L.; TUNG, H.-H.; MIDLER, M. Organic crystallization processes. **Powder Technology**, v. 150, n. 2, p. 133–143, fev. 2005.

POLI, M. et al. TorchDyn: A Neural Differential Equations Library. **arXiv.org**, 2020.

PRICE, C. J. Take some solid steps to improve crystallization. **Chemical Engineering Progress**, v. 93, n. 9, set. 1997.

RACKAUCKAS, C. et al. **Universal Differential Equations for Scientific Machine Learning**. Disponível em: <https://doi.org/10.48550/arXiv.2001.04385>. Acesso em: 10 nov. 2024.

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.

RANDOLPH, A. D.; LARSON, M. A. Transient and steady state size distributions in continuous mixed suspension crystallizers. **AIChE Journal**, v. 8, n. 5, p. 639–645, nov. 1962.

S. MYERSON, A. **Handbook of Industrial Crystallization | ScienceDirect**. Disponível em: <https://www.sciencedirect.com/book/9780750670128/handbook-of-industrial-crystallization>.

SAAD, M.; KAO, K.; KWON, J. S. Physicsinformed neural networks for hybrid modeling of labscale batch fermentation for βcarotene production using Saccharomyces cerevisiae. **Chemical Engineering Research and Design**, v. 179, n. 2022, p. 415–423, 2022.

SCHWAAB, M. et al. Nonlinear parameter estimation through particle swarm optimization. **Chemical Engineering Science**, v. 63, n. 6, p. 1542–1552, mar. 2008.

SHEKUNOV, B. YU.; YORK, P. Crystallization processes in pharmaceutical technology and drug delivery design. **Journal of Crystal Growth**, v. 211, n. 1-4, p. 122–136, abr. 2000.

SITI ZUBAIDAH ADNAN; SAMAD, A. Effects of different seed forms on crystal size distribution for seeded batch crystallization process. **Materials Today Proceedings**, v. 2023, 1 jun. 2023.

TAVARE, N. S. **Industrial Crystallization**. Boston, MA: Springer US, 1995.

TEIXEIRA, A. P. et al. Hybrid semi-parametric mathematical systems: Bridging the gap between systems biology and process engineering. **Journal of biotechnology**, v. 132, n. 4, p. 418–425, 1 dez. 2007.

THOMPSON, M. L.; KRAMER, M. A. Modeling chemical processes using prior knowledge and neural networks. **AIChE Journal**, v. 40, n. 8, p. 1328–1340, ago. 1994.

WANG, S.; TENG, Y.; PERDIKARIS, P. Understanding and mitigating gradient pathologies in physics-informed neural networks. **arXiv preprint**, 13 jan. 2020.

ZHONG, Y. D.; DEY, B.; CHAKRABORTY, A. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control. **arXiv preprint**, 29 fev. 2024.