



Titre: Développement d'algorithmes d'apprentissage profond adaptés au
génie chimique : cinétique chimique et mélange

Auteur: Valérie Bibeau
Author:

Date: 2023

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Bibeau, V. (2023). Développement d'algorithmes d'apprentissage profond adaptés
au génie chimique : cinétique chimique et mélange [Mémoire de maîtrise,
Citation: Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/53339/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/53339/>
PolyPublie URL:

**Directeurs de
recherche:** Bruno Blais, & Daria Camilla Boffito
Advisors:

Programme: Génie chimique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Développement d'algorithmes d'apprentissage profond adaptés au génie
chimique : cinétique chimique et mélange**

VALÉRIE BIBEAU

Département de génie chimique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie chimique

Août 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Développement d'algorithmes d'apprentissage profond adaptés au génie
chimique : cinétique chimique et mélange**

présenté par **Valérie BIBEAU**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Louis FRADETTE, président

Bruno BLAIS, membre et directeur de recherche

Daria Camilla BOFFITO, membre et codirectrice de recherche

Moncef CHIOUA, membre

DÉDICACE

*À ma famille qui m'a soutenue depuis toujours,
et à mes amis qui m'ont remplie de chimique love.*

REMERCIEMENTS

J'aimerais tout d'abord adresser mes plus sincères remerciements à mon directeur de recherche, Bruno. C'est en 2019 que j'ai décidé de faire un stage en pédagogie dans son équipe afin de remodeler un cours de génie chimique à Polytechnique. En 2020, avec la pandémie, Bruno m'a prise sous son aile une deuxième fois en tant que stagiaire de recherche. Je crois que c'est à ce moment que, tous les deux, nous avons eu la piqure de l'apprentissage profond. Finalement, Bruno a accepté d'être mon directeur de recherche pour superviser ma maîtrise. Depuis maintenant 4 ans, Bruno m'accorde une confiance sans précédent, et je lui en suis extrêmement reconnaissante. Je le considère comme un mentor. J'ai énormément appris sur moi-même, en tant que jeune chercheuse, grâce à lui. C'est ainsi avec grande fierté que je présente deux beaux projets, dont j'ai eu l'immense chance de faire avec lui durant ma maîtrise. Merci mille fois, Bruno.

Je souhaite également remercier ma codirectrice de recherche, Daria. Il était important pour moi d'avoir une professeure forte et passionnée qui superviserait ma maîtrise. Daria était toujours là, présente et prête à répondre à toutes mes questions, petites et stupides soient elles. Sa rigueur scientifique m'a grandement inspirée à moi-même restée minutieuse dans ma méthodologie. Merci d'avoir dit oui, sans aucune hésitation, à faire partie de cette aventure.

Mon expérience en recherche n'aurait pas été la même sans les collègues incroyables que j'ai côtoyés durant ces dernières années : Lucka, Laura, Ghazaleh, Audrey, Toni, Hélène, Olivier, Gabo, Pierre, Paul, Amishga, Victor, Lucas, Jeanne, Shahab. La recherche, ça doit se faire en ayant du plaisir, et ce n'est que de la joie que je ressens lorsque je pense à eux et à elles. En réalité, je ne les considère pas vraiment comme des collègues, mais comme des amis. Ils et elles m'ont vu dans mes hauts et dans mes bas, et je les remercie de tout mon cœur de m'avoir supportée durant toutes ces années.

Un grand merci à Carole-Anne. Ta bravoure m'a servie d'élan pour terminer ma maîtrise. Je me suis beaucoup inspirée de ton mémoire pour écrire certaines sections. Tu étais notre rayon de soleil dans le groupe et la trace que tu as laissée derrière toi en a inspiré plusieurs, dont moi.

Un énorme merci à l'équipe de Daria : Hela, Dalma, Igor, Nicolas. Sans vous, je n'aurais jamais réussi à compléter mes expériences au laboratoire. Hela, Igor et Dalma, ce fameux GC... Il nous a donné beaucoup de fil à retordre, mais en équipe, on l'a remis sur pieds quelques fois. Nicolas, grâce à ta formation impeccable et ta disponibilité exemplaire, faire des expériences avec le réacteur à micro-ondes était une partie de plaisir. Merci à vous tous.

Merci, Jérôme, mon complice depuis le cours de dynamique des systèmes. On s'est vu grandir, de notre baccalauréat à aujourd'hui. Merci de m'avoir accompagnée dans mes moments de réflexion intenses. Je t'en dois une pour ton doctorat.

Merci, Philou, pour ton support inconditionnel. Tu m'as remise plusieurs fois les pieds sur terre, tu m'as souvent répété à quel point j'étais capable de tout accomplir et tu m'as toujours encouragée à rêver plus grand.

Merci à ma famille, papa, maman, Isabelle, Milo, Sara et Audrey, pour leurs immenses encouragements.

J'aimerais finalement remercier l'Institut de valorisation de données et le Conseil de recherches en sciences naturelles et en génie du Canada pour les bourses offertes. Ces dernières m'ont permis de mettre les deux projets à bon terme. Un grand merci au département de génie chimique de Polytechnique Montréal pour leur soutien académique constant.

RÉSUMÉ

Les opérations unitaires nécessitent des méthodes de conception robustes et adéquatement définies pour remplir leur rôle. Par exemple, afin de concevoir les mélangeurs, la puissance d'agitation est un élément clé de conception. Afin de déterminer le nombre de puissance, un nombre adimensionnel caractérisant les rendements énergétiques des agitateurs, des corrélations algébriques existent. Elles sont construites en établissant une régression sur des données obtenues en laboratoire. Un autre exemple est la conception des réacteurs, se trouvant au cœur des procédés chimiques. Afin de déterminer le temps de résidence des réactifs ou de dimensionner le volume d'un réacteur, des équations de conception existent. Ces équations sont dépendantes de la cinétique des réactions. Encore une fois, afin de construire de telles équations, une régression sur des données expérimentales est envisageable. Toutefois, les corrélations du nombre de puissance sont non linéaires en raison de leur dépendance complexe aux propriétés du fluide et à la géométrie de l'agitateur. La cinétique chimique présente également des dépendances non linéaires avec la concentration des espèces et la température. Afin de quantifier ces différentes relations, des outils d'apprentissage profond sont développés afin de contourner certaines limitations des méthodes numériques actuelles. Le projet consiste donc à étudier le potentiel d'utilisation de réseaux de neurones pour répondre à des défis rencontrés en génie chimique. Le projet se divise en deux parties. D'une part, un réseau de neurones est employé afin d'établir une fonction hypothétique entre le nombre de puissance, le nombre de Reynolds et les caractéristiques géométriques des mélangeurs. D'autre part, un réseau de neurones informé de la physique est développé pour prédire la cinétique chimique d'une production de biodiesel dans un réacteur à micro-ondes. Dans les deux cas, le présent mémoire revoit la méthodologie derrière la génération de données, le développement des architectures des réseaux et finalement, la performance des prédictions suite à leur entraînement.

L'article I s'intéresse à la possibilité de remplacer certaines corrélations empiriques du nombre de puissance par une fonction hypothétique déterminée par un réseau de neurones. Avant d'arriver à cet objectif, l'article décrit la façon dont les données sur le nombre de puissance sont générées. Un logiciel libre de CFD, appelé *Lethe*, permet de simuler numériquement, via la méthode des éléments finis, l'écoulement du fluide dans une cuve agitée par un agitateur à pales inclinées. La sensibilité des résultats numériques au maillage est étudiée avant de lancer 100 000 simulations de mélangeurs sur des grappes de calcul. Pour chaque simulation, le couple est extrait et le nombre de puissance est calculé. Ces données numériques servent à alimenter le réseau de neurones. Ensuite, une validation croisée est employée afin de dé-

terminer le nombre de neurones et le nombre de couches de l'architecture du réseau. Après avoir construit l'algorithme avec la librairie TensorFlow, le réseau de neurones est entraîné pendant 5 000 itérations sur 70 % de la base de données. Le réseau de neurones émet des prédictions sur le 30 % restant. L'erreur moyenne sur les prédictions est de 0.5 %, ce qui est une excellente métrique de performance. Les prédictions ont également été comparées avec la corrélation de Hiroaka et al. (1997). Le réseau de neurones est capable de surpasser la corrélation de par le fait qu'il est apte à généraliser ses prédictions à plusieurs géométries différentes.

L'article II s'intéresse, quant à lui, à l'utilisation d'un réseau de neurones informé de la physique (ou PINN) afin d'identifier la cinétique chimique d'une production de biodiesel. Pour ce faire, un réacteur à micro-ondes est utilisé pour provoquer la réaction de transestérification de l'huile de canola en présence d'un catalyseur basique. La réaction est faite à plusieurs temps de réaction (de 1 à 10 minutes) et à des puissances de micro-ondes différentes (de 4 à 6 W). Le biodiesel produit est analysé par chromatographie gazeuse (GC-FID) et la température est mesurée par un capteur infrarouge, présent dans la cavité du réacteur. Ensuite, le PINN est développé en utilisant la librairie PyTorch. Le PINN s'entraîne sur les données de concentration et de température des expériences et également sur la forme résiduelle des équations différentielles ordinaires décrivant les bilans molaires des espèces chimiques. L'entraînement du PINN utilise une régularisation afin de respecter le plus possible la physique du problème à l'étude. Après l'entraînement du PINN, ce dernier prédit des concentrations proches des données expérimentales et respecte les équations différentielles encodées dans l'algorithme. De plus, le PINN peut prédire la cinétique chimique d'une réaction à différentes puissances du réacteur, dont les données de concentration ne font pas partie de l'entraînement. Les résultats obtenus permettent de démontrer le potentiel du PINN à jouer le rôle de jumeau numérique du réacteur.

Somme toute, le projet de maîtrise cherche à démontrer le potentiel de l'apprentissage profond à résoudre des problèmes d'ingénierie complexes. L'objectif est donc de développer des algorithmes d'apprentissage profond, plus spécifiquement un réseau de neurones et un réseau de neurones informé de la physique, adaptés au génie chimique pour déterminer les requis énergétiques d'un mélangeur et identifier la cinétique chimique d'un système.

ABSTRACT

Unit operations require properly defined methods for their design. For example, the agitation power consumption is a key design element in mixing operations. To characterize the energy requirements of industrial agitators, we need to determine the power number. Algebraic correlations exist to correlate this dimensionless number to fluid properties and geometrical characteristics of the agitator. These correlations are constructed by establishing a regression on experimental data. Another example is the design of reactors, which are at the core of chemical processes. To determine the residence time of batch reactors or the volume of continuous reactors, differential equations exist. These equations depend on reaction kinetics. Once again, regression on experimental data is required to construct such equations. However, power number correlations have nonlinear dependencies with the Reynolds number and the agitator geometry. Chemical kinetics also exhibit nonlinear dependencies with species concentration and temperature. To quantify these complex relationships, researchers develop deep learning algorithms to overcome limitations of current numerical methods. Therefore, the project aims to study the use of neural networks to address challenges encountered in chemical engineering. The project is divided into two parts. Firstly, an artificial neural network (ANN) is established to create a hypothetical function between the power number, the Reynolds number, and geometric characteristics of mixers. Secondly, a physics-informed neural network (PINN) is developed to predict the reaction kinetics of biodiesel production in a microwave reactor. In both cases, this thesis reviews the methodology behind data generation, the development of network architectures, and ultimately, the performance of predictions following their training.

The first paper focuses on the possibility of replacing empirical power number correlations with a hypothetical function made by an ANN. To do so, the article describes how power number data are generated. Lethe, an open-source CFD software, numerically simulate the fluid flow around pitched blade impellers of mechanically agitated tanks. We study the sensitivity of the mesh size on the numerical results before running 100,000 mixer simulations on computing clusters. For each simulation, we extract the torque and we calculate the power number. We use these data to feed the neural network. We then use cross-validation to determine the number of neurons and the number of layers in the network architecture. After building the algorithm with the TensorFlow library, we train the ANN for 5,000 iterations on 70 % of the database. The neural network makes predictions on the remaining 30 %. The average error on predictions is 0.5 %, which is an excellent performance metric. The predictions were also compared with the correlation of Hiroaka et al. (1997). The ANN is

able to outperform the correlation because it is able to generalize its predictions to several different geometries.

The second article focuses on the use of a PINN to identify the reaction kinetics of biodiesel production. To achieve this, we use a microwave reactor to perform the transesterification reaction of canola oil in the presence of an alkaline catalyst. We carry out the reaction at various reaction times (from 1 to 10 minutes) and at different microwave powers (from 4 to 6 W). We analyse the biodiesel by gas chromatography with flame ionization detection (GC-FID) and we measure the temperature with an infrared sensor. We then develop the PINN using the PyTorch library. We train the PINN on concentration and temperature data, and also on the residual form of the ordinary differential equations describing the molar balances of chemical species. The PINN's training uses regularization to respect as closely as possible the physics of the problem under study. After training, the PINN predicts concentrations close to the experimental data and respects the differential equations encoded within the algorithm. In addition, PINN can predict the chemical kinetics of a reaction at different power values, for which the concentration data are not part of the training. The results indicate that the PINN has the potential to perform as a digital twin of the reactor.

All in all, both projects from this master's thesis seek to demonstrate the potential of deep learning to solve complex engineering problems. Therefore, the main objective is to develop deep learning algorithms, more specifically an ANN and a PINN, adapted to chemical engineering problems in order to determine the energy requirements of a mixer and identify the chemical kinetics of a system.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xiii
LISTE DES FIGURES	xiv
LISTE DES SIGLES ET ABRÉVIATIONS	xvii
LISTE DES ANNEXES	xix
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 REVUE DE LITTÉRATURE	4
2.1 Méthodes de conception pour les mélangeurs industriels	4
2.1.1 Le mélange	4
2.1.2 Nombre de puissance	8
2.1.3 Résolution numérique du nombre de puissance	9
2.2 Méthodes de conception de réacteurs	12
2.2.1 Le réacteur	12
2.2.2 Équations de conception	13
2.2.3 Résolution numérique des équations de conception	15
2.3 Apprentissage profond	16
2.3.1 Réseau de neurones	16
2.3.2 Réseau de neurones informé par la physique	25
2.4 Synthèse et objectifs de recherche	33
CHAPITRE 3 MÉTHODOLOGIE ET ORGANISATION	35
3.1 Méthodologie associée aux objectifs	35

3.2	Organisation du mémoire	37
3.3	Librairies Python en apprentissage profond	37
3.4	Lethe	40
CHAPITRE 4 ARTICLE I. ARTIFICIAL NEURAL NETWORK TO PREDICT THE POWER NUMBER OF AGITATED TANKS FED BY CFD SIMULATIONS		41
4.1	Abstract	41
4.2	Introduction	42
4.3	Models and methods	44
4.3.1	Database generation	44
4.3.2	Artificial Neural Network	47
4.4	Mesh sensitivity	51
4.5	Results and discussion	54
4.6	Conclusion	57
4.7	Aknowledgements	59
CHAPITRE 5 ARTICLE I. DISCUSSION		61
5.1	Génération des données numériques	61
5.2	Validation croisée	62
5.3	Effet de la taille des données alimentées au réseau	62
CHAPITRE 6 ARTICLE II. PHYSICS-INFORMED NEURAL NETWORK TO PRE- DICT KINETICS OF BIODIESEL PRODUCTION IN MICROWAVE REACTORS		64
6.1	Abstract	64
6.2	Graphical abstract	65
6.3	Introduction	65
6.3.1	Physics-informed Neural Networks	66
6.4	Methodology	68
6.4.1	Concentration and temperature data	69
6.4.2	Physical model	70
6.4.3	PINN implementation	72
6.5	Results and Discussion	74
6.5.1	Regularization	74
6.5.2	Identification of the system	75
6.5.3	Extrapolation to conditions not trained for	78
6.6	Conclusion	79
6.7	Aknowledgements	81

CHAPITRE 7	ARTICLE II. DISCUSSION	82
7.1	Génération des données expérimentales	82
7.2	Neurones d'entrée et de sortie	82
7.3	Régularisation	84
7.4	Respect des bilans molaires	86
CHAPITRE 8	CONCLUSION	90
8.1	Synthèse des travaux	90
8.2	Limitations de la solution proposée	91
8.3	Améliorations futures	92
8.4	Mot de la fin	92
RÉFÉRENCES		94
ANNEXES		102

LISTE DES TABLEAUX

Tableau 2.1	Fonctions d'activation, leurs avantages et désavantages.	20
Tableau 3.1	Section du mémoire associé aux sous-objectifs du premier objectif principal.	37
Tableau 3.2	Section du mémoire associé aux sous-objectifs du deuxième objectif principal.	37
Tableau 3.3	Distinction de la structure de code entre TensorFlow et PyTorch. . .	39
Table 4.1	Input variables and their interval.	47
Table 4.2	Performance metrics on each set.	55
Table 4.3	Correlation equations for the power number for pitched and paddle impellers in unbaffled condition.	56
Table 6.1	Power output of the microwave and reaction time of the experiments.	69
Table 6.2	Time domain and number of collocation points for each microwave power.	73
Tableau A.1	Erreur relative des prédictions des constantes cinétiques avec des données artificielles.	104
Tableau A.2	Erreur relative des prédictions des constantes cinétiques avec des données artificielles bruitées.	104

LISTE DES FIGURES

Figure 2.1	Pyramide de la revue de littérature.	5
Figure 2.2	Mélangeur et ses composantes, adapté de [1].	7
Figure 2.3	Types d'agitateurs. a) Ruban hélicoïdale. b) Hélice. c) Turbine Rushton.	7
Figure 2.4	Nombre de puissance selon le nombre de Reynolds pour quelques types d'agitateur, tiré de [2].	8
Figure 2.5	Maillage d'un domaine 2D, adapté de [3].	11
Figure 2.6	Types de réacteur, adapté de [4]. a) Discontinu. b) Continu.	13
Figure 2.7	Réacteur à écoulement tubulaire (PFR), tiré de [4].	14
Figure 2.8	La place de l'apprentissage profond en intelligence artificielle.	17
Figure 2.9	Réseau de neurones avec $\mathbf{x}_1$ et $\mathbf{x}_2$ les entrées, $\mathbf{y}$ la sortie et σ les neurones actifs.	18
Figure 2.10	Fonctions d'activation.	20
Figure 2.11	Descente de gradient.	22
Figure 2.12	Effet du <i>batch size</i> sur la descente de gradient.	23
Figure 2.13	Exemple d'une rétropropagation pour le calcul de $\partial J / \partial w^{[i]}$	24
Figure 2.14	Points de collocation ($\bullet$), conditions initiales ($\circ$) pour une EDO.	26
Figure 2.15	Points de collocation ($\bullet$), conditions initiales ($\circ$) et conditions frontières (Δ) pour une EDP.	27
Figure 2.16	Réseau de neurones informé par la physique (PINN).	29
Figure 2.17	Points de collocation ($\bullet$), conditions initiales et données connues ($\circ$) pour une EDO.	30
Figure 2.18	Points de collocation ($\bullet$), conditions initiales et données connues ($\circ$), conditions frontières (Δ) pour une EDP.	31
Figure 2.19	Visualisation de la fonction de perte, adapté de [5]. a) Fonction parfaitement convexe. b) Fonction de référence. c) Fonction chaotique.	31
Figure 2.20	Diagramme sommaire de la revue de littérature.	34
Figure 2.21	L'utilisation des données versus l'emploi de modèles physiques pour les deux objectifs, inspiré de [6].	34
Figure 4.1	Scheme of the mixing rig used for the simulations.	45
Figure 4.2	Artificial Neural Network (ANN) architecture with two hidden layers to predict the power number based on features that describes the mixer.	48
Figure 4.3	Velocity in the radial direction in a mixer at $Re = 100$ in a top view perspective.	51

Figure 4.4	Velocity in the axial direction at $Re = 100$ in a side view perspective.	52
Figure 4.5	Mesh refinement near the blades and the shaft of the impeller.	52
Figure 4.6	Normalized torque Γ according to the maximum characteristic length of the mesh near the agitator for average and worse geometries. . . .	54
Figure 4.7	Predictions using the ANN and the correlation from Hiraoka et al. (1997) versus values of the power number N_p from the CFD simulations.	56
Figure 4.8	Power number versus Reynolds number for a specified geometry ($T/D = 3$, $H/T = 1$, $T/C = 4$, $D/W = 5$, $E/W = 0.1$, $\theta = \pi/4$) using the ANN, CFD simulations (Lethe) and Hiraoka et al. (1997) correlation.	58
Figure 4.9	Power number versus Reynolds number for a specified geometry ($T/D = 2.5$, $H/T = 1.2$, $T/C = 3$, $D/W = 3.5$, $E/W = 0.15$, $\theta = \pi/6$) using the ANN, CFD simulations (Lethe) of two different element sizes L_c and Hiraoka et al. (1997) correlation.	58
Figure 4.10	Validation of the ANN's prediction of a power curve for a specified geometry ($T/D = 3$, $H/T = 1$, $T/C = 4$, $D/W = 5$, $E/W = 0.15$, $\theta = \pi/4$).	59
Figure 5.1	Analyse du lot sur la base de données du nombre de puissance.	63
Figure 6.1	Graphical abstract of the PINN article.	65
Figure 6.2	PINN's architecture.	68
Figure 6.3	Reaction kinetics architecture.	73
Figure 6.4	Impact of the regularization η_r on the predictions of the concentration of triglycerides and on the evaluation of the rate constants.	76
Figure 6.5	Temperature predicted by the PINN.	77
Figure 6.6	Parity plot between the concentration of glycerides C predicted by the PINN and the experimental results.	77
Figure 6.7	PINN's predictions of the concentration of diglycerides over time compared to the numerical method and the experiments.	78
Figure 6.8	PINN's predictions of the concentration of diglycerides over time with only data from 4 W are known during training.	79
Figure 6.9	PINN's predictions of the concentration of diglycerides over time with only data from 6 W are known during training.	80
Figure 7.1	Évolution du mélange. a) Formation du vortex. b) Gouttelettes. c) Mélange homogène.	83
Figure 7.2	Emplacements possibles du neurone de température dans l'architecture.	84
Figure 7.3	Fonction de perte des données de concentration (en bleu) et des EDOs (en rouge).	85

Figure 7.4	Évolution de la variation de température selon le temps de la réaction de transestérification et de l'huile de canola (sans réaction).	86
Figure 7.5	Évolution de la température selon le temps prédite par le PINN, résolue par Runge-Kutta et décrite par les données expérimentales du réacteur.	87
Figure 7.6	Concentration de G et ME selon le temps prédite par le PINN et résolue par Runge-Kutta pour une régularisation $\eta_r = 5$	88
Figure 7.7	Concentration de G et ME selon le temps prédite par le PINN et résolue par Runge-Kutta pour une régularisation $\eta_r = 5000$	88
Figure A.1	Prédictions de la concentration des espèces chimiques par le PINN (—) comparées aux données artificielles (- - -) et aux données d'entraînement (●).	103
Figure A.2	Prédictions de la concentration des espèces chimiques par le PINN (—) comparées aux données artificielles (- - -) et aux données d'entraînement bruitées (●).	105

LISTE DES SIGLES ET ABRÉVIATIONS

Mémoire

NTU	<i>Number of transfer units</i>
IA	Intelligence artificielle
CFD	<i>Computational fluid dynamics</i>
EDP	Équations différentielles partielles
MEF	Méthode des éléments finis
CSTR	<i>Continuous stirred tank reactor</i>
PFR	<i>Plug flow reactor</i>
PBR	<i>Packed bed reactor</i>
EDO	Équation différentielle ordinaire
ReLU	<i>Rectified linear unit</i>
MSE	<i>Mean squared error</i>
MAE	<i>Mean absolute error</i>
MAPE	<i>Mean absolute percentage error</i>
CNN	<i>Convolutional neural network</i>
RNN	<i>Recurrent neural network</i>
PINN	<i>Physics-informed neural network</i>
RK	Runge-Kutta
URPEI	Unité de recherche en procédés d'écoulement industriels

Article I

ANN	<i>Artificial neural network</i>
LHS	<i>Latin hypercube sampling</i>
PBT	<i>Pitched blade turbine</i>
MPC	<i>Model predictive control</i>
SUPG	<i>Streamline upwind Petrov-Galerkin</i>
PSPG	<i>Pressure-stabilized Petrov-Galerkin</i>
SRF	<i>Single rotating frame</i>
CPU	<i>Central processing unit</i>
PCA	<i>Principal component analysis</i>
CV	<i>Cross validation</i>
IVADO	Institut de valorisation des données

Article II

PI	<i>Process intensification</i>
GC	<i>Gas chromatography</i>
GC-FID	<i>Gas chromatography with flame ionization detection</i>
ODE	<i>Ordinary differential equation</i>
PDE	<i>Partial differential equation</i>
IR	<i>Infrared</i>
TG	Triglycéride
DG	Diglycéride
MG	Monoglycéride
G	Glycérol
MeOH	Méthanol
NaOH	Hydroxide de sodium
ME	<i>Methyl ester</i>
GPU	<i>Graphics processing unit</i>
NSERC	<i>Natural Sciences and Engineering Research Council of Canada</i>

LISTE DES ANNEXES

Annexe A	Vérification de code avec des données artificielles	102
----------	---	-----

CHAPITRE 1 INTRODUCTION

Les techniques de conceptions d'aujourd'hui

Les opérations unitaires en génie sont multiples : mélangeurs, pompes, colonnes à distiller, échangeurs de chaleur, fournaies, réacteurs et plus encore. De leur conception à leur mise en marche, leur contrôle et leur maintenance, le génie chimique est mis de l'avant. Pour concevoir de telles unités, l'ingénieur chimiste se réfère essentiellement à des méthodes techniques, des heuristiques, des corrélations et des diagrammes.

Pour analyser et concevoir une colonne à distiller, la méthode traditionnelle de McCabe-Thiele est employée. Celle-ci consiste à établir un diagramme basé sur l'équilibre liquide-vapeur de deux composantes qui permet, entre autres, de déterminer le nombre de plateaux de la colonne [7]. Un autre exemple est la conception d'un agitateur au sein d'un mélangeur industriel. Des courbes de puissance sont construites pour différents types d'agitateurs et permettent à l'ingénieur.e d'approximer la puissance consommée par le mélange [1]. Pour ce qui est de la conception des échangeurs de chaleur, la méthode NTU est l'une des techniques les plus employées. Une efficacité ε , obtenue via des relations algébriques et des diagrammes, est prise en compte pour l'analyse thermique de l'échangeur [8].

Les principales opérations unitaires intégrant la mécanique des fluides, le transfert thermique et les procédés de séparation émettent de fortes hypothèses qui simplifient les systèmes, rendant les méthodes de conception plus faciles à employer. En d'autres termes, ces méthodes suivent des « règles du pouce » qui permettent d'approximer la théorie afin de réduire le temps investi dans la conception d'équipements [9]. Or, les procédés chimiques regorgent de phénomènes d'échange complexes, et leur complexité ne devrait pas être négligée. Les méthodes traditionnelles présentées précédemment manquent généralement d'universalité, empêchant ainsi la conception adéquate de tels systèmes.

Ce problème est encore plus vrai avec l'effervescence de l'intensification des procédés. Ce nouveau paradigme consiste à réinventer les opérations unitaires de sorte à rendre les usines plus petites, plus sélectives et plus efficaces d'un point de vue énergétique [10]. Cette nouvelle philosophie cherche à combiner plusieurs unités opérationnelles ensemble, comme la distillation réactive où l'on combine les tâches de séparation et de réaction. Elle cherche également à utiliser de nouveaux vecteurs d'énergie compatibles avec les énergies renouvelables, comme les micro-ondes. L'intégration de l'intensification des procédés, accompagnée de ses technologies innovantes au génie chimique, rend l'utilisation des méthodes de design conventionnelles contestable.

L'intelligence artificielle, un concept plus vieux qu'on ne le pense

L'arrivée de l'intelligence artificielle (IA) a grandement influencé la manière dont les unités opérationnelles sont conçues. La première intelligence artificielle en génie chimique, apparue autour des années 1980, était sous forme d'un système expert. Un système expert est un programme permettant de résoudre des problèmes complexes d'ingénierie en rassemblant tous les modèles physiques, chimiques et thermodynamiques dans une base de connaissance interne au programme. Les systèmes experts utilisent également le calcul symbolique et des algorithmes d'optimisation pour adresser ces problèmes. Par contre, ces systèmes ont eu de la difficulté à prendre leur envol. Étant donné qu'ils étaient basés sur des heuristiques, certaines applications pratiques et industrielles sont difficilement reproductibles [11].

Les limitations des systèmes experts en génie chimique ont fait en sorte que la recherche en intelligence artificielle a pris une expansion fulgurante. En effet, dans les années 1990, et encore aujourd'hui, plusieurs avancées scientifiques ont été accomplies en apprentissage profond [11]. La clé du succès des réseaux de neurones réside dans sa capacité à créer des fonctions universelles non linéaires à partir d'un lot de données provenant de la réalité, ce que les systèmes experts n'arrivent pas à faire. Toutefois, malgré la recherche vibrante qui se fait en IA, la concrétisation de ses algorithmes en industrie est presque inexistante.

L'éthique et les limitations de l'IA

L'IA semble à première vue une solution alléchante, mais un bémol doit être apporté. L'IA regorge de limitations, qu'elles soient technologiques ou même éthiques. D'une part, les réseaux de neurones ne peuvent garantir de trouver la meilleure solution compte tenu des contraintes imposées par le problème d'optimisation. De plus, ils requièrent une quantité phénoménale de données afin de prédire des solutions adéquates, ce qui peut demander des coûts de calcul élevés [12]. En laboratoire, les données ne sont pas toujours facilement accessibles. D'autre part, les réseaux de neurones tentent d'imiter l'intelligence humaine en suivant un algorithme d'entraînement. Ils sont en mesure d'émettre des prédictions, basées sur une fonction dite hypothétique qui ne peut être représentée par aucun modèle mathématique ni physique. Ainsi, la façon dont ils arrivent à émettre des prédictions demeure floue, voire incompréhensible. Cela peut apporter de grands problèmes éthiques, à savoir si l'humain peut faire confiance à l'IA [13]. En règle générale, dans tout ce qui demande l'application d'une méthode scientifique, le raisonnement logique ou même la suscitation d'une émotion, l'IA éprouve des difficultés. Bref, même si la recherche tente de contourner ses limitations en proposant des algorithmes toujours plus puissants, l'utilisation de l'IA ne doit pas être prise à la légère.

L'arrivée de l'industrie 4.0

L'industrie 4.0 est une révolution qui influence la façon dont les ingénieur.es conçoivent, innovent, maintiennent et contrôlent les technologies d'aujourd'hui. L'industrie 4.0 renvoie à des champs d'expertise tels que la modélisation numérique, les simulations numériques, les données massives, l'internet des objets, les jumeaux numériques et l'intelligence artificielle [14, 15]. Cette révolution industrielle permet non seulement d'optimiser les industries actuelles, mais également de les transformer. En effet, par de l'apprentissage automatique sur des données numériques et expérimentales d'une opération unitaire, des outils d'intelligence artificielle peuvent améliorer sa conception, faciliter sa mise à l'échelle, optimiser son rendement, et même prédire certains pronostics de maintenance tout au long de sa durée de vie.

Nous avons la clé, soit les outils d'apprentissage profond, et nous avons une grande porte à ouvrir, celle de l'industrie 4.0, qui nous mènera vers une évolution qui améliorera la façon dont on conçoit les procédés d'aujourd'hui. Il ne manque plus qu'à tourner la clé dans la serrure, et ce mémoire tentera de le faire avec des applications concrètes du génie chimique.

Ce mémoire se concentrera davantage sur l'utilisation de l'apprentissage profond (ou plus couramment appelé *deep learning*), permettant la conception, l'analyse et l'optimisation d'une opération unitaire classique, soit le mélangeur, et d'une opération unitaire intensifiée, soit le réacteur à micro-ondes.

Le mémoire sera présentée comme suit :

- Au chapitre 2, le mémoire présente une revue de la littérature brève sur les méthodes de conception d'opérations unitaires ainsi que sur les méthodes de résolution numérique, suivie d'une revue complète sur l'apprentissage profond et les réseaux de neurones.
- Au chapitre 3, la méthodologie menant vers l'atteinte des objectifs sera détaillée ainsi que l'organisation du mémoire sera présentée.
- Au chapitre 4, le premier article scientifique intitulé « Artificial neural network to predict the power number of agitated tanks fed by CFD simulations » sera présenté.
- Au chapitre 5, le premier article sera discuté.
- Au chapitre 6, le deuxième article scientifique intitulé « Physics-informed Neural Network to Predict Kinetics of Biodiesel Production in Microwave Reactors » sera présenté.
- Au chapitre 7, le deuxième article sera discuté.
- Au chapitre 8, une conclusion fera une synthèse du projet de maîtrise, expliquera les limitations de celui-ci et proposera des travaux futurs permettant son approfondissement.

CHAPITRE 2 REVUE DE LITTÉRATURE

La revue de littérature de ce présent mémoire est construite de manière à revoir les fondements de conception du génie chimique sur lesquels les articles I et II se concentrent. Les outils d'apprentissage profond développés dans cette étude ont pour but de proposer une nouvelle alternative aux méthodes de conception d'opérations unitaires. La revue commence donc par revoir les méthodes de conception classiques des mélangeurs industriels, d'une part, et des réacteurs d'autre part. Cela permet de pointer précisément sur quels paramètres clés de conception les articles traitent. Au cours de la revue sur les méthodes de conception, plusieurs équations feront surface. En effet, ces méthodes se basent sur des modèles mathématiques, où les lois physiques et les réactions chimiques sont mises en équation. Les équations en découlant peuvent être sous forme de corrélations algébriques ou d'équations différentielles. De par leur complexité non linéaire, les solutions analytiques de ces dites équations sont souvent difficiles à déterminer. Des méthodes numériques permettent de mitiger ce problème et une revue sur celles-ci est faite de manière brève. Finalement, l'apprentissage profond s'introduit comme une option d'amélioration à ces méthodes numériques, et ces algorithmes seront revus de manière détaillée.

La Figure 2.1 permet d'illustrer la division de la revue de la littérature, allant des méthodes de conception à leur mise en équation, leur résolution numérique et leur potentiel d'application en apprentissage profond. Étant donné que l'étude porte spécifiquement sur les mélangeurs et les réacteurs, la revue revisite ces deux opérations unitaires. Somme toute, la revue de littérature permet de converger vers l'essentiel des deux articles qui se trouvent au cœur du mémoire.

2.1 Méthodes de conception pour les mélangeurs industriels

L'objectif de cette section est de définir l'utilité du mélange en génie chimique et de revoir les concepts fondamentaux permettant la conception de mélangeurs industriels. Cette discussion générale se base essentiellement sur la référence [1] (sauf indication contraire).

2.1.1 Le mélange

Le mélange se trouve dans plusieurs industries telles que l'agroalimentaire, la pétrochimie, les bioprocédés, les polymères, le traitement des eaux, et plus encore. Le but d'un mélangeur est d'augmenter l'homogénéité d'un ou de plusieurs fluides. Le mélange permet ainsi de

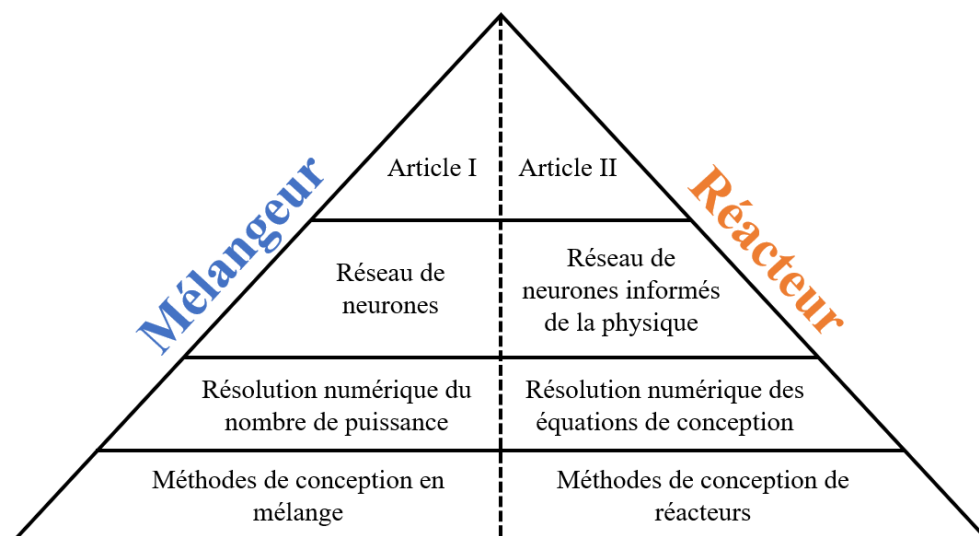


FIGURE 2.1 Pyramide de la revue de littérature.

répartir uniformément la concentration d'une espèce chimique et la température au sein du mélange. Par exemple, le mélange augmente les transferts de masse et de chaleur qui sont nécessaires à la provocation d'une réaction. Ainsi, une conception complète et une mise à l'échelle rigoureuse sont primordiales à la minimisation des coûts d'opération ainsi qu'à l'atteinte d'un rendement profitable.

Les cuves munies d'un agitateur sont les plus couramment utilisées en industrie. L'agitation mécanique consiste essentiellement en un mouvement physique de la matière au sein d'une cuve, généré par la rotation des pales d'un agitateur. Afin de concevoir de tels mélangeurs, quelques étapes clés doivent être suivies :

- Définir adéquatement les exigences de mélange, les propriétés physiques du ou des fluides et le régime d'écoulement ;
- Dimensionner la cuve et son agitateur ;
- Déterminer la puissance d'agitation requise via le nombre de puissance.

Régime d'écoulement

La définition des conditions de mélange du fluide permet d'établir le régime d'écoulement. Le régime est dicté par le nombre de Reynolds, qui relie les forces inertielles du fluide à ses forces visqueuses. Ainsi, le nombre de Reynolds dépend de la densité du fluide ρ , de la vitesse de rotation N de l'agitateur (en tour par seconde), du diamètre de l'agitateur D et de la viscosité μ .

$$\text{Re} = \frac{\rho N D^2}{\mu} \quad (2.1)$$

Lorsque le fluide est visqueux et que le nombre de Reynolds est en deçà d'un seuil de l'ordre de 10, le régime est laminaire. Avec l'augmentation de la vitesse d'écoulement causée par l'agitation, les forces inertielles du fluide prennent le dessus. Le mouvement du fluide devient alors plus chaotique et instable, ce qui forme des tourbillons au sein du mélange. Ainsi, lorsque le nombre de Reynolds atteint des valeurs de l'ordre de 10^4 , le régime devient alors turbulent.

Dimensionnement

La Figure 2.2 illustre les différentes composantes du mélangeur. La cuve, généralement cylindrique, est composée d'un agitateur, d'un moteur, relié à l'agitateur par un arbre, et de chicanes. Les chicanes servent principalement à stabiliser l'apparition de vortex. Lorsque les forces inertielles deviennent importantes par rapport à la force gravitationnelle, un vortex se forme et diminue l'efficacité de mélange.

Le choix de l'agitateur dépend du régime d'écoulement. Les ancrs et les rubans hélicoïdaux permettent de pomper les fluides visqueux (à bas Reynolds) alors que les turbines mélangent les fluides dont l'écoulement est en régime transitoire ou turbulent. Le type d'agitateur est également choisi en fonction des exigences de mélange. Les turbines Rushton font circuler le fluide dans la direction radiale de la cuve, ce qui permet de faciliter la dispersion d'un gaz dans le mélange. Les hélices, quant à elles, pompent le fluide dans la direction axiale, rendant la suspension de particules solides plus efficace. La Figure 2.3 illustre trois types d'agitateurs (parmi tant d'autres).

Puissance d'agitation

La qualité d'un mélange se base sur sa capacité à répartir uniformément l'énergie cinétique transmise au fluide par l'agitateur [16]. Le nombre de puissance N_P permet de caractériser l'efficacité énergétique des mélangeurs en reliant la puissance fournie par l'agitateur P à sa vitesse de rotation N et son diamètre D .

$$P = N_P \rho N^3 D^5 \quad (2.2)$$

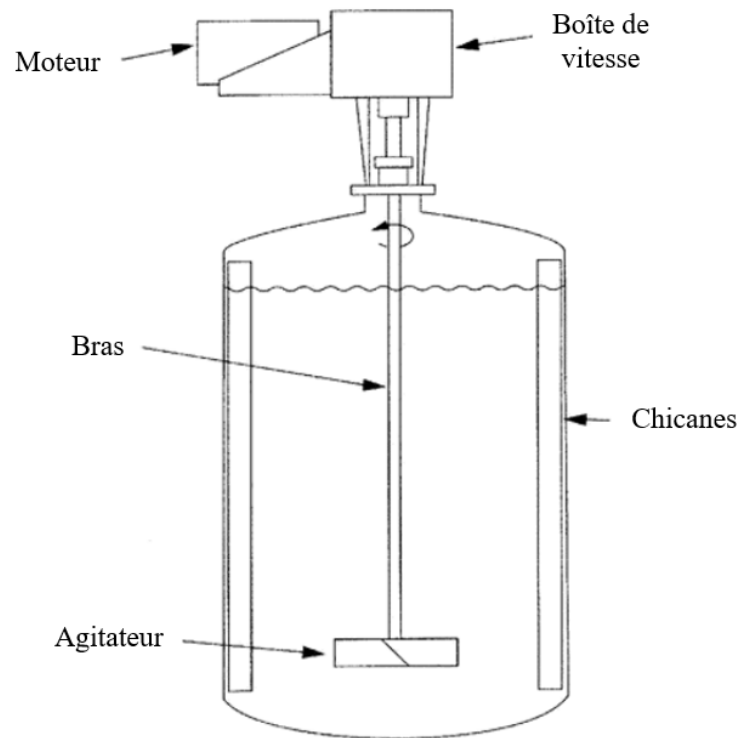


FIGURE 2.2 Mélangeur et ses composantes, adapté de [1].



a)



b)



c)

FIGURE 2.3 Types d'agitateurs. a) Ruban hélicoïdale. b) Hélice. c) Turbine Rushton.

Dans le cas des cuves agitées mécaniquement, la qualité de mélange dépend du type d'agitateur, de ses dimensions et du régime d'écoulement [16]. Le nombre de puissance est donc fonction du nombre de Reynolds, du nombre de pales, de leur largeur, de leur angle d'inclinaison, du diamètre de l'agitateur et de l'élévation de l'agitateur par rapport au fond de la cuve. La Figure 2.4 montre la variation du nombre de puissance en fonction du nombre de Reynolds pour différents agitateurs. On note que, sur une échelle logarithmique, le nombre de puissance décroît linéairement à bas Reynolds et devient constant à plus haut Reynolds. La transition du nombre de puissance entre le régime laminaire et le régime transitoire est dépendante de la géométrie de l'agitateur.

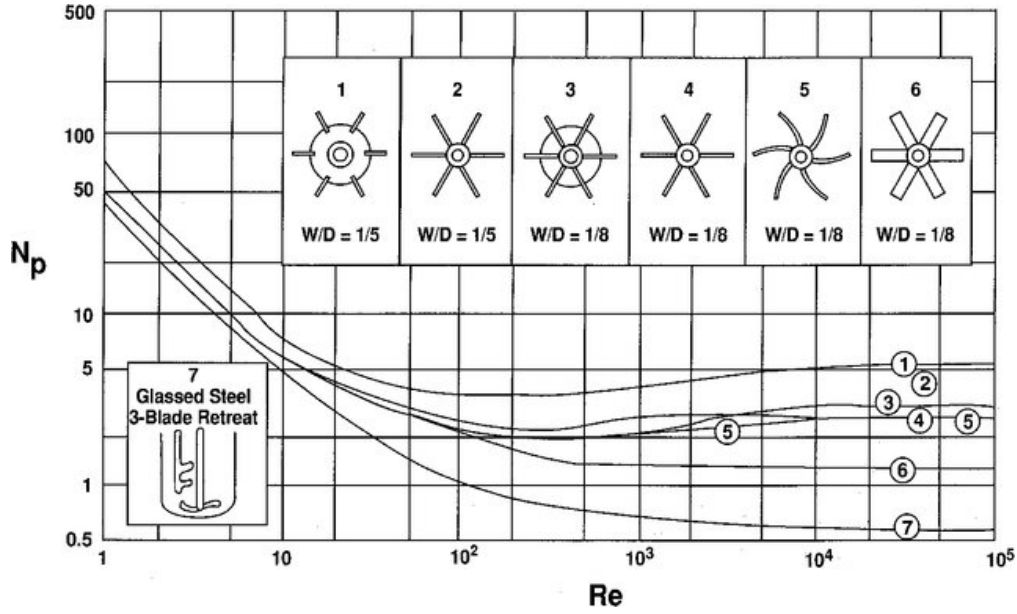


FIGURE 2.4 Nombre de puissance selon le nombre de Reynolds pour quelques types d'agitateur, tiré de [2].

2.1.2 Nombre de puissance

Afin de quantifier la relation entre les conditions physiques du fluide et les caractéristiques géométriques de l'agitateur au nombre de puissance, des corrélations non linéaires, sous forme d'équations polynomiales (ou algébriques), existent. Selon le théorème de Buckingham, une corrélation du nombre de puissance prend la forme suivante :

$$N_P = k (\Pi_1)^a (\Pi_2)^b (\Pi_3)^c \dots \quad (2.3)$$

Les variables Π sont des nombres adimensionnels. Une corrélation du nombre de puissance est généralement fonction du nombre de Reynolds, du nombre de Froude et des ratios entre les dimensions géométriques de l'agitateur [17]. Il existe deux façons principales pour déterminer le nombre de puissance d'un agitateur, soit par expériences en laboratoire ou soit par des simulations de dynamique des fluides numérique (*computational fluid dynamics*, CFD). Les deux approches tentent de calculer le nombre de puissance en mesurant le couple T_q sur l'agitateur.

$$P = 2\pi NT_q \quad (2.4)$$

Dynamique des fluides numérique

La CFD consiste à simuler la dynamique des écoulements en résolvant numériquement les équations de la mécanique des fluides. La CFD est donc facilement applicable au mélange. La CFD résout la vitesse et la pression d'un fluide en mouvement par l'entremise des équations de Navier-Stokes dans un domaine discrétisé, appelé maillage. L'équation de continuité satisfait la conservation de la masse :

$$\nabla \cdot \mathbf{u} = 0 \quad (2.5)$$

avec $\mathbf{u}$ le vecteur vitesse et ∇ l'opérateur différentiel del, alors que la conservation de la quantité de mouvement est décrite par l'équation :

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (2.6)$$

Il est donc possible de retrouver le couple T_q en intégrant le tenseur des contraintes sur la surface de l'agitateur. Ensuite, le nombre de puissance peut être directement calculé.

2.1.3 Résolution numérique du nombre de puissance

Comme vu précédemment, les rendements énergétiques d'un mélangeur sont obtenus via le nombre de puissance. Sa corrélation avec le nombre de Reynolds et la géométrie de l'agitateur est non linéaire, rendant sa mise en équation plus difficile qu'il en a l'air. Le nombre de

puissance peut également être déterminé en résolvant numériquement les équations de Navier-Stokes. Outre pour des géométries très simples, aucune solution analytique de ces équations n'existe à ce jour. Afin de contrer ces limitations, des méthodes numériques permettent de résoudre, à une tolérance choisie par l'utilisateur, de telles équations.

Résolution des corrélations

Les corrélations du nombre de puissance, construites à partir du théorème de Buckingham, sont non linéaires du fait que les nombres adimensionnels Π sont élevés à une certaine puissance. Pour trouver les constantes de ces corrélations, des données sur les mélangeurs doivent être recueillies et une régression non linéaire peut être employée. Les méthodes de régression non linéaire cherchent généralement à résoudre un problème d'optimisation au sens des moindres carrés [18]. On cherche alors à minimiser la somme des erreurs quadratiques Φ entre les valeurs expérimentales $\hat{\mathbf{y}}$ et les valeurs calculées par la corrélation $\mathbf{y} = f(\mathbf{x}; \mathbf{b})$, où $\mathbf{x}$ englobe les nombres adimensionnels de la corrélation et $\mathbf{b}$ ses constantes (ou les exposants de la dite corrélation).

$$\Phi = (\hat{\mathbf{y}} - \mathbf{y})^T (\hat{\mathbf{y}} - \mathbf{y}) \quad (2.7)$$

Des méthodes numériques existent afin de résoudre l'équation non linéaire de manière itérative. Le principe est d'appliquer une correction $\Delta \mathbf{b}$ sur les constantes, jusqu'à obtenir Φ proche d'une valeur nulle. La correction provient du développement en série de Taylor de Φ [19]. La correction est alors déterminée de la façon suivante :

$$\Delta \mathbf{b} = 2H^{-1}J^T(\hat{\mathbf{y}} - \mathbf{y}) \quad (2.8)$$

avec J la matrice jacobienne et H la matrice hessienne, représentant respectivement les dérivées premières et secondes de Φ par rapport à $\mathbf{b}$. Cette méthode est appelée la méthode de Newton [18]. Ensuite, la correction met à jour les valeurs des constantes :

$$\mathbf{b} := \mathbf{b} + \Delta \mathbf{b} \quad (2.9)$$

Il est possible de calculer les matrices jacobienne et hessienne numériquement à l'aide de

différences finies [20].

Méthodes numériques pour la CFD

Comme mentionné plus tôt, la CFD prédit les solutions de vitesse et de pression d'un système de mélange en résolvant numériquement les équations de Navier-Stokes. Ces dernières sont des équations différentielles partielles (EDPs), c'est-à-dire qu'elles ont au moins deux variables indépendantes. Elles sont résolues selon le temps t , si le régime transitoire de l'écoulement du fluide est à l'étude, et l'espace $\mathbf{x} \in \Omega$. Le domaine du système Ω fait partie de $\mathbb{R}^d$, où d est la dimension du système. Les EDPs sont accompagnées de conditions frontières afin de fermer le problème à résoudre. Celles-ci sont résolues sur le contour du domaine $\Gamma \in \mathbb{R}^{d-1}$, là où on distingue le système de son environnement. Afin de résoudre les EDPs de façon numérique, il faut d'abord discrétiser le domaine de calcul Ω en plusieurs points de collocation. La Figure 2.5 présente une discrétisation d'un domaine 2D. L'interconnexion entre les points de collocation forme plusieurs quadrilatères qui, ensemble, représente le maillage du système.

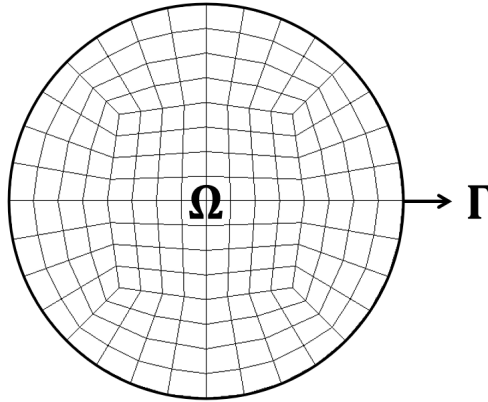


FIGURE 2.5 Maillage d'un domaine 2D, adapté de [3].

À partir de ce maillage, quelques méthodes numériques existent afin de résoudre les EDP. Dans le cadre de ce projet, la méthode des éléments finis (MEF) est employée afin de simuler l'écoulement du fluide autour de l'agitateur. En bref, celle-ci consiste à approximer les solutions en vitesse et en pression par des polynômes et d'intégrer numériquement, via des quadratures, la formulation variationnelle (ou la formulation faible) des EDPs sur chaque élément du maillage [21]. L'article I revoit plus en profondeur cette méthode.

2.2 Méthodes de conception de réacteurs

L'objectif de cette section est de revoir les concepts fondamentaux permettant la conception de réacteurs. La discussion se porte essentiellement sur la référence [4] (sauf indication contraire).

2.2.1 Le réacteur

Il va de soi que le génie chimique perd de son sens sans l'opération unitaire centrale qui transforme la matière brute en produits finis : le réacteur. La conception de réacteurs permet de spécifier les différentes conditions de température, de pression et de concentration permettant une conversion efficace des réactifs, une sélectivité optimale des produits désirés et ultimement au succès économique d'un procédé. Pour concevoir de tels réacteurs, plusieurs éléments clés doivent être considérés.

- Établir les réactions en vigueur et leur bilan molaire ;
- Choisir le type de réacteur ;
- Établir l'équation de conception qui dimensionne le réacteur (i.e, son volume) ou qui détermine les conditions d'opération du réacteur (i.e, son temps de résidence) ;
- Déterminer la cinétique ;
- Résoudre les équations de conception avec des données expérimentales.

Bilan molaire

L'étude des réacteurs nécessite un bilan molaire. Le principe de Lavoisier disant : « rien ne se perd, rien ne se crée, tout se transforme » est toujours omniprésent dans la façon dont les procédés sont conçus. En génie chimique, ce principe se traduit par un bilan molaire, statuant le fait que le taux d'accumulation dans un système est dépendant de ce qui entre dans un réacteur, ce qui en sort et ce qui est généré par une réaction chimique.

$$\frac{d(C_j)}{dt} = \text{Entrée} - \text{Sortie} + \text{Génération} \quad (2.10)$$

Ici, C_j est la concentration d'une espèce chimique (en mol L^{-1}). La génération d'une espèce dépend du taux auquel cette dernière est produite, représenté par r_j . Par convention, si une espèce est consommée, son taux de génération est négatif ($-r_j$).

2.2.2 Équations de conception

La Figure 2.6 illustre les deux types de réacteurs : discontinu (ou plus communément appelé *batch*) et continu.

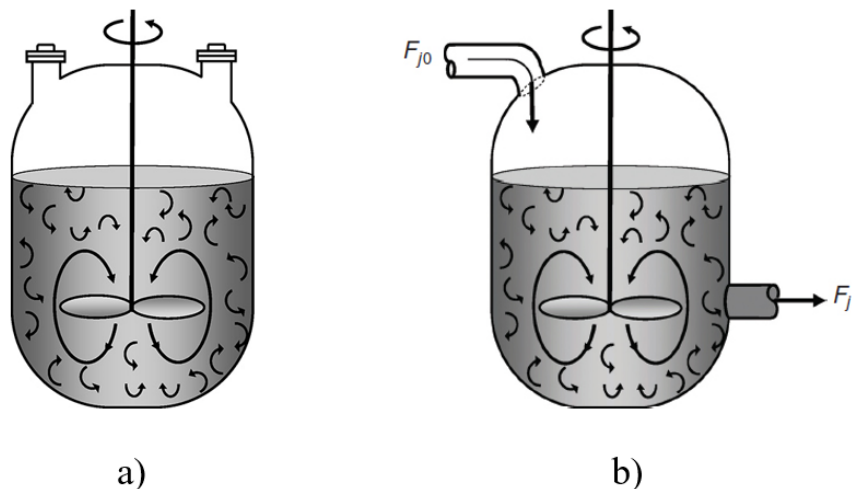


FIGURE 2.6 Types de réacteur, adapté de [4]. a) Discontinu. b) Continu.

Les réacteurs *batch* sont plus utilisés pour des procédés à petite échelle. L'avantage des procédés *batch* est qu'ils peuvent atteindre de haut rendement puisque les réactifs sont toujours présents dans le réacteur. Ils ne possèdent aucune entrée ni sortie de matière, faisant en sorte que le bilan molaire se simplifie :

$$\frac{dC_j}{dt} = r_j \quad (2.11)$$

Cette équation différentielle est appelée l'équation de conception d'un réacteur discontinu. Il est possible d'intégrer cette équation afin d'obtenir le temps de résidence t des réactifs pour qu'ils se convertissent d'une concentration initiale C_0 à une concentration finale C .

Les réacteurs continus, incluant les réacteurs continus en agitation (*continuous stirred tank reactor*, CSTR), les réacteurs à écoulement tubulaire (*plug flow reactor*, PFR) et les réacteurs à lit fixe (*packed bed reactor*, PBR), sont utilisés en régime permanent. Pour un CSTR, l'accumulation de matière est nulle et on assume que les espèces sont parfaitement mélangées, l'équation de conception devient :

$$V = \frac{v_0 C_j - v C_{j0}}{-r_j} \quad (2.12)$$

Ici, V est le volume du CSTR, v_0 et v les débits volumiques entrant et sortant du réacteur respectivement.

La Figure 2.7 présente l'écoulement d'un fluide dans un réacteur PFR ou PBR (à l'exception que ce dernier contient un lit catalytique). Étant donné que la réaction évolue au fur et à mesure que les espèces avancent dans le réacteur, la concentration devient dépendante de sa position. Avec un bilan molaire, cette fois-ci appliqué sur une section de volume ΔV , l'équation de conception d'un PFR est :

$$\frac{dC_j}{dV} = r_j \quad (2.13)$$

En intégrant cette équation, le volume du réacteur peut être calculé.

Les équations de conception des différents types de réacteurs ont été présentées. Dans le cas d'un procédé discontinu, le temps de résidence est la variable de conception alors que pour un procédé continu, le volume est calculé. Dans tous les cas, les équations de conception dépendent du taux de génération r_j . Ce dernier dépend de lois cinétiques.

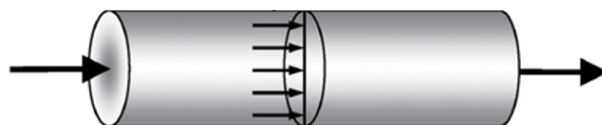


FIGURE 2.7 Réacteur à écoulement tubulaire (PFR), tiré de [4].

Cinétique

Considérons une réaction de la forme :



avec a , b , c et d les coefficients stoechiométriques des espèces A, B, C et D respectivement. La

loi cinétique stipule que le taux de génération d'une espèce dépend d'une constante cinétique k et du produit des concentrations des réactifs, chacune élevée à une puissance.

$$-r_A = k_A (C_A)^\alpha (C_B)^\beta \quad (2.15)$$

Ici, $-r_A$ est le taux de consommation de l'espèce A , C_A et C_B sont les concentrations des réactifs, α et β sont les exposants de la loi cinétique. L'ordre de la réaction est donné par $n = \alpha + \beta$. Lorsque la somme des coefficients stœchiométriques des réactifs correspondent exactement à l'ordre de la réaction, on dit que cette dernière suit une loi cinétique élémentaire. Ainsi, α équivaut au coefficient a et β au coefficient b de la réaction chimique. Il est à noter que la loi cinétique d'une réaction n'est pas toujours élémentaire. Elle dépend de la réaction et de son mécanisme, et non du réacteur employé. Il est donc important de confirmer la loi cinétique employée avec des données expérimentales.

2.2.3 Résolution numérique des équations de conception

Ayant en main les équations de conception, basées sur un bilan molaire, et les lois cinétiques, il reste à déterminer les constantes cinétiques et l'ordre de la réaction, qui demeurent jusqu'ici inconnus. Par exemple, pour un procédé discontinu, des données expérimentales de concentrations en fonction du temps sont primordiales pour obtenir les constantes. Pour une réaction simple ou une espèce A se consomme dans un réacteur discontinu, on a :

$$\frac{dC_A}{dt} = -k (C_A)^\alpha \quad (2.16)$$

En intégrant cette équation différentielle ordinaire, on obtient la fonction analytique du temps de résidence :

$$t = \frac{1}{k} \left[\frac{C_{A0}^{(1-\alpha)} - C_A^{(1-\alpha)}}{(1-\alpha)} \right] \quad (2.17)$$

Dans le cas présent, la réaction ne contient qu'une seule étape et est irréversible, ce qui permet d'établir une solution analytique du temps de résidence assez facilement. Cependant, dans la très grande majorité des cas, les réactions sont réversibles et possèdent plusieurs

intermédiaires, rendant les équations de conception beaucoup plus complexes et leur solution analytique plus laborieuse à déterminer. Des méthodes numériques existent afin de résoudre les équations différentielles ordinaires (EDOs) et d'approximer les solutions analytiques. Grâce à ces méthodes numériques, les EDOs à condition initiale sont résolues sur une plage de temps, discrétisée avec des points espacés régulièrement par un pas de temps, Δt [20]. Par exemple, la méthode de Runge-Kutta permet d'avancer la solution dans le temps par le calcul de trajectoires entre deux pas de temps [18].

Somme toute, la solution $\mathbf{y}$ d'une EDO (ou d'un système d'EDOs) peut être approximée par une fonction $f(\mathbf{x}; \mathbf{b})$ déterminée via une méthode numérique appropriée. Dans le cas d'un réacteur discontinu, $\mathbf{y}$ est la concentration d'une espèce chimique qui évolue dans le temps, $\mathbf{x}$ représente la variable indépendante, soit le temps discrétisé par Δt et $\mathbf{b}$ sont les constantes inconnues, soit la constante cinétique k et l'ordre de la réaction α . Il est alors possible d'employer la méthode de Newton qui corrige les constantes $\mathbf{b}$ de sorte à optimiser, au sens des moindres carrés, les prédictions $\mathbf{y}$ selon les données expérimentales de concentration $\hat{\mathbf{y}}$ [4, 18].

2.3 Apprentissage profond

La Figure 2.8 montre la place de l'apprentissage profond en intelligence artificielle. L'intelligence artificielle (IA) est un concept qui englobe tous les algorithmes informatiques, performés par un ordinateur, permettant d'imiter et, potentiellement un jour, de surpasser l'intelligence humaine. L'apprentissage automatique, quant à lui, est une forme d'IA qui, à partir de données et d'expériences, tente d'apprendre une tâche précise par elle-même. Finalement, l'apprentissage profond est un type d'apprentissage automatique. L'apprentissage profond fait usage d'un réseau de neurones pour comprendre différents liens non linéaires et complexes entre les données qui lui sont fournies [22].

Les prochaines sections se basent sur [23] pour la revue des réseaux de neurones et sur [24] pour la revue des réseaux de neurones informés de la physique (sauf indication contraire). Ces références permettront d'offrir une vue d'ensemble sur les fondements de l'apprentissage profond. Une revue de la littérature qui reflète l'état de l'art des réseaux de neurones en mélange et en cinétique est approfondie dans les articles I et II respectivement.

2.3.1 Réseau de neurones

La littérature établit une analogie entre un réseau de neurones artificiel et un système nerveux humain. En effet, le système nerveux contient des milliards de neurones qui sont connectés

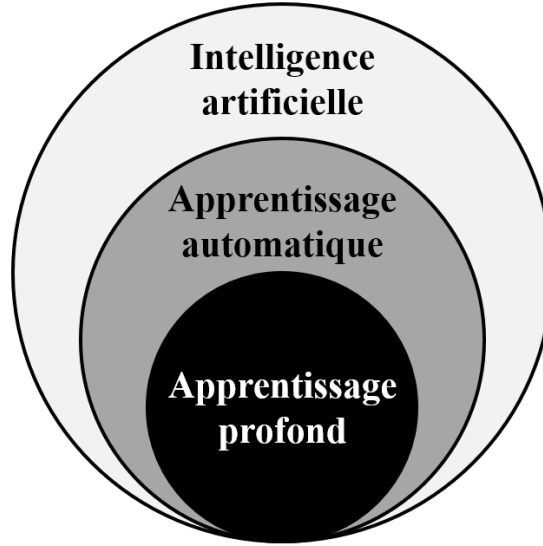


FIGURE 2.8 La place de l'apprentissage profond en intelligence artificielle.

entre elles par des axones et des synapses. Des impulsions électrochimiques sont transmises d'un neurone à l'autre afin de transmettre l'information et ultimement d'effectuer une tâche quelconque. L'algorithme du réseau de neurones suit le même principe, mais avec des données numériques, des opérations matricielles et des algorithmes d'optimisation.

Le réseau de neurones est reconnu pour faire de la classification ou de la régression. La classification permet de catégoriser une donnée, selon ses attributs, dans un groupe précis représenté par une variable discrète. La régression permet de trouver une corrélation entre plusieurs attributs d'un problème afin de prédire une variable continue.

Un réseau de neurones cherche donc à trouver une fonction hypothétique f qui approxime une certaine fonction f^* . Il permet de prédire une valeur y selon ses attributs $\mathbf{x}$, un vecteur élément de $\mathbb{R}^n$ contenu dans l'espace de la base de données $\mathbb{X} \in \mathbb{R}^{n \times m}$ où m est le nombre d'instances (ou d'enregistrements) disponibles dans la base de données et n est le nombre d'attributs à l'étude.

$$y = f(\mathbf{x}) \quad (2.18)$$

La Figure 2.9 illustre un réseau de neurones classique, avec deux attributs (représentés par x_1 et x_2) dans la couche d'entrée, deux couches cachées de quatre neurones et une sortie y dans la couche finale. Tous les neurones d'une couche sont connectés avec tous les neurones de la

couche qui la précède, ce qui forme, dans son ensemble, un réseau. Tout comme l'impulsion électrochimique entre les neurones du cerveau humain, les connexions entre les neurones d'un réseau possèdent chacune un certain poids w . Plus le poids d'une connexion est grand, plus le lien entre les neurones est important, ce qui a un impact majeur sur l'apprentissage et la prédiction de y .

Un réseau de neurones est considéré comme une forme d'apprentissage supervisé, c'est-à-dire qu'il ne peut apprendre sans données étiquetées. Il cherche à minimiser l'erreur, ou le coût, entre la prédiction y et sa valeur réelle $\hat{y}$ en deçà d'une certaine tolérance ε . Le réseau de neurones effectue donc une régression non linéaire via un algorithme d'optimisation.

Dans les prochaines sections, chaque étape de l'apprentissage du réseau de neurones sera présentée, allant de la passe avant permettant l'évaluation de y , l'algorithme d'optimisation, qui utilise la rétropropagation, et finalement son entraînement.

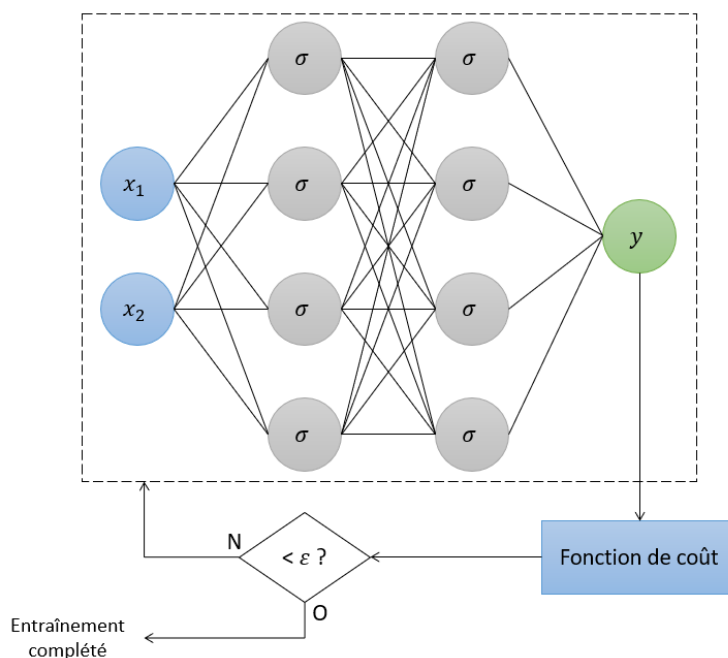


FIGURE 2.9 Réseau de neurones avec x_1 et x_2 les entrées, y la sortie et σ les neurones actifs.

Passe avant (*forward pass*)

La passe avant consiste à faire une prédiction $y = f(\mathbf{x})$ via le réseau, en passant par plusieurs couches de neurones interreliés entre eux. Chaque couche du réseau effectue une combinaison linéaire entre les neurones $\mathbf{x}$ et les poids des connexions $\mathbf{w}$.

$$f^{[i]} = \mathbf{x}^T \mathbf{w} + b \quad (2.19)$$

Ici, $f^{[i]}$ est la fonction donnée par une couche et b est un biais, une constante ajoutée à la combinaison linéaire. Le réseau est associé à un graphe cyclique dans lequel les fonctions des couches sont calculées en chaîne. Par exemple, pour un réseau de neurones ayant trois couches (une couche d'entrée, une cachée et une de sortie), la prédiction se fera comme suit :

$$f = f^{[3]} \left(f^{[2]} \left(f^{[1]}(\mathbf{x}) \right) \right) \quad (2.20)$$

La « magie noire » du réseau se cache derrière l'activation des neurones. En effet, les fonctions en chaîne décrites ci-haut effectuent une simple régression linéaire. En d'autres mots, les neurones sont « désactivés » de par le fait qu'ils proviennent de fonctions strictement linéaires. Pour activer un neurone, une fonction d'activation σ est appliquée. Cette fonction σ permet essentiellement au réseau d'apprendre des liens complexes et non linéaires entre les attributs du problème, rendant les prédictions du réseau riches et universelles. La fonction donnée par une couche devient alors :

$$f^{[i]} = \sigma(\mathbf{x}^T \mathbf{w} + b) \quad (2.21)$$

Il existe plusieurs fonctions d'activation utiles, comme la fonction sigmoïde et la fonction linéaire pour les neurones de la couche de sortie ainsi que la fonction tangente hyperbolique (*tanh*) et la fonction ReLU (*rectified linear unit*) pour les neurones des couches cachées [25]. Le Tableau 2.1 présente les différentes fonctions d'activation avec leurs utilités respectives et la Figure 2.10 montre l'allure de ces fonctions.

Le choix de la fonction d'activation est très important pour l'algorithme d'optimisation. La difficulté est de trouver un compromis entre la saturation et la stabilité des fonctions d'activation [25]. Une fonction qui sature signifie que son gradient est nul, ce qui entraîne un effet de neurones morts et ralentit grandement l'apprentissage. La stabilité d'une fonction d'activation repose sur l'effet zigzag (ou les oscillations) qu'elle occasionnera durant l'apprentissage.

En bref, pour toutes les couches du réseau, la passe avant consiste à :

TABLEAU 2.1 – Fonctions d’activation, leurs avantages et désavantages.

Activation	Fonction	Avantages	Désavantages
Sigmoïde	$s(x) = \frac{1}{1+e^{-x}}$	Prédiction d’une probabilité (pour les neurones de sortie)	Saturation, descente instable
Linéaire	$f(x) = x$	Prédiction d’une valeur continue (pour les neurones de sortie)	Aucune
Tanh	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Stabilité de la descente	Saturation, opérations coûteuses
ReLU	$f(x) = x, x > 0$ $f(x) = 0, \text{ sinon}$	Aucune saturation, convergence accélérée	Descente instable, effet de neurones morts

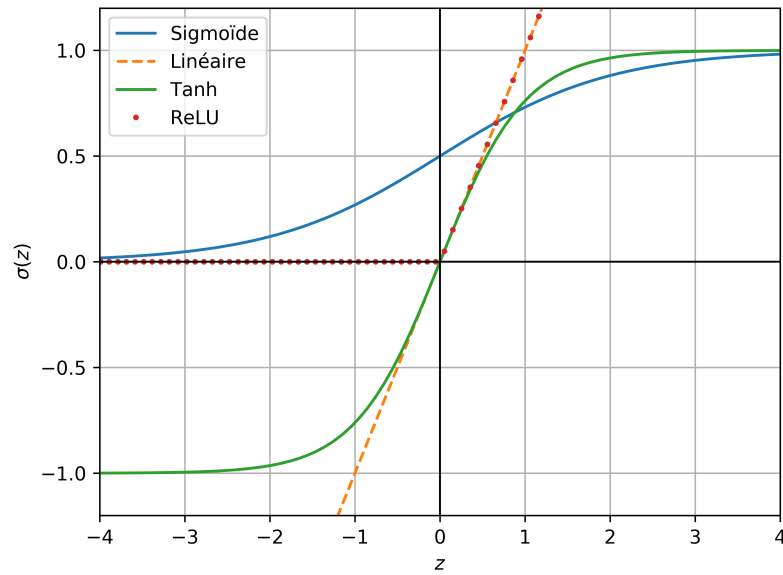


FIGURE 2.10 Fonctions d’activation.

1. effectuer une combinaison linéaire entre les neurones actifs d'une couche ainsi que les poids des connexions les reliant à la couche précédente ;
2. activer les neurones en suivant la fonction σ ;
3. répéter en suivant la structure en chaîne jusqu'à obtenir une prédiction y .

Il est à noter que la passe avant ne peut pas s'entamer sans initialiser les poids du réseau. Les poids doivent être initialisés convenablement, afin d'accélérer le début de l'apprentissage. La méthode de Glorot initialise les poids de manière aléatoire en suivant une loi uniforme sur le nombre de neurones présents dans les couches [26]. Cette méthode permet aux neurones de sortir de leur état saturé, ce qui engendre une convergence plus rapide dès le début de l'apprentissage.

Algorithme d'optimisation

Afin de déterminer si la fonction d'approximation est adéquate ou non, l'erreur entre les prédictions faites par f et les valeurs réelles est évaluée. Pour une architecture de réseau donnée, une « bonne » fonction hypothétique correspond à celle qui, avec ses poids ajustés adéquatement, permet d'avoir une erreur minimale sur ses prédictions. Pour obtenir une telle fonction, on résout un problème d'optimisation qui détermine les poids $\mathbf{w}$ du réseau qui minimisent l'erreur.

L'algorithme d'optimisation utilisé pour un réseau de neurones est la descente de gradient. Celle-ci consiste à mettre à jour les poids du réseau de manière itérative. Elle parcourt la fonction décrivant l'erreur globale (dénnotée J) et tente de descendre vers le minimum de cette fonction. Pour viser le minimum, l'algorithme nécessite la direction de la descente, c'est-à-dire le gradient de la fonction J . Le principe de l'algorithme de descente de gradient est exprimé comme suit :

$$\mathbf{w} := \mathbf{w} - \alpha \frac{\partial J}{\partial \mathbf{w}} \quad (2.22)$$

avec $\mathbf{w}$ les poids des connexions entre les neurones, α un taux d'apprentissage et J la fonction de perte. La fonction de perte est la moyenne des coûts (ou des erreurs) des prédictions de chaque enregistrement. Le taux d'apprentissage décrit la distance à laquelle l'algorithme descend la fonction. La Figure 2.11 illustre la descente de gradient, qui permet de converger vers une fonction de perte minimale en suivant ses dérivées.

Les fonctions de perte les plus utilisées pour les régressions sont : l'erreur quadratique

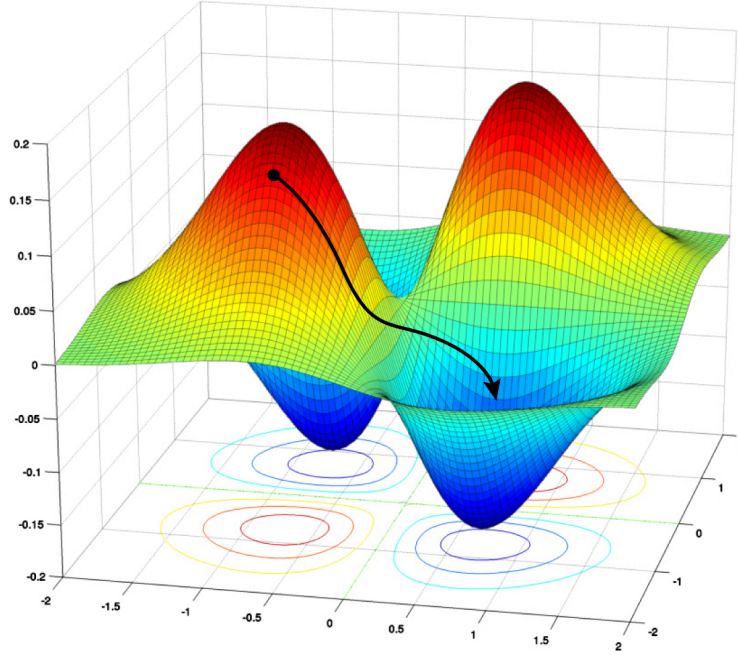


FIGURE 2.11 Descente de gradient.

moyenne (*mean squared error*, MSE), l'erreur absolue moyenne (*mean absolute error*, MAE) et l'erreur relative moyenne (*mean absolute percentage error*, MAPE) [25].

La fonction de perte utilisant l'erreur quadratique moyenne est alors écrite comme suit :

$$J = \frac{1}{m} \sum (f(\mathbf{x}; \mathbf{w}, b) - f^*(\mathbf{x}))^2 = \frac{1}{m} \sum_i^m (y^{(i)} - \hat{y}^{(i)})^2 \quad (2.23)$$

avec $\mathbf{x} \in \mathbb{X}$ (la base de données), $y^{(i)} = f(\mathbf{x}; \mathbf{w}, b)$ la prédiction du réseau selon les attributs d'une instance $\mathbf{x}$ et $\hat{y}^{(i)} = f^*(\mathbf{x})$ est la valeur réelle de cette même instance.

L'une des limitations majeures de la descente de gradient est l'atteinte du minimum global de manière rapide et efficace. Tout d'abord, des algorithmes améliorés de la descente du gradient permettent d'accélérer significativement l'apprentissage du réseau. Des stratégies, telles qu'Adam [27], ajoutent un vecteur momentum $\mathbf{m}$ à l'algorithme en se basant sur les gradients antérieurs de l'apprentissage [25].

$$\mathbf{m} := \beta \mathbf{m} - \alpha \frac{\partial J}{\partial \mathbf{w}} \quad (2.24)$$

$$\mathbf{w} := \mathbf{w} + \mathbf{m} \quad (2.25)$$

avec β , le momentum appliqué à la descente.

Une autre stratégie utilise une approximation de la matrice hessienne et emploie la méthode de (quasi) Newton, qui est d'ordre 2, afin de mettre à jour les poids du réseau [28].

Ensuite, la fonction de perte n'est rarement, voire jamais convexe. Cela fait apparaître plusieurs minimums locaux qui empêchent l'algorithme d'atteindre la solution optimale. Pour contrer ce problème, au lieu de passer tout le lot de données (*batch*) au réseau, la mise à jour des poids se fait sur une donnée à la fois (stochastique) ou sur une sous-partie de la base de données (*mini-batch*). Cela permet d'explorer davantage la fonction de perte et de détecter plus facilement le potentiel d'un minimum global [25]. La Figure 2.12 montre la différence entre une descente stochastique, par *mini-batch* et par *batch*. Dans les trois cas, le minimum global est trouvé. Par contre, malgré le fait que la descente stochastique est plus erratique, elle permet d'explorer davantage la fonction de perte autour du minimum, dans la possibilité de tomber dans un second minimum plus près de la solution optimale (si elle existe).

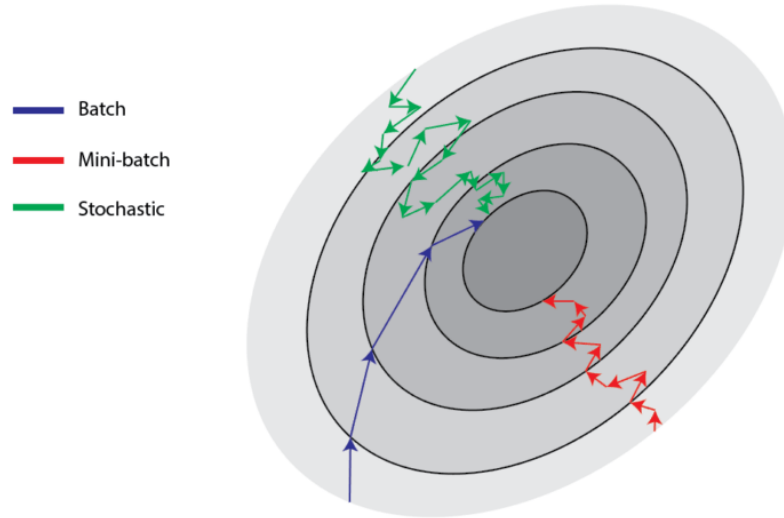


FIGURE 2.12 Effet du *batch size* sur la descente de gradient.

Rétropropagation

Le calcul du gradient de J par rapport à un poids se fait par rétropropagation [29]. La rétropropagation est une méthode utilisant le principe des dérivées en chaîne, partant de la fin du réseau jusqu'à son commencement. Pour le calcul du gradient de J par rapport à un certain poids se trouvant dans une couche, la dérivée en chaîne se fait comme suit :

$$\frac{\partial J}{\partial w^{[i]}} = \frac{\partial J}{\partial \sigma^{[i]}} \frac{\partial \sigma^{[i]}}{\partial z^{[i]}} \frac{\partial z^{[i]}}{\partial w^{[i]}} \quad (2.26)$$

avec

$$\sigma^{[i]} = \sigma(z^{[i]}) = \sigma(w^{[i]} \sigma^{[i-1]} + b^{[i-1]}) \quad (2.27)$$

Seule la dérivée partielle de J par rapport au neurone actif $\sigma^{[i]}$ n'est pas connue. Elle nécessite alors la dérivée partielle de J par rapport au neurone actif $\sigma^{[i+1]}$ de la couche suivante dont celle-ci a déjà été évaluée préalablement durant la rétropropagation et est donc connue. La Figure 2.13 illustre les dérivées en chaîne à évaluer pour le calcul de la dérivée partielle de J par rapport à un certain poids $w^{[i]}$ du réseau.

$$\frac{\partial J}{\partial \sigma^{[i]}} = \frac{\partial J}{\partial \sigma^{[i+1]}} \frac{\partial \sigma^{[i+1]}}{\partial z^{[i+1]}} \frac{\partial z^{[i+1]}}{\partial \sigma^{[i]}} \quad (2.28)$$

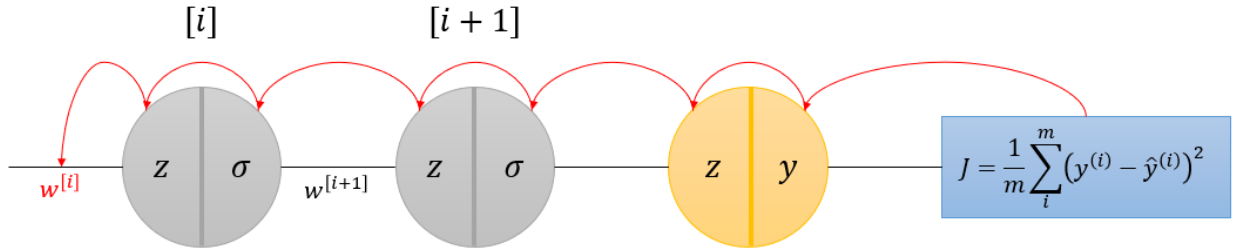


FIGURE 2.13 Exemple d'une rétropropagation pour le calcul de $\partial J / \partial w^{[i]}$.

Entraînement

Une itération (appelé *epoch*) de l'entraînement du réseau de neurones, nécessite :

1. une passe avant, qui prédit la sortie y , en effectuant des produits scalaires entre les neurones et leurs poids respectifs ainsi qu'en les activant via une fonction d'activation ;
2. une évaluation de la fonction de perte J ;
3. une rétropropagation qui évalue les gradients de la fonction J par rapport aux poids du réseau ;
4. une descente de gradient qui met à jour les poids selon la direction donnée par les gradients et un taux d'apprentissage donnant la distance de la descente.

Les grandes difficultés des réseaux de neurones reposent sur i) la qualité et le volume des données et ii) le choix des hyperparamètres, soit les paramètres permettant de contrôler le processus d'apprentissage. Les hyperparamètres sont nombreux dans un réseau de neurones : le nombre de neurones, le nombre de couches, la fonction d'activation, la fonction de perte (MSE, MAE, MAPE, etc.), et l'algorithme d'optimisation.

Plusieurs variantes d'un réseau de neurones existent pour répondre à différents problèmes de régression non linéaire. Par exemple, les réseaux de neurones convolutifs (*convolutional neural networks*, CNN) sont utiles pour l'analyse d'images en exploitant la convolution de la base de donnée $\mathbb{X}$ [30]. Il existe également les réseaux de neurones récurrents (*recurrent neural network*, RNN) qui créent des cycles au sein de neurones afin de prédire les comportements dynamiques de données temporelles [31]. Les réseaux de neurones informés par la physique, quant à eux, tentent d'approximer la solution d'équations différentielles, et sont discutés de manière plus exhaustive dans la section suivante.

2.3.2 Réseau de neurones informé par la physique

Un réseau de neurones informé par la physique (*physics-informed neural network*, PINN) est construit de la même manière qu'un réseau classique à une exception près : la fonction de perte est ajustée selon la physique du problème à l'étude. Des équations différentielles ordinaires (EDOs) ou bien des équations différentielles partielles (EDPs) ajoutent une contrainte mathématique au modèle du réseau. Le PINN prédit une solution $\mathbf{u}$ qui tente d'approximer la solution analytique $\mathbf{u}^*$ de ces équations.

Les attributs d'un PINN (ou *inputs*) diffèrent légèrement d'un réseau de neurones traditionnel. Les entrées d'un PINN vont inclure les variables indépendantes du problème. Pour une

EDO, une seule variable indépendante est considérée, par exemple le temps t . Pour une EDP, l'espace du problème ($\mathbf{x} = (x, y, z)$) est ajoutée.

L'avantage du PINN par rapport aux méthodes numériques de résolution d'équations différentielles repose sur le fait qu'il nécessite aucun maillage. Ce sont plutôt des points de collocation, sans aucune connexion, qui décriront, de manière discrète, le domaine Ω du problème à résoudre. Pour chaque point de collocation, la physique est résolue. Contrairement à un réseau classique, le volume de données étiquetées nécessaires pour alimenter le PINN est nettement moins considérable, car les points de collocation remplacent les données étiquetées. Chaque point prédit une valeur dont celle-ci est évaluée par l'équation différentielle. Moins l'équation est respectée, plus le coût est élevé, demandant davantage d'entraînement par le PINN pour satisfaire l'équation. Les seules données étiquetées nécessaires à l'apprentissage sont les conditions initiales (pour une EDO), les conditions sur les frontières délimitées par Γ (pour une EDP) et certaines mesures. La Figure 2.14 présente les points de collocation d'une EDO et sa condition initiale alors que la Figure 2.15 présente les points de collocation d'une EDP et ses conditions frontières.



FIGURE 2.14 Points de collocation (●), conditions initiales (○) pour une EDO.

Ainsi, le problème à résoudre par le PINN consiste à résoudre une équation de type :

$$\mathbf{u}_t + F(\mathbf{u}, \boldsymbol{\lambda}) = 0 \quad (2.29)$$

avec u_t la dérivée partielle de $\mathbf{u}$ (la solution latente, ou la prédiction faite par le PINN) par rapport au temps $t \in [0, T]$, $\boldsymbol{\lambda}$ les paramètres ou les constantes de l'équation et $F(\mathbf{u}, \boldsymbol{\lambda})$ la fonction du membre de droite d'une équation différentielle (souvent non linéaire selon u et λ). Dans le cas d'une EDP, cette fonction est une opération différentielle selon les coordonnées du problème $\mathbf{x} \in \Omega$.

Il est à noter que l'équation 2.29 est écrite sous sa forme résiduelle. Le résidu de l'équation différentielle est dénoté R comme suit :

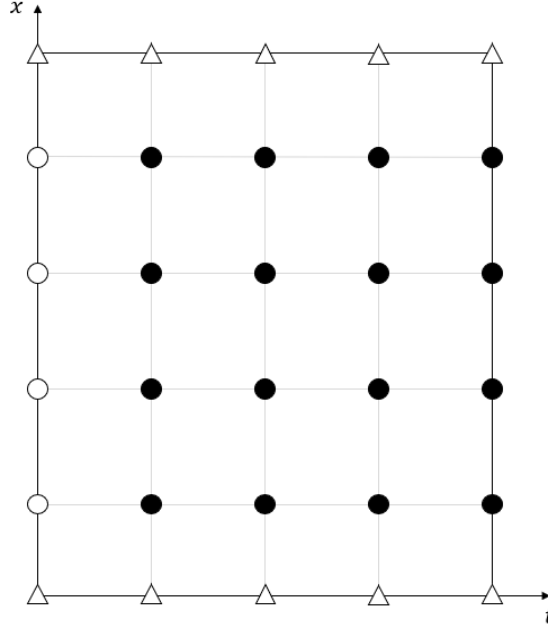


FIGURE 2.15 Points de collocation (●), conditions initiales (○) et conditions frontières (△) pour une EDP.

$$R := \mathbf{u}_t + F(\mathbf{u}, \boldsymbol{\lambda}) \quad (2.30)$$

Les différences entre le PINN et un réseau de neurones classique seront présentées dans les sections suivantes. D'une part, la fonction de perte sera détaillée afin d'expliquer comment celle-ci inclut les équations différentielles. Ensuite, étant donné que l'architecture particulière du PINN ajoute une étape supplémentaire à son entraînement, celui-ci est donc revisité brièvement. Finalement, une attention particulière sera portée sur sa capacité à résoudre des problèmes inverses.

Fonction de perte

Dorénavant, la fonction de perte n'inclut plus seulement l'erreur entre les prédictions et les données étiquetées, elle inclut également l'erreur entre les prédictions et leur accord avec les équations décrivant la physique. La fonction de perte du PINN est alors :

$$J = J_r + J_u \quad (2.31)$$

avec J_r la fonction de perte du résidu évaluée aux points de collocation du domaine Ω , J_u celle des conditions initiales, des conditions frontières sur Γ et des mesures. Pour la fonction de perte J_r , celle-ci évalue l'erreur quadratique moyenne du résidu d'une EDP comme suit :

$$J_r = \frac{1}{N_f} \sum |R(t, \mathbf{x})|^2 \quad (2.32)$$

avec $t \in T \setminus \{0\}$, $\mathbf{x} \in \Omega$, N_f le nombre de points de collocation et $R(t, \mathbf{x})$ le résidu évalué à un point de collocation dans le domaine de calcul. Pour la fonction de perte J_u , celle-ci évalue l'erreur quadratique moyenne des prédictions du réseau comme suit :

$$J_u = \frac{1}{N_u} \sum |\mathbf{u}(t, \mathbf{x}) - \mathbf{u}^*(t, \mathbf{x})|^2 = \frac{1}{N_u} \sum_i^{N_u} |u^{(i)} - \hat{u}^{(i)}|^2 \quad (2.33)$$

avec $t \in T$, $\mathbf{x} \in \Omega \cup \Gamma$, N_u le nombre de points là où une condition ou une mesure s'appliquent, $u^{(i)}$ la prédiction d'une condition ou d'une mesure faite par le réseau au point de collocation (i) et $\hat{u}^{(i)}$ la réelle valeur de la condition ou de la mesure au même point. À noter que, pour une EDO, $\mathbf{x}$ est simplement inexistant.

Entraînement

La Figure 2.16 montre un exemple d'un PINN avec t et x en entrée, 2 couches cachées de 4 neurones, la couche de sortie incluant une solution latente u , la couche des dérivées partielles de l'EDP (u_t la dérivée selon le temps, u_x la dérivée selon la dimension x et u_{xx} la dérivée seconde) et R la prédiction du résidu. Afin d'évaluer le résidu d'une EDO ou d'une EDP, le PINN utilise la différentiation automatique. En effet, tout comme la rétropropagation, l'emploi de la règle de dérivation en chaîne s'avère très utile pour évaluer les dérivées partielles des équations différentielles. Une autre façon d'évaluer le résidu est d'utiliser la méthode de Runge-Kutta. Le PINN tente alors de prédire toutes les trajectoires menant de la condition initiale vers la solution au temps final.

Tout comme un réseau de neurones classique, un PINN doit effectuer une itération en sui-

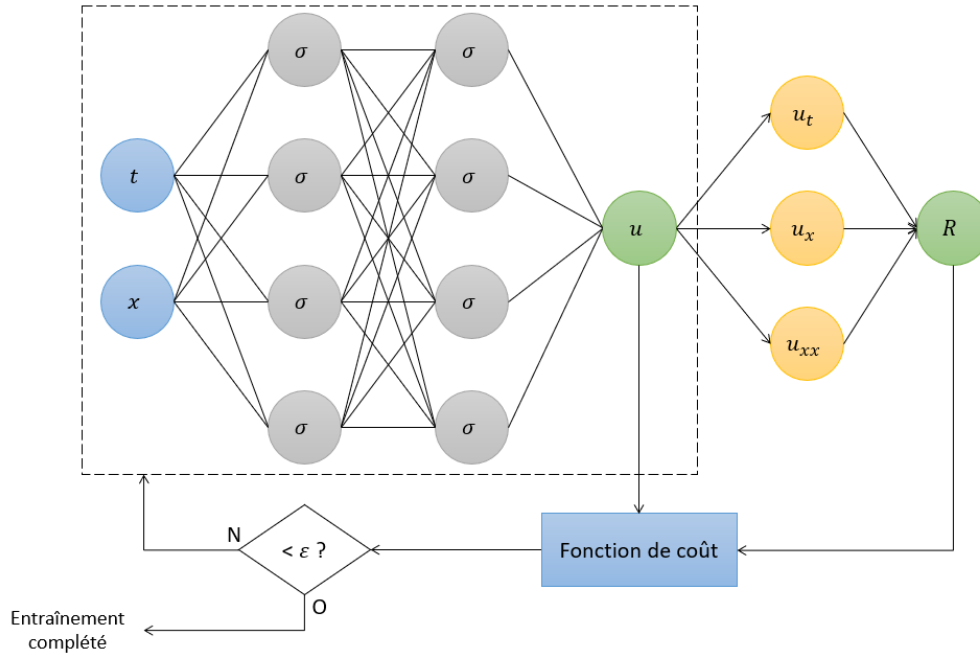


FIGURE 2.16 Réseau de neurones informé par la physique (PINN).

vant les mêmes étapes afin d'avancer son apprentissage, à quelques exceptions près qui sont adaptées au PINN :

1. une passe avant, qui prédit la solution u du problème, en effectuant des produits scalaires entre les neurones et leurs poids respectifs ainsi qu'en les activant via une fonction d'activation ;
2. une différentiation automatique qui évalue les dérivées partielles de la solution u par rapport à ses entrées (t et/ou x) ;
3. une évaluation de la fonction de perte J , qui additionne la fonction de perte du résidu évalué aux points de collocation à la fonction de perte des conditions initiales et/ou des conditions frontières et des mesures (si disponibles) ;
4. une rétropropagation qui évalue les gradients de la fonction J par rapport aux poids du réseau ;
5. une descente de gradient qui met à jour les poids selon la direction donnée par les gradients et un taux d'apprentissage donnant la distance de la descente.

Problème inverse

Si λ de l'Équation 2.29 est un paramètre connu de l'EDO ou l'EDP, on résout un problème direct, tandis que si λ est inconnu, on résout alors un problème inverse. Dans un réseau de neurones, λ devient un paramètre « entraînable », tout comme les poids du réseau. Ainsi, de manière itérative, λ est corrigé à l'aide de l'algorithme d'optimisation, soit la descente de gradient.

$$\lambda := \lambda - \alpha \frac{\partial J}{\partial \lambda} \quad (2.34)$$

Étant donné qu'un problème inverse introduit des équations différentielles non complètes, c'est-à-dire que la valeur de ses paramètres reste encore à être découverts, le PINN nécessite davantage de mesures $\hat{u}^{(i)}$, provenant d'expériences, afin d'aider la convergence du PINN vers un minimum idéalement global. Les Figures 2.17 et 2.18 illustrent qu'à certains points de collocations dans le temps et/ou dans l'espace, la solution est connue via une expérience. Ainsi, la fonction de coût ajoute un terme représentant l'erreur entre la solution prédite $u^{(i)}$ et une observation $\hat{u}^{(i)}$.

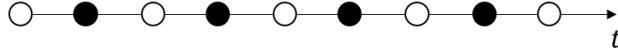


FIGURE 2.17 Points de collocation (●), conditions initiales et données connues (○) pour une EDO.

Comme mentionné précédemment, les fonctions de perte ne sont pas convexes, faisant apparaître des minimums locaux. L'ajout d'une équation différentielle à la fonction de perte influence grandement son paysage, le rendant plus chaotique [32]. Cela fait en sorte que la découverte du minimum global est assez laborieuse, plus encore lorsque l'équation différentielle en question comporte des paramètres inconnus.

Une des méthodes proposée par [32] est d'ajouter un paramètre de régularisation η_r à la fonction de perte du résidu J_r .

$$J = \eta_r J_r + J_u \quad (2.35)$$

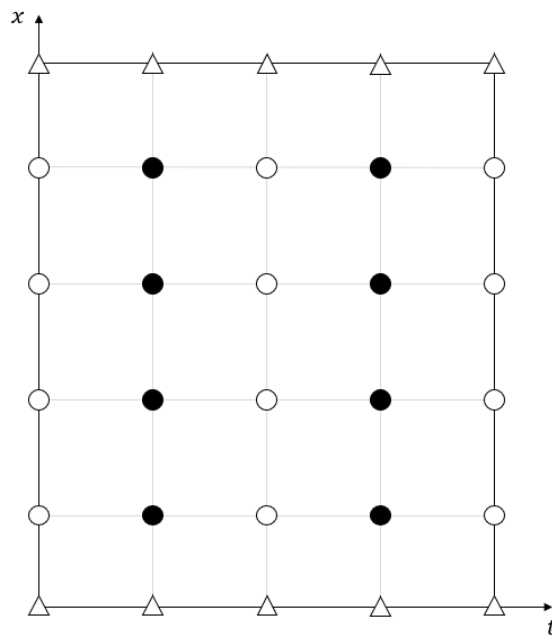


FIGURE 2.18 Points de collocation ($\bullet$), conditions initiales et données connues ($\circ$), conditions frontières (Δ) pour une EDP.

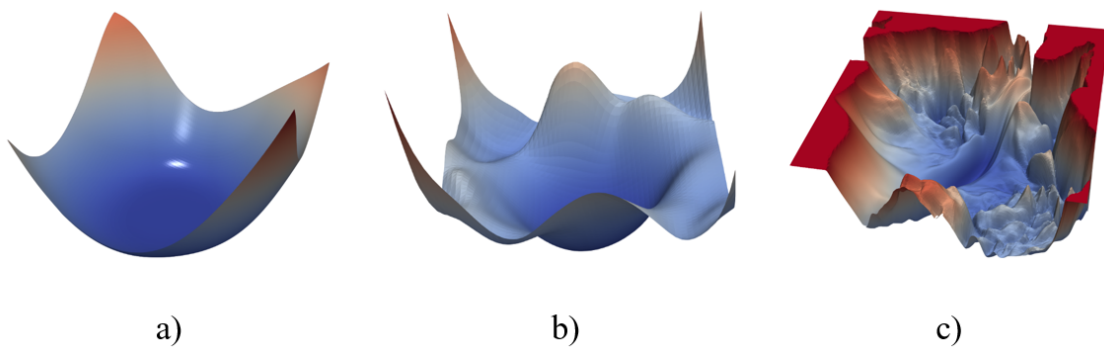


FIGURE 2.19 Visualisation de la fonction de perte, adapté de [5]. a) Fonction parfaitement convexe. b) Fonction de référence. c) Fonction chaotique.

Il a été démontré par [32] que le paramètre de régularisation, pour une valeur faible, peut alléger la complexité du paysage de la fonction de perte, rendant la convergence du PINN vers un minimum global plus facile. Cependant, une régularisation peu sévère fait en sorte que les équations différentielles sont moins respectées. Pour mitiger ce problème, deux solutions sont proposées par [32]. La première est de progressivement augmenter la valeur du paramètre de régularisation, ce qui permet au PINN de commencer son entraînement sur des contraintes plus simples et de terminer sur des contraintes physiques renforcées. Ainsi, à la fin de l’entraînement, le PINN est en mesure de prédire des solutions qui respectent davantage les équations différentielles. La deuxième stratégie est de découper le domaine en plusieurs sous-domaines, ce qui rend l’entraînement du PINN plus efficace.

Plusieurs autres avancées sur les PINNs, qui sont très récentes (entre 2019 et aujourd’hui), ont été accomplies afin d’extrapoler leur utilisation à différents problèmes. Dans la revue de littérature, on remarque que chaque nouvelle découverte sur les PINNs, une lettre est ajoutée en préfixe à l’acronyme.

- Les VPINNs emploient le calcul variationnel des EDP [33].
- Les BPINNs emploient un réseau de neurones bayésien afin de faire face au bruit des données [34].
- Les XPINNs décomposent le domaine du temps en plusieurs sous-domaines afin d’améliorer la convergence et la parallélisation durant l’entraînement [35].

Limitations des PINN

Il est important de prendre un certain recul et de se poser la question si les PINNs peuvent vraiment surpasser les méthodes numériques conventionnelles, comme la méthode des éléments finis (MEF). On assiste alors à un affrontement entre deux paradigmes, celui d’utiliser des méthodes numériques bien implémentées depuis les années 1960 [36], versus celui d’employer des algorithmes d’apprentissage profond. Afin de comparer les deux approches, la précision et le coût de calcul sont analysés pour la résolution de l’équation de Poisson en 1D, 2D et 3D [37]. Il a été démontré par [37] que les PINNs ne sont pas en mesure de battre la MEF. En effet, pour une même précision, la MEF est jusqu’à 3 fois plus rapide que le temps d’entraînement du PINN. Ces résultats permettent d’avoir une vision différente sur comment les PINNs peuvent être utilisés à leur plein potentiel. C’est pourquoi, dans l’article II, le PINN n’est pas utilisé que pour la résolution d’une équation mathématique. On tente de mettre de l’avant son avantage à résoudre des problèmes inverses ainsi que sa capacité d’interpolation et d’extrapolation à d’autres attributs d’un problème, autre que les variables indépendantes classiques du temps et de l’espace.

2.4 Synthèse et objectifs de recherche

La revue de la littérature a permis de faire un survol assez large des méthodes de conception d'opérations unitaires. La Figure 2.20 illustre bien la façon dont certaines opérations unitaires, dont les mélangeurs et les réacteurs, ont convergé vers plusieurs méthodes numériques permettant leur conception.

D'une part, afin de déterminer l'efficacité énergétique de mélangeurs, le nombre de puissance est étudié. Il peut être calculé en résolvant numériquement les équations différentielles partielles (EDP) de Navier-Stokes. Il peut aussi être déterminé via des corrélations polynomiales dont les exposants sont établis par une régression non linéaire. Le premier objectif est donc d'utiliser les résultats numériques de mélangeurs, simulés à l'aide de la méthode des éléments finis (MEF), pour développer un outil d'apprentissage profond. Cet outil permet d'établir une fonction du nombre de puissance selon les conditions physiques du fluide et la géométrie de l'agitateur. Ainsi, on tente de démontrer la possibilité de remplacer les corrélations algébriques existantes par des réseaux de neurones. L'objectif est illustré par la flèche bleue sur la Figure 2.20.

D'autre part, le temps de résidence est une variable de conception importante des réacteurs discontinus. Cette variable provient d'équations de conception, où une équation différentielle ordinaire (EDO) permet de mettre en relation l'évolution de la concentration d'une espèce chimique selon son taux de génération (ou de consommation). Lorsque les réactions sont multiples, complexes et prennent en compte plusieurs espèces en même temps, les EDOs présentent des formes hautement non linéaires. Des méthodes numériques, dont celle de Runge-Kutta (RK), permettent leur résolution. Celles-ci peuvent être couplées avec une régression non linéaire afin de découvrir les constantes inconnues des réactions. Le deuxième objectif est d'utiliser les données expérimentales d'un réacteur discontinu, faisant l'usage de micro-ondes, pour développer un outil d'apprentissage profond. Un réseau de neurones informé par la physique permet de découvrir les constantes cinétiques d'un procédé de production de biodiesel qui respectent la physique du problème, c'est-à-dire les EDOs provenant des bilans molaires. Ultimement, l'objectif est de créer un jumeau numérique du réacteur, en exploitant le potentiel d'extrapolation du réseau à plusieurs conditions d'opération. L'objectif est illustré par la flèche orange sur la Figure 2.20.

À remarquer qu'en regroupant les deux objectifs, on tente d'avoir une vue d'ensemble sur le potentiel des réseaux de neurones à utiliser les données et les modèles physiques. En ce qui concerne les mélangeurs, on fait l'usage de beaucoup de données, provenant de simulations numériques. Autrement dit, les équations de Navier-Stokes sont résolues à l'extérieur du ré-

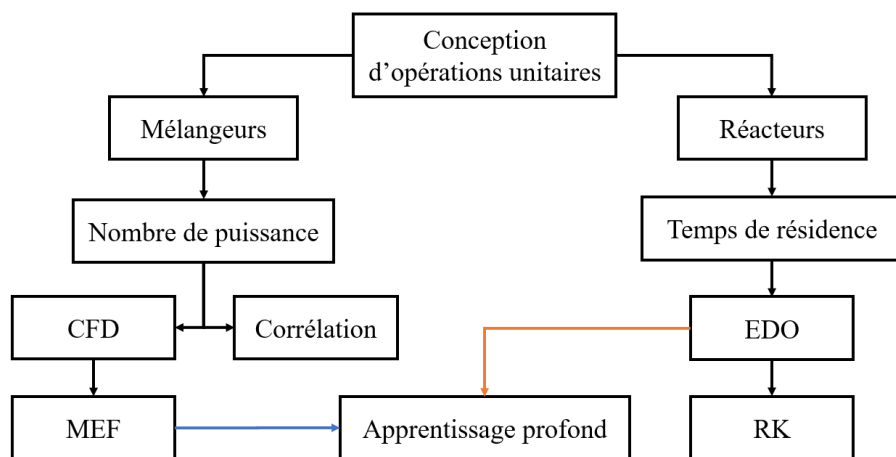


FIGURE 2.20 Diagramme sommaire de la revue de littérature.

seau, par l'emploi de la MEF, et servent à alimenter le réseau de neurones. Pour les réacteurs, c'est tout à fait le contraire. Peu de données expérimentales du réacteur à micro-ondes sont alimentées au réseau. Par contre, les bilans molaires et les lois de cinétique chimique sont implémentés au sein de l'algorithme du réseau de neurones. La Figure 2.21 illustre ce propos.



FIGURE 2.21 L'utilisation des données versus l'emploi de modèles physiques pour les deux objectifs, inspiré de [6].

Somme toute, les deux approches ont des objectifs communs, soit i) de réduire le plus possible le passage au laboratoire, que ce soit par l'entremise de simulations numériques sur un ordinateur ou en diminuant le nombre d'expériences à faire et ii) d'explorer les avantages des réseaux de neurones à remplacer les méthodes de régression non linéaire classiques.

CHAPITRE 3 MÉTHODOLOGIE ET ORGANISATION

3.1 Méthodologie associée aux objectifs

L'hypothèse de ce présent mémoire est de démontrer que les outils d'apprentissage profond pour des applications d'ingénierie chimique telles que la prédiction de la puissance d'agitation de mélangeurs et l'analyse de la cinétique chimique d'une réaction à micro-ondes sont faciles à implémenter, précis et performants. Ce mémoire est donc séparé en deux objectifs principaux qui traitent chacune des applications (le *quoi*). Les étapes de la méthodologie permettant l'atteinte de chaque objectif sont jointes aux sous-objectifs (le *comment*).

- **Premier objectif principal** : Développer un réseau de neurones pour prédire le nombre de puissance d'agitation de mélangeurs à partir de simulations CFD.

S.O.1 : Créer une base de données de simulations de mélangeurs et leur nombre de puissance.

1. Paramétrer et mailler le domaine de calcul des mélangeurs.
2. Vérifier la sensibilité du maillage sur l'erreur des simulations.
3. Échantillonner 100 000 mélangeurs selon une plage de paramètres géométriques et physiques.
4. Lancer les 100 000 simulations des mélangeurs sur plusieurs nœuds de calcul à l'aide de Lethe, un logiciel de CFD.

S.O.2 : Développer une architecture d'un réseau de neurones permettant de prédire le nombre de puissance d'un agitateur.

1. Programmer l'architecture du réseau de neurones avec la librairie TensorFlow.
2. Faire une recherche exhaustive par validation croisée sur les hyperparamètres du réseau de neurones.

S.O.3 : Évaluer la performance du réseau de neurones.

1. Évaluer la métrique de performance sur un lot de données de test.
2. Comparer les prédictions du nombre de puissance avec des corrélations présentes dans la littérature.

- **Deuxième objectif principal** : Développer un réseau de neurones informé de la physique pour prédire les constantes cinétique de la réaction de transestérification à partir de données expérimentales.

S.O.1 : Générer une base de données des expériences en laboratoire de la réaction de transestérification.

1. Établir un plan d'expérience qui explore les différents paramètres affectant la cinétique chimique tels que le temps de réaction et la puissance du réacteur à micro-ondes.
2. Recueillir les données de température et de puissance du réacteur à micro-ondes pour chaque expérience.
3. Recueillir les données de concentration de chaque espèce de la réaction via chromatographie gazeuse pour chaque expérience.

S.O.2 : Développer une architecture d'un réseau de neurones informé de la physique permettant de prédire les constantes cinétiques de la réaction de transestérification.

1. Programmer l'architecture du réseau de neurones avec PyTorch.
2. Vérifier l'implémentation du code avec des données numériques artificielles.
3. Adapter l'architecture et la fonction de perte du réseau pour répondre à la cinétique et la conservation d'énergie du réacteur à micro-ondes.

S.O.3 : Évaluer la performance du réseau de neurone informé de la physique.

1. Comparer les prédictions du réseau aux données expérimentales.
2. Comparer les prédictions du réseau à une méthode numérique.
3. Vérifier la capacité d'extrapolation du réseau à divers conditions d'opération.

3.2 Organisation du mémoire

Il est à noter que deux articles, présentant chacune une application, serviront de soutien à ce mémoire et aideront à répondre à ces multiples objectifs. Les Tableaux 3.1 et 3.2 indiquent dans quelle section du mémoire se trouvent les détails de la méthodologie permettant l'atteinte des sous-objectifs.

TABLEAU 3.1 – Section du mémoire associé aux sous-objectifs du premier objectif principal.

Sous-objectif	Méthodologie	Section du mémoire
S.O.1.1	Paramétrage et maillage	Section 4 (article I)
S.O.1.2	Sensibilité du maillage	Section 4 (article I)
S.O.1.3	Échantillonnage	Section 4 (article I)
S.O.1.4	Simulations avec Lethe	Section 3
S.O.2.1	Programme TensorFlow	Section 3
S.O.2.2	Recherche d'hyperparamètres	Section 4 (article I)
S.O.3.1	Métrique de performance	Section 4 (article I)
S.O.3.2	Prédictions versus corrélations	Section 4 (article I)

TABLEAU 3.2 – Section du mémoire associé aux sous-objectifs du deuxième objectif principal.

Sous-objectif	Méthodologie	Section du mémoire
S.O.1.1	Plan d'expérience	Section 6 (article II)
S.O.1.2	Données du réacteur	Section 6 (article II)
S.O.1.3	Données du GC-FID	Section 6 (article II)
S.O.2.1	Programme PyTorch	Section 3
S.O.2.2	Données numériques artificielles	Annexe A
S.O.2.3	Fonction de perte	Section 6 (article II)
S.O.3.1	Prédictions versus expériences	Section 6 (article II)
S.O.3.2	Prédictions versus numérique	Section 6 (article II)
S.O.3.3	Extrapolation	Section 6 (article II)

3.3 Bibliothèques Python en apprentissage profond

Pour implémenter un réseau de neurones facilement et rapidement, le langage de programmation Python est le plus utilisé. Plusieurs bibliothèques en Python offrent des méthodes permettant de construire, de compiler et d'entraîner un réseau de neurones. Les deux plus populaires sont TensorFlow [38], développé par Google, et PyTorch [39], développé par Meta. Chaque bibliothèque possède leurs avantages et leurs inconvénients qui seront discutés ici. La discussion se base principalement sur [40] (sauf indication contraire) et sur mon expérience personnelle,

puisque j’ai eu la chance de travailler avec les deux librairies pour l’accomplissement de cette maîtrise.

TensorFlow

TensorFlow est reconnu pour sa bonne documentation, sa communauté vibrante ainsi que ses méthodes permettant de bien visualiser les données, l’évolution de l’entraînement et les prédictions. TensorFlow est considéré plus performant puisqu’il se base sur un design plus statique, c’est-à-dire que le graphe du réseau est construit et compilé avant d’être exécuté. TensorFlow gère la mémoire plus efficacement, rendant l’entraînement plus rapide et efficace. Ainsi, si le modèle est plus imposant et que les données d’entraînement sont plus volumineuses, TensorFlow reste un bon choix de librairie. Finalement, TensorFlow peut paraître plus convivial. En effet, le code est plus compact et facile à suivre.

Pour le premier objectif principal, celui qui vise à développer un réseau de neurones pour prédire la puissance d’agitation dans les mélangeurs, la librairie TensorFlow est utilisée. Étant donné que cet objectif est le préambule de la maîtrise, cette librairie permettait un premier saut moins brusque dans l’univers des réseaux de neurones. De plus, étant donné que la base de données est plutôt volumineuse (100 000 simulations de mélangeurs), TensorFlow peut s’avérer plus performant. L’algorithme du réseau de neurones, nommé Mixing-ANN, ainsi que sa base de données se trouvent dans un référentiel libre d’accès Github [41].

PyTorch

PyTorch est considéré comme un langage plus « pythonique », rendant sa programmation plus facile et son débogage plus commode. Contrairement à TensorFlow, PyTorch se base sur un design dynamique. L’implémentation de l’architecture, la passe avant, l’évaluation de la fonction de perte et la rétropropagation sont définies séparément. Le graphe est donc construit au fur et à mesure que le code s’exécute. Le code en PyTorch est donc moins compact que TensorFlow, ce qui laisse une plus grande flexibilité de programmation et une meilleure capacité d’adaptation au problème à résoudre. Par contre, PyTorch peut souffrir de la surcharge de l’interpréteur de Python, rendant le code moins performant.

Pour le deuxième objectif principal, soit de développer un réseau de neurones informé de la physique (PINN) pour prédire la cinétique d’un procédé de biodiesel, la librairie PyTorch est utilisée. Étant donné que l’implémentation du PINN est plus complexe, PyTorch permet d’écrire une fonction de perte plus élaborée et un entraînement plus adapté, tout ça avec plus de flexibilité. L’algorithme du PINN, nommé Bio-PINN, ainsi que sa base de données se

trouvent dans un référentiel libre d'accès Github [42].

Comparaison des structures de code

Le Tableau 3.3 présente la structure du code en TensorFlow et en PyTorch pour l'implémentation d'un réseau de neurones comportant 2 attributs, 1 couche cachée de 10 neurones et 1 sortie. Les réseaux utilisent la fonction tangente hyperbolique (tanh) pour l'activation des neurones et emploient l'optimisateur Adam pour la descente de gradient. Les codes utilisent X qui correspond au tenseur d'entrée contenant les attributs des instances et Y qui est le tenseur de sortie contenant les données étiquetées. Les codes sont séparés en quatre sections : l'importation des librairies, l'implémentation de l'architecture, la compilation du modèle et l'entraînement.

La structure de code de TensorFlow est en effet plus compacte. La compilation et l'entraînement se font respectivement en une seule ligne. Dans le cas contraire, PyTorch nécessite plus de lignes de code pour exécuter la passe avant et l'entraînement par descente de gradient. Par contre, le style plus « pythonique » de PyTorch permet de mieux vérifier ce qui se passe à chaque ligne.

TABLEAU 3.3 – Distinction de la structure de code entre TensorFlow et PyTorch.

TensorFlow	PyTorch
<pre>import tensorflow as tf import keras</pre>	<pre>import torch import torch.nn as nn</pre>
<pre>model = Sequential() model.add(Dense(neurons, input_dim = 2, activation = 'tanh')) model.add(Dense(neurons, activation = 'tanh')) model.add(Dense(1, activation = 'linear'))</pre>	<pre>class NN(nn.Module): def __init__(self): self.layer1 = nn.Linear(2, neurons) self.layer2 = nn.Linear(neurons, neurons) self.layer3 = nn.Linear(neurons, 1)</pre>
<pre>model.compile(loss = 'mse', optimizer = optimizer)</pre>	<pre>def forward(self, x): z1 = layer1(x) a1 = nn.Tanh(z1) z2 = layer2(a1) a2 = nn.Tanh(z2) a3 = layer3(a2)</pre>
<pre>model.fit(X, Y, epochs = epochs)</pre>	<pre>NN = NN() params = NN.parameters() optimizer = torch.optim.Adam(params) for _ in epochs: optimizer.zero_grad() criterion = nn.MSELoss() loss = criterion(NN(X),Y) loss.backward()</pre>

3.4 Lethe

Un logiciel libre, basé sur la méthode des éléments finis, est utilisé afin de simuler les 100 000 mélangeurs sur les grappes de calcul. Lethe est un logiciel de CFD permettant de résoudre les écoulements incompressibles. Lethe se base en grande partie sur la librairie de deal.II [3] qui emploie des méthodes avancées de résolution matricielle (Trillinos). De plus, Lethe peut raffiner le maillage de manière adaptative, rendant la résolution en vitesse et en pression plus précise (p4est). Il est également possible d'exécuter le logiciel sur plusieurs cœurs de calcul en parallèle, permettant de diminuer les coûts de calcul lorsque le degré de liberté devient important. Depuis sa publication en 2020 [43], Lethe a continué de grandir, et ce, de manière exponentielle. Lethe peut maintenant résoudre des problèmes de CFD-DEM afin de résoudre, en plus de l'écoulement incompressible du fluide, le contact entre des particules solides [44]. Un modèle pour résoudre l'écoulement des fluides non newtoniens est également disponible au sein de Lethe [45]. Des méthodes à conditions immersives telles que *sharp-edge interface* [46] ou Nitsche [47] permettent de résoudre la dynamique des fluides de systèmes dont le domaine est extrêmement complexe à mailler. Étant donné que le logiciel a été dûment vérifié et validé, Lethe est un outil de haute qualité pour effectuer des simulations de mélangeurs. À noter que tout le logiciel de Lethe ainsi que sa documentation, contenant plusieurs exemples industriels, se trouve dans un référentiel libre d'accès Github [48].

CHAPITRE 4 ARTICLE I. ARTIFICIAL NEURAL NETWORK TO PREDICT THE POWER NUMBER OF AGITATED TANKS FED BY CFD SIMULATIONS

Auteur.e.s : Valérie Bibeau, Lucka Barbeau, Daria Camilla Boffito et Bruno Blais

Date de soumission : Le 13 octobre 2022

Date de parution : Le 6 février 2023

Journal : *The Canadian Journal of Chemical Engineering* (CJCE)

Note : Le journal a suggéré l'article pour paraître sur la couverture d'une édition spéciale qui apparaîtra au mois d'Octobre 2023.

Mots-clés : *computational fluid dynamics, mixing, pre-processing methods, artificial neural networks*

4.1 Abstract

The power consumption of the agitator is a critical variable to consider in the design of a mixing system. It is generally evaluated through a dimensionless number known as the power number N_p . Multiple empirical equations exist to calculate the power number based on the Reynolds number Re and dimensionless geometrical variables that characterize the tank, the impeller and the height of the fluid. However, correlations perform poorly outside of the conditions in which they were established. We create a rich database of 100k computational fluid dynamics (CFD) simulations. We simulate paddle and pitched blade turbines in unbaffled tanks from Re 1 to 100 and use an Artificial Neural Network (ANN) to create a robust and accurate predictor of the power number. We perform a mesh sensitivity analysis to verify the precision of the N_p values given by the CFD simulations. To sample the 100k mixers by their geometrical and physical properties, we use the Latin Hypercube Sampling (LHS) method. We then normalize the data with a MinMax transformation to put all features in the same scale and thus avoid bias during the ANN's training. Using a grid search cross-validation, we find the best architecture of the ANN that prevents overfitting and underfitting. Finally, we quantify the performance of the ANN by extracting 30% of the database, predicting the N_p using the ANN and evaluating the mean absolute percentage error. The mean absolute error in the ANN prediction is 0.5% and its accuracy surpasses correlations even for untrained geometries.

4.2 Introduction

Mixing is one of the most important unit operations in chemical engineering. Applications such as homogenization, emulsification and aeration for fermentation processes require agitation. To design mixing systems, important quantities need to be considered such as the power consumption of the agitator and the dimensionless mixing time. The choice of the agitator is often motivated by the regime of operation (laminar or turbulent), the kinematic viscosity and the volume of the fluid being mixed. Different kinds of impellers are used for different operating conditions. For example, axial or radial discharge turbines (propellers, Rushton, hydrofoils, pitched blades) are used for fluids with lower viscosity whereas close-clearance impellers (helical ribbons, anchors) are used for viscous fluids [1].

To characterize the power consumption, engineers rely extensively on the dimensionless power number N_p , which relates the power consumption of the agitator to the inertial work in the fluid and characterizes the energy requirement of an agitated system :

$$N_p = \frac{P}{\rho N^3 D^5} \quad (4.1)$$

with ρ fluid's density, N the impeller rotational speed, D the impeller diameter, and P the power consumption [1, 49, 50].

The power number depends on dimensionless geometrical properties that describe a given mixing system as well as the Reynolds number Re :

$$Re = \frac{D^2 N}{\nu} \quad (4.2)$$

with ν the kinematic viscosity [1, 49, 50].

The power number N_p can be obtained through two approaches : empirical correlations and CFD simulations.

Empirical correlations are power curves where N_p is a logarithm function of Re and geometrical ratios using the impeller diameter D as the reference length. These correlations are generally built using experimental data gathered on mixing systems [51].

$$N_p = f \left(Re, \frac{T}{D}, \frac{H}{D}, \frac{C}{D}, \frac{l}{D}, \frac{w}{D} \right) \quad (4.3)$$

Here, T is the width of the tank, H the height of the fluid inside the tank, C the clearance of the impeller from the bottom, l the length of the blades and w the width of the blades.

Those power curves behaves differently for the laminar regime (at low Reynolds) and for the turbulent regime (at higher Reynolds). In the laminar regime, $N_p \propto Re^{-1}$, meaning that the slope of N_p is constant in logarithmic scale. On the other hand, in the turbulent regime, N_p becomes constant. Usually, the transition between the laminar regime and the turbulent regime occurs around $Re = 200$ [1].

Multiple correlations can be used to estimate the power consumption of paddle and PBT impellers [52–55]. These correlations generally fit on power numbers obtained experimentally.

Computational Fluid Dynamics (CFD) is a robust and accurate tool to simulate mixing systems. With a validated software and an appropriate mesh, velocity field, pressure and torque can be easily computed from CFD simulations [49].

These two approaches present some limitations. On one hand, the correlations do not generalize to multiple geometries without requiring new experimental data. Indeed, the dependency between the power number N_p and the mixer geometry is so complex and non-linear that an algebraic function cannot adequately represent all possibilities. Furthermore, only few variations of geometrical characteristics are investigated in the experiments, which leads to undercoverage biases and thus inaccuracies in the calculation of the power number N_p .

On the other hand, if we use CFD simulations to predict the power consumption of a mixing system, a mesh sensitivity analysis and multiple simulations are needed to establish a power curve. This requires significant time, computational resources and expertise in meshing and CFD.

Artificial Neural Networks (ANNs) are statistical models that can predict continuous values through a regression technique. By propagating signals through multiple interconnected nodes that form a deep network, ANNs take into account non-linear relationships between inputs and outputs, like empirical correlations [25]. ANNs have been shown to be able to model accurately multiple chemical engineering problems such as internal faults detection in heat exchangers, [56] process control using model predictive control (MPC) [57] and also kinetic modeling to optimize fluidized bed reactor performance [58]. In mixing applications, it was demonstrated that ANNs coupled with empirical correlations can predict the power and head characteristics of pump-mixers from experimental data [59]. Nonetheless, ANNs have never been used as an independent model to predict the power number N_p .

To overcome the lack of generality of the traditional correlations and to avoid gathering an unrealistic amount of experimental data, we propose the combination of CFD and machine

learning to estimate the power number of impellers. CFD can simulate mixing systems in large numbers (e.g. 100k configurations) in a relatively small time thanks to high-performance clusters. The machine learning approach gives a simpler predictor of the power number than CFD and a more precise statistical regressor than correlations.

The main objective of this study is to predict power curve from Re 1 to 100 of paddle and pitched blade turbines (PBT) mixers given their geometrical characteristics and operating conditions. First, we present the CFD methods and settings to predict the power number for a mixing configuration. We then describe the sampling method used to generate efficiently a hundred thousand (100k) mixing simulations for the purpose of feeding a rich and large database to the ANN. Using a grid search with cross-validation, we construct the ANN's architecture by finding its optimized hyperparameters. Finally, we test and analyse the performance of the ANN's training by evaluating the loss function and by comparing the ANN's results to an empirical correlation.

Using an effective sampling method of the CFD simulations and performing a data pre-processing, the ANN is able to predict the power number with an absolute error of 0.5%. It significantly outperforms the correlation because it is able to generalize its predictions of the power number to a larger breadth of geometrical configurations.

4.3 Models and methods

4.3.1 Database generation

This section discusses the strategy employed to generate the mixer database. We describe the dimensionless parametrization of the geometry, the meshing procedure, the CFD solver and the sampling strategy.

Domain parametrization

The geometry of the mixer is defined by seven different parameters, some of which are shown in Figure 4.1. The tank is unbaffled, has a diameter of T , and the fluid inside the tank has a height of H . The impeller has a diameter D . The blades of the impeller have a width of W , a thickness of E , and can be tilted with an angle θ . Finally, the impeller has a clearance C from the bottom of the tank. The geometry is scaled using common ratios seen in the literature to generate a dimension-independent problem : T/D , H/T , T/C , D/W and E/W [49,51,60–62]. The reference dimension is set to $T = 1$.

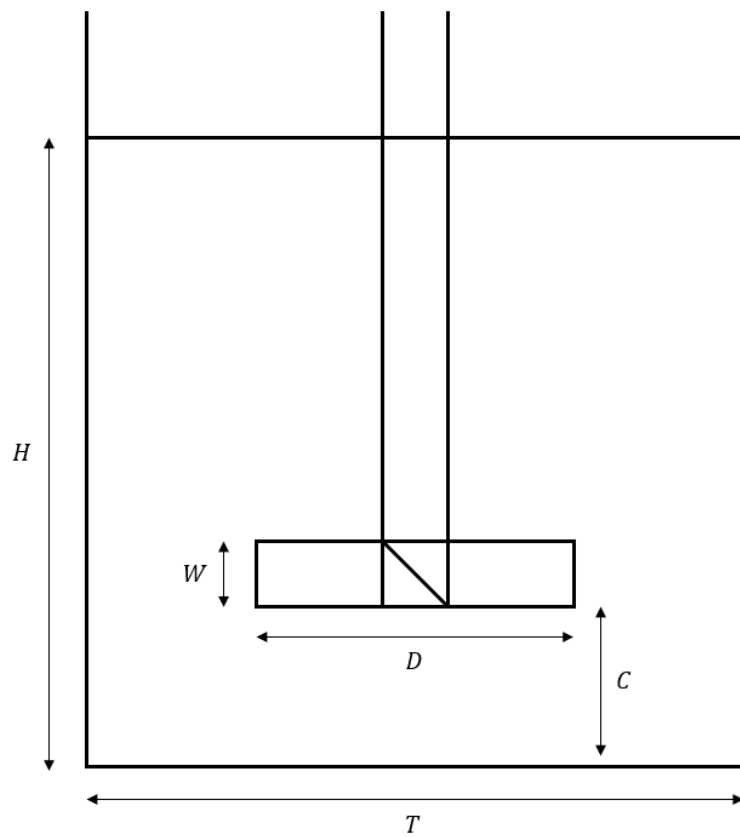


FIGURE 4.1 Scheme of the mixing rig used for the simulations.

CFD simulations

The open-source software Gmsh is used to generate a finite element mesh of tetrahedral elements [63]. The open-source CFD software Lethe is used to solve the incompressible Navier-Stokes equations using a continuous Galerkin SUPG-PSPG formulation [43]. We use P1-P1 interpolation for both velocity and pressure. The mixer is simulated in steady state regime at a low Reynolds number ($Re \in [1, 100]$) where a stable stationary solution is reached. A single rotating frame (SRF) technique is used, meaning that the rotational velocity is imposed on the lateral and bottom walls while the impeller has a zero velocity condition [64]. The fluid interface at the top has a slip condition to mimic the liquid's free surface. Note that this method is limited to vessels of a single shaft with no baffle, because the static part of the tank needs to stay invariant by rotation [49].

Because of the SRF, the Coriolis and centrifugal forces are added to the Navier-Stokes equations as follows [49, 50] :

$$\nabla \cdot \mathbf{u} = 0 \quad (4.4)$$

$$\nabla \cdot \rho \mathbf{u} \otimes \mathbf{u} + 2\rho \boldsymbol{\omega} \times \mathbf{u} + \rho \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \quad (4.5)$$

where $\mathbf{u}$ is the velocity, ρ the density, $\boldsymbol{\omega}$ the angular velocity of the SRF, r the distance to the axis rotation, p the pressure and $\boldsymbol{\tau}$ is the deviatoric stress tensor [43, 49, 50] :

$$\boldsymbol{\tau} = \nu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \quad (4.6)$$

with ν the kinematic viscosity of the fluid.

The rotational speed of the reference frame N is set to one rotation per second (or $\|\boldsymbol{\omega}\| = 2\pi \text{ rad} \cdot \text{s}^{-1}$) and the kinematic viscosity is changed in the simulation parameters to satisfy a specified Reynolds number Re described by Equation 4.2.

The torque is obtained by integrating the stress tensor on the impeller. The power consumption P is a function of the torque Γ and the rotational speed N [1, 49, 50].

$$P = 2\pi \Gamma N \quad (4.7)$$

The power number N_p is obtained from the power consumption P following Equation 4.1.

Sampling strategy

To generate 100k samples that are standardized in all geometrical ratios and Re , Latin Hypercube Sampling (LHS) is used. LHS is a sampling method that stratifies the features to represent uniformly the inputs space [65]. The open-source library pyDOE is used to generate the 100k combinations of the input variables. Table 4.1 lists the interval of the dimensionless geometrical ratios and the Reynolds numbers that are used to sample the mixers.

The 100k simulations are launched on the Narval computer cluster owned by the Digital Research Alliance of Canada. Each simulation is launched on a single core of one CPU and takes approximately 40 minutes to reach convergence at a relative tolerance of 10^{-8} . A script is launched to gather the torque output of all the simulations and the power numbers N_p are stored in a data file.

4.3.2 Artificial Neural Network

The architecture of the ANN is a deep structure composed of multiple layers of neurons. Figure 4.2 represents an example of a deep neural network with one input layer of seven neurons, two hidden layers and an output layer of the variable of interest N_p . The input layer regroups all the variables that characterize agitation : six geometrical parameters (ratios) and one physical parameter (Re). Giving all dimensionless parameters as inputs to the ANN allows more stability to the training rather than giving variables with unit of measurement.

When training an ANN, the goal is to find the right weights between the interconnected nodes from a layer $[i - 1]$ to another layer $[i]$ and construct the best hypothesis function that links the inputs to the outputs. Each set of nodes $\mathbf{z}^{[i]}$ is a linear combination of their

Table 4.1 – Input variables and their interval.

Inputs	Minimum	Maximum
T/D	2	4
H/T	1	1.5
T/C	2	5
D/W	3	6
E/W	0.1	0.2
θ	0	$\pi/3$
Re	1	100

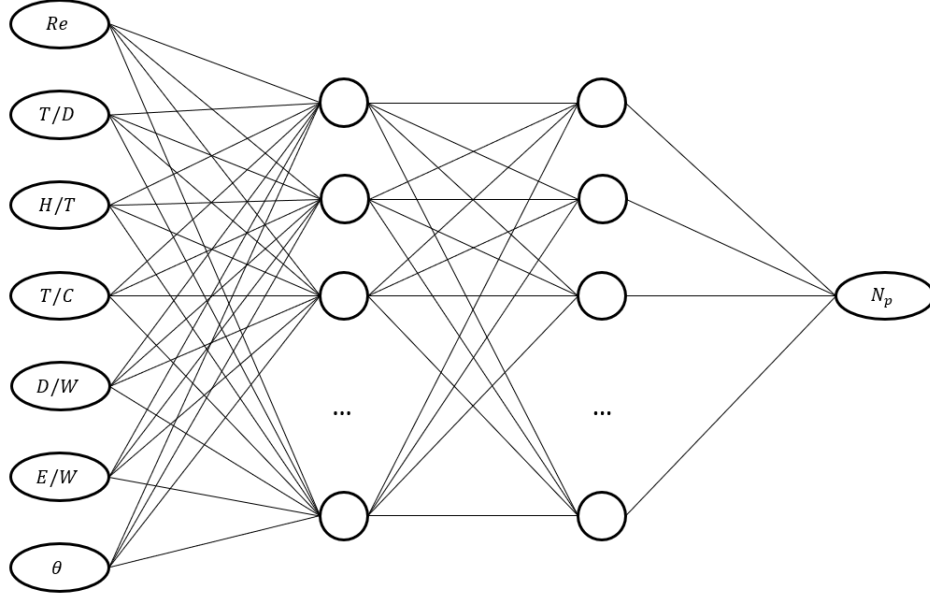


FIGURE 4.2 Artificial Neural Network (ANN) architecture with two hidden layers to predict the power number based on features that describes the mixer.

input neurons $\mathbf{a}^{[i-1]}$ and their respective weights $W^{[i]}$. A bias $\mathbf{b}^{[i]}$ is added as a constant to the linear function [66].

$$\mathbf{z}^{[i]} = W^{[i]} \mathbf{a}^{[i-1]} + \mathbf{b}^{[i]} \quad (4.8)$$

The number of layers $[i]$ and the number of neurons in one layer (size of $\mathbf{a}^{[i]}$) are hyperparameters that affect the performance of the ANN. Another hyperparameter to study is the non-linear activation function σ that transforms a set of nodes $\mathbf{z}^{[i]}$ into neurons $\mathbf{a}^{[i]}$.

$$\mathbf{a}^{[i]} = \sigma(\mathbf{z}^{[i]}) \quad (4.9)$$

Some activation functions, such as the rectified linear unit function (ReLU) and the hyperbolic tangent function (tanh), are more commonly used for deep networks and will be further investigated in this study. The activation function for the output layer should differ from the hidden layers. Given that the problem to be solved is a regression, the output neuron is activated using a linear function [25].

The quantification of the ANN's performance needs to be consistent with the formulation of the problem. As the power number is a continuous variable, the loss function L computed at the output layer that verifies the ANN's performance is the squared error between the predicted output $\hat{y}^{(i)}$ and the known value $y^{(i)}$. In this study, $y^{(i)}$ is the power number (N_p) calculated using CFD simulations.

$$L(\hat{y}^{(i)}, y^{(i)}) = (\hat{y}^{(i)} - y^{(i)})^2 \quad (4.10)$$

The cost function J is the mean squared error (MSE) over all the outputs.

$$J = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2 \quad (4.11)$$

The optimization problem of an ANN is to minimize the cost function J , which means that the derivative of the cost function with respect to the weights needs to converge close to zero. Backpropagation allows the evaluation of the derivatives automatically. The algorithm that performs the optimization is the gradient descent technique. It updates the weights according to a learning rate α and a direction given by the derivatives [25].

$$W^{[i]} := W^{[i]} - \alpha \frac{\partial J}{\partial W^{[i]}} \quad (4.12)$$

When the algorithm completes one pass of all the dataset over the neural network, one epoch (iteration) is considered finished. The number of epochs is also another important hyperparameters to consider.

The learning may be slow and the ANN may not converge to a global minimum. Some strategies are implemented to optimize the gradient descent algorithm. First, the Adam optimizer uses a momentum optimization by keeping track of past gradients to accelerate the convergence [25]. Second, the weights W are initialized using the Glorot uniform distribution. This technique randomly initializes the connection weights depending on the number of neurons in each layer. The Glorot distribution has been proven to improve the flow of the signal through the layers in the forward and backward directions [26].

Pre-processing

Before training the ANN, pre-treatment of the input data is important to avoid biases. Normalization of data consists in bringing all the features on the same interval. Indeed, the Reynolds number does not have the same order of magnitude as the geometrical ratios. The update of the weights of the ANN in the backward pass can be affected by the different scales of the features. Training performed on unnormalized mixers data tended to converge to high MSE, which indicates poor performance.

The normalization method used for the purpose of inputs pre-treatment is the MinMax scaling. Each feature x_i is normalized into x'_i using the minimum x_{min} and the maximum x_{max} of the given feature.

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (4.13)$$

When MinMax scaling is applied to the data, each variables is exactly scaled between 0 and 1 and keeps the covariance structure of the original data [67]. A principal component analysis (PCA) on the normalized data shows that there are no dominant directions where the variance of the features is higher. Therefore, all features will be used in the training of the ANN, with no dimensionality reduction being employed.

Training

The architecture of the ANN is set with an exhaustive grid search with cross-validation (CV) that consists of selecting the most suitable hyperparameters. The normalized data are divided into three sets : training, validation and testing. 70% of the data is used for the training and 30% is used for testing the performance of the ANN. To validate the ANN, cross-validation is used to monitor the training in real time and to verify if the model is over-fitted [25]. The grid search allows to train different architectures by changing the number of hidden layers and neurons. The best ANN structure is chosen based on the lowest validation MSE averaged over multiple folds of the data. The cost function of the validation over each epochs is also checked. If the cost function increases, the model is more biased on the training set and fails to generalize. Finally, when the best architecture is found, all the training data is used to fit the model with the right number of epochs that prevents overfitting. Predictions are then made with the testing set to verify the performance of the ANN. The grid search CV is implemented using the Sklearn library in Python [68]. All the implementation and the

training of the ANN is done with the Tensorflow and Keras libraries in Python [38]. All the code programs, scripts and data files of the ANN's implementation are available on a public repository [41].

4.4 Mesh sensitivity

In CFD, the mesh impacts the quality of the results and the computation time. The mesh around the impeller needs to be sufficiently fine to have an adequate prediction of the flow and thus to adequately predict the torque. Using Gmsh, we locally refine the mesh near the agitator. The characteristic length L_c (the representative size of a tetrahedral element), is ten times smaller near the blades and the shaft than for near the walls. The value of L_c is 0.003 near the agitator and 0.03 near the walls. Figure 4.3 shows the velocity in the radial direction while Figure 4.4 shows the velocity in the axial direction of the same vessel with $T/D = 3$, $H/T = 1$, $T/C = 3$, $D/W = 4$, $E/W = 0.1$, $\theta = \pi/4$ and $Re = 100$. Both figures are in an Eulerian reference and use a refined mesh as illustrated in Figure 4.5.

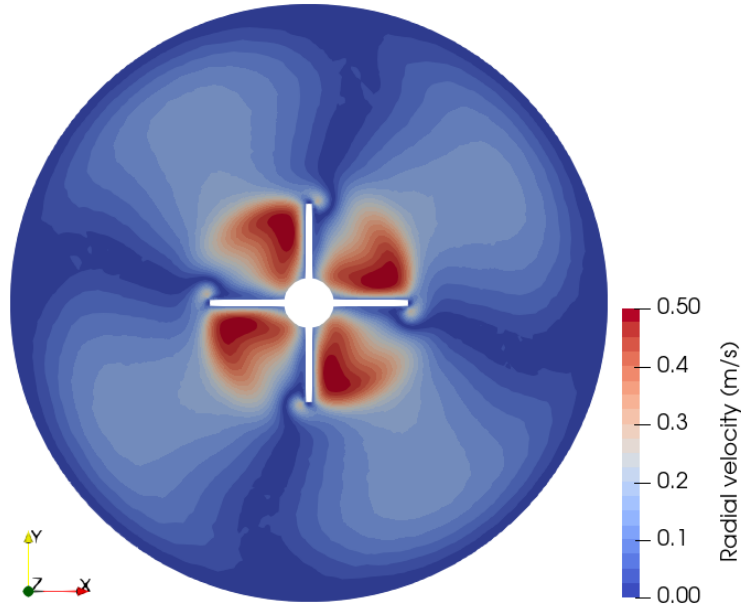


FIGURE 4.3 Velocity in the radial direction in a mixer at $Re = 100$ in a top view perspective.

To evaluate the error on the torque caused by the CFD simulations, we need to do a mesh sensitivity analysis. We gradually refine the global mesh by reducing the characteristic length of the elements and evaluate the torque on the impeller. We do a mesh sensitivity using both an average geometry (medium impeller in a medium tank) at low Reynolds ($Re = 5$), defined

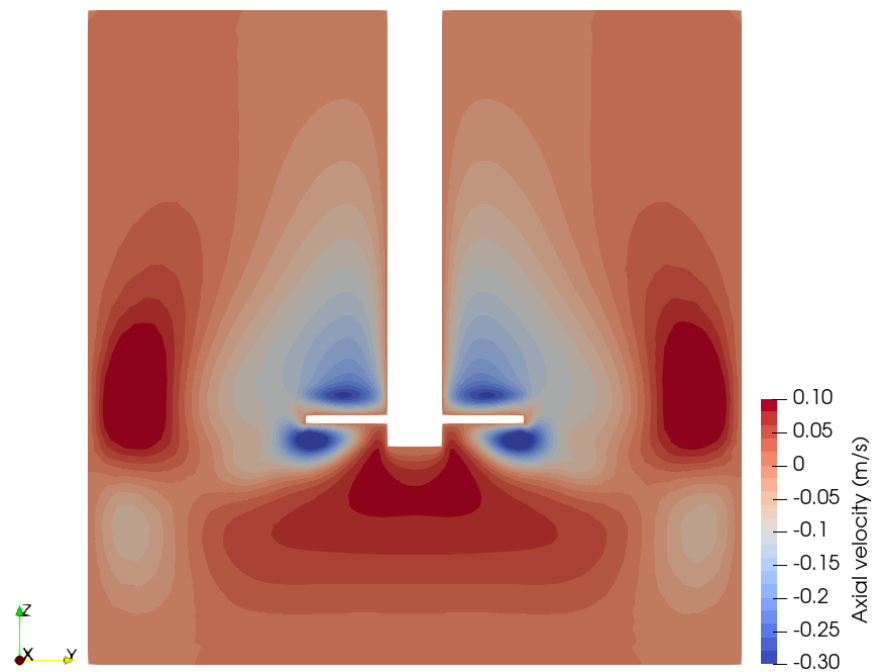


FIGURE 4.4 Velocity in the axial direction at $Re = 100$ in a side view perspective.

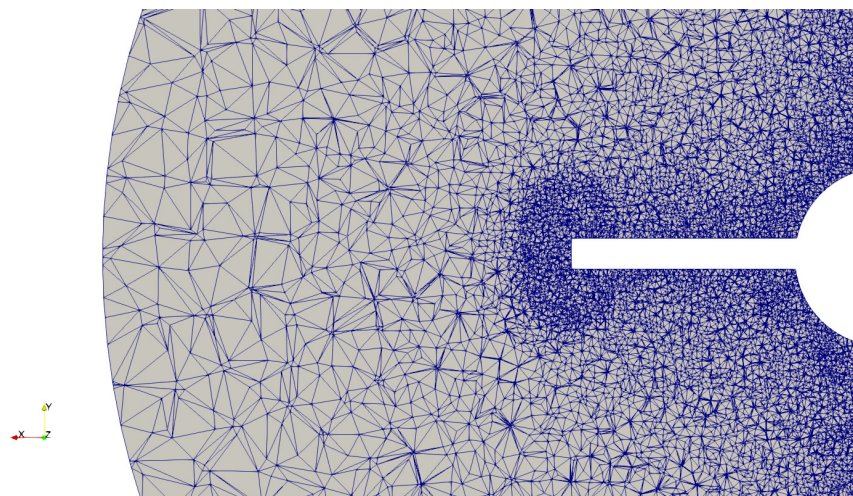


FIGURE 4.5 Mesh refinement near the blades and the shaft of the impeller.

as the average case, and a more complex geometry (smaller impeller in a larger tank) at higher Reynolds ($Re = 100$), defined as the worst case.

Figure 4.6 shows the torque obtained with a given characteristic length normalized to the torque calculated with the mesh used in the 100k CFD simulations ($L_c = 0.03$). We observe that the torque is still mesh-dependent, even with the finer mesh which contains almost 100M elements. We use a generalized Richardson extrapolation to approximate the reference value of the torque and then to calculate the error caused by the domain discretization [69].

$$\hat{f} = f_h + \frac{f_h - f_{rh}}{r^p - 1} \quad (4.14)$$

Here, $\hat{f}$ is the extrapolated value, or the approximation of the solution of the torque. f_h and f_{rh} are the torque evaluated on elements of different characteristic lengths h . r is the refinement factor between the meshes and p is the order of convergence of the torque, which is 1.

The extrapolated normalized torque is 1.07 for the average mixer and 1.14 for the worse mixer. With these values, it is possible to have an estimation of the relative error of the torque e_r .

$$e_r = \frac{|\hat{f} - f_{0.03}|}{\hat{f}} \quad (4.15)$$

Relative errors of 7% for the average mixer and 12% for the worse mixer are obtained with a mesh of a maximum characteristic length of 0.03. These errors are not significant, as they are equivalent to an uncertainty on the impeller speed of about 2%.

Furthermore, the run time of a simulation using a mesh of 3 million elements ($L_c = 0.03$) is 40 minutes on a single CPU while a mesh of 5 million elements ($L_c = 0.025$) takes 80 minutes to simulate, which is two times longer. Simulations with close to 100M cells required a full cluster node in order to have sufficient memory.

Using a coarser mesh is a good compromise between computational time and accuracy. Using this characteristic length, the cost a single simulation was approximatively $1.2 \cdot 10^{-4}$ core-year. In total the simulations carried out in this work consumed 12 core-years.

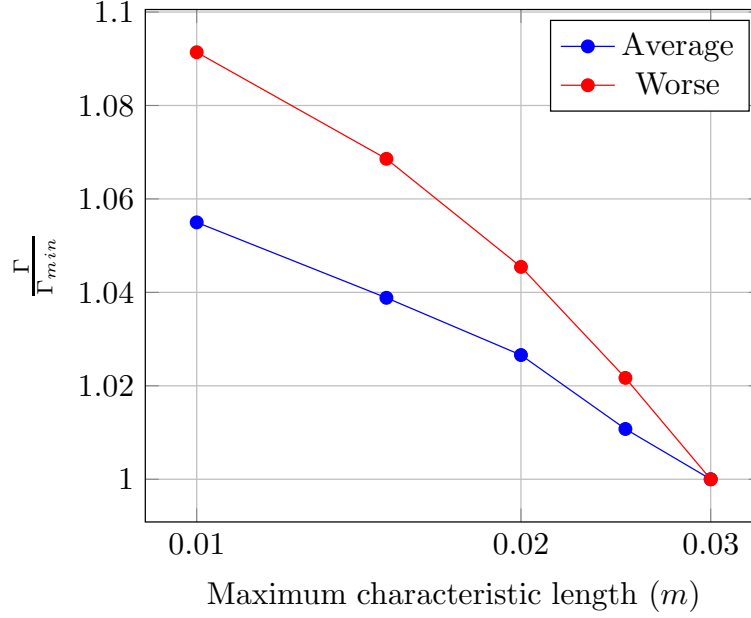


FIGURE 4.6 Normalized torque Γ according to the maximum characteristic length of the mesh near the agitator for average and worse geometries.

4.5 Results and discussion

The grid search CV finds the right balance between the complexity of the architecture (number of parameters) and the cycles of training (amount of times the parameters are updated). An ANN with 3 layers and 50 neurons in each layer is the architecture that gives the lowest averaged MSE over all the cross-validation folds. In this case, the tanh activation function converges more rapidly than the ReLU function. A smaller batch size of 200 updates more frequently the weights during one epoch and stabilizes the variation of the MSE during training.

After cross-validation, the ANN was trained for 5,000 epochs with the optimal architecture on all the training set. Table 4.2 shows the MSE, the mean absolute error (MAE) and the mean absolute percentage error (MAPE) of the training and testing sets with the raw data. The scale of each performance metric is the same for the training and testing, meaning that there is no underfitting nor overfitting of the data. The MAPE of the testing set, which was never seen by the ANN before, is lower than 1%.

Figure 4.7 shows the predictions of the ANN and the estimations of a correlation developed by Hiraoka et al. for paddle impellers [54, 70] using a portion of the testing set. The correlation is detailed in Table 4.3. It uses a friction factor f , a laminar term C_L , a turbulent term C_t , a term that characterizes the transition from laminar to turbulent flow C_{tr} , an exponent term

Table 4.2 – Performance metrics on each set.

Set	MSE (10^{-4})	MAE (10^{-2})	MAPE (%)
Training	6.10	1.84	0.538
Testing	7.60	1.88	0.540

m and other coefficients. It also uses a modified Reynolds number Re_G . Note that n_p is the number of blades, which is 4 in this case.

According to Figure 4.7, the ANN’s predictions are really close to the true power number values and the errors are normally distributed, with no apparent outliers. These results show that the ANN performs well. Normalization of the data plays a major role in obtaining low error metrics. The correlation of Hiraoka et al. exceeds mostly the 25% error threshold illustrated in Figure 4.7, showing its poor performance. The correlation severely underpredicts and overpredicts N_p .

To further validate the accuracy and the robustness of the ANN, we consider impeller configurations which were not included in the database and compare the predictions that are made with the ANN with new CFD simulations and the correlation developed by Hiraoka et al. over the entire range of Reynolds number considered. The results of the comparison is shown in Figures 4.8 and 4.9.

In Figures 4.8 and 4.9, predictions agrees almost perfectly with the Lethe simulations of the same mixer geometry, proving that the ANN performance is related to the pre-processing of the 100k runs (LHS sampling, coherent CFD tolerance, proportional refinement of the mesh, etc.).

According to Figure 4.9, even when compared to results obtained with a finer mesh (smaller element size L_c), the ANN predictions are extremely close to the CFD simulations. This shows that the precision of the torque due to the mesh resolution does not affect the performance of the ANN. Having a discretization error of 7% on the torque is more than acceptable and allows the ANN to obtain significantly better results than the correlation of Hiraoka et al.

Indeed, the correlations do not fit the N_p values predicted by the ANN or calculated by Lethe. In Figure 4.8, the correlation deviates close to $Re = 100$. It is well known that in this area of the power curve, the transition between the laminar flow regime and the turbulent flow regime is difficult to predict [71, 72]. While in Figure 4.9, the correlation completely deviates from the CFD simulations and the ANN predictions. Because correlations reproduce only few experimental values, they are ill-posed to estimate the power number for higher Reynolds, where the transient regime occurs, and for other specific geometries. Lethe has been verified and validated for mixing problems, thus it is able to simulate the flow of complex systems

Table 4.3 – Correlation equations for the power number for pitched and paddle impellers in unbaffled condition.

$$\begin{aligned}
 N_p &= ([1.2\pi^4\beta^2] / [8D^3 / (T^2H)]) f \\
 f &= C_L / Re_G + C_t \left([(C_{tr} / Re_G) + Re_G]^{-1} + (f_{\sim} / C_t)^{1/m} \right)^m \\
 Re_G &= ([\pi\eta \ln(T/D)] / (4D/\beta T)) Re \\
 C_L &= 0.215\eta n_p (D/H) \left[1 - (D/T)^2 \right] + 1.83 (W \sin \theta / H) (n_p / 2 \sin \theta)^{1/3} \\
 C_t &= \left[(1.96X^{1.19})^{-7.8} + (0.25)^{-7.8} \right]^{-1/7.8} \\
 m &= \left[(0.71X^{0.373})^{-7.8} + (0.333)^{-7.8} \right]^{-1/7.8} \\
 C_{tr} &= 23.8 (D/T)^{-3.24} (W \sin \theta / T)^{-1.18} X^{-0.74} \\
 f_{\sim} &= 0.0151 (D/T) C_t^{0.308} \\
 X &= \gamma n_p^{0.7} W \sin^{1.6} \theta / H \\
 \beta &= 2 \ln(T/D) / [(T/D) - (D/T)] \\
 \gamma &= \left[\eta \ln(T/D) / (\beta T/D)^5 \right]^{1/3} \\
 \eta &= 0.711 \left(0.157 + [n_p \ln(T/D)]^{0.611} \right) / \left(n_p^{0.52} [1 - (D/T)^2] \right)
 \end{aligned}$$

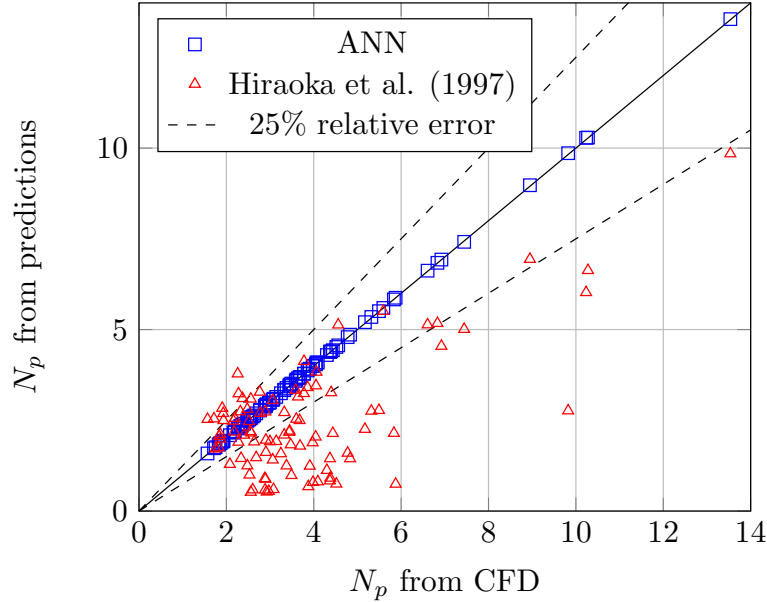


FIGURE 4.7 Predictions using the ANN and the correlation from Hiraoka et al. (1997) versus values of the power number N_p from the CFD simulations.

and accurately predict the power number.

To prove the validation furthermore, we compared the N_p values predicted by the ANN with experimental data [49]. Figure 4.10 shows that the ANN's predictions are within the error intervals of the experiments. The correlation deviates more from the experimental values than the ANN's predictions and it often lies outside of the error bounds, especially at higher Reynolds numbers.

In general, the ANN greatly surpasses the correlation in terms of fitting the CFD data and, consequently, is in much better agreement with the experimental data.

4.6 Conclusion

The determination of power consumption is a key step in designing a mixing system. The power number N_p represents the energy requirement of the agitator depending on its geometry and the Reynolds number Re . The power number can be obtained from correlations. However, these correlations do not generalize when the mixing system at study slightly deviates from those used to build and verify the correlations.

To overcome this lack of generality, we proposed in this work to couple CFD simulations with an ANN. On one hand, CFD simulations provide the database of 100k different PBT mixers that are used as an input of the ANN. On the other hand, the ANN learns non-linear patterns within the database and ultimately estimates the power consumption of a PBT mixing system.

We presented the methodology behind the generation of the 100k mixers, such as the domain parametrization, the CFD settings and the latin hypercube sampling method. We also presented the pre-processing of the data and the grid search cross-validation that allowed to have an optimized architecture and an effective training of the ANN.

Finally, we presented the performance of the ANN by keeping 30% of the data and making predictions of the power number N_p over the testing set using the ANN correlation. The ANN reached a MAPE of 0.5%, which is an incredibly low metric error. The key to get such a low MAPE is in the pre-processing of the data and the monitoring of the training :

- Features were sampled using a LHS method, so that all inputs represents faithfully the dimensions of the problem.
- Mesh generation was standardized and all simulations converged to the same tolerance.
- Normalization of the data using a MinMax scaling helped the ANN to overpass inevitable biases.

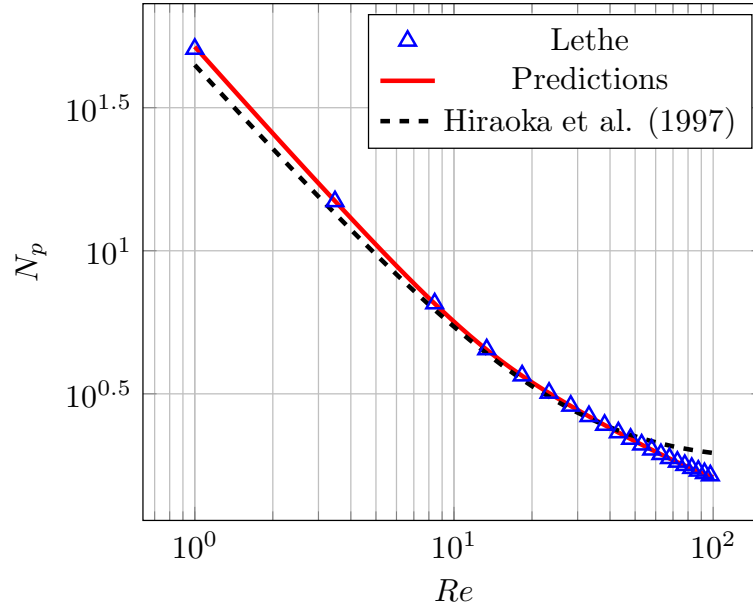


FIGURE 4.8 Power number versus Reynolds number for a specified geometry ($T/D = 3$, $H/T = 1$, $T/C = 4$, $D/W = 5$, $E/W = 0.1$, $\theta = \pi/4$) using the ANN, CFD simulations (Lethe) and Hiraoka et al. (1997) correlation.

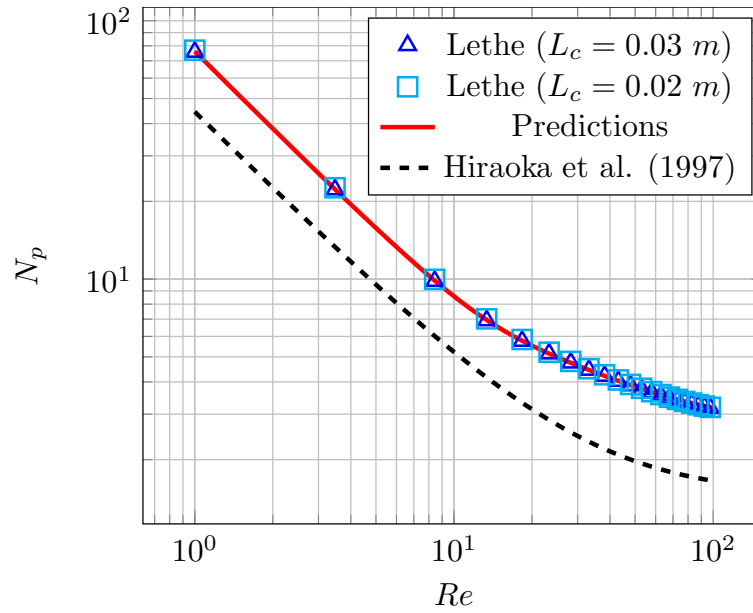


FIGURE 4.9 Power number versus Reynolds number for a specified geometry ($T/D = 2.5$, $H/T = 1.2$, $T/C = 3$, $D/W = 3.5$, $E/W = 0.15$, $\theta = \pi/6$) using the ANN, CFD simulations (Lethe) of two different element sizes L_c and Hiraoka et al. (1997) correlation.

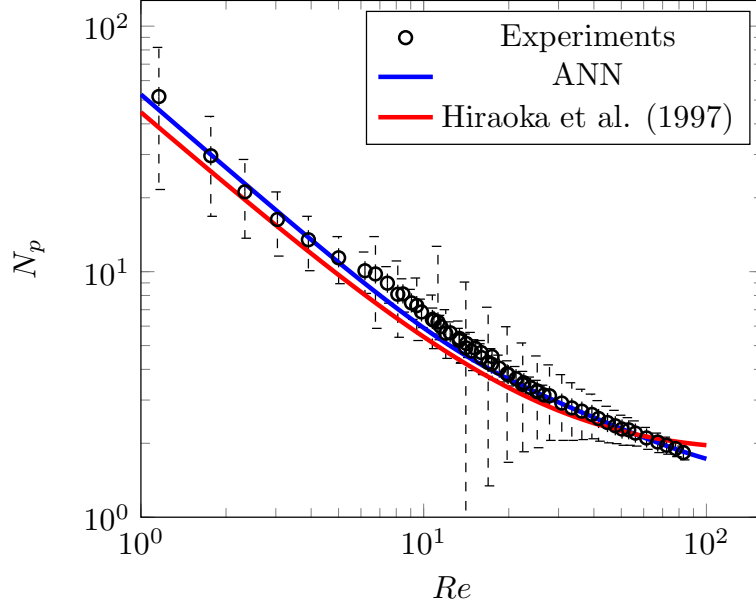


FIGURE 4.10 Validation of the ANN’s prediction of a power curve for a specified geometry ($T/D = 3$, $H/T = 1$, $T/C = 4$, $D/W = 5$, $E/W = 0.15$, $\theta = \pi/4$).

- Grid search CV was essential to validate the architecture and to monitor the training, so that the ANN does not overfit nor underfit the data.

Also, we presented some comparisons between the ANN predictions and the correlations from Hiraoka et al. The results showed that the predictions of the power number N_p made by the ANN outperforms the correlation for all cases considered, even for mixers that has never been trained by the ANN.

With these results, we demonstrated that Artificial Neural Networks, trained with CFD data, accurately predict the power consumption of paddle and pitched blade impellers. This approach can be easily generalized to other types of impellers or to the turbulent regime by enriching the database with new simulations. It could also be extended to predict other quantities such as the mixing time, the pumping number, etc. We think that this work demonstrates that simulation-fed ANN are an ideal candidate to replace traditional correlations in the design of unit operations such as agitated tanks.

4.7 Acknowledgements

The authors would like to acknowledge financial support from the “Institut de valorisation des données” (IVADO) through the Undergraduate introduction to research scholarship. Computations were made on clusters Beluga and Narval managed by Digital Research Alliance of

Canada. In particular, the authors would like to acknowledge the efficient support received from Calcul Québec systems analysts.

CHAPITRE 5 ARTICLE I. DISCUSSION

Ce présent chapitre a pour but de revenir sur certains éléments clés à retenir de l'article I ou d'approfondir certains concepts qui ont été effleurés dans celui-ci. Pour ce faire, la discussion revoit trois grands thèmes décrivant la méthodologie sur laquelle se base le développement du réseau de neurones, soit la génération des données numériques à partir de la CFD, la définition de l'architecture du réseau par validation croisée et finalement, l'influence du nombre d'itérations et de la taille du lot de données sur l'entraînement.

5.1 Génération des données numériques

La raison pour laquelle le réseau de neurones proposé dans l'article I fait des prédictions du nombre de puissance avec des métriques d'erreur si basses (une MAPE en deçà de 1 %), provient de la qualité des données de CFD.

Tout d'abord, la méthode d'échantillonnage LHS a eu une influence majeure sur les prédictions du réseau de neurones. D'autres méthodes d'échantillonnage existent, mais elles n'ont pas réussi à obtenir des prédictions avec une aussi faible erreur. Par exemple, un échantillonnage des attributs sur une grille a été tenté. Cette méthode espace les attributs linéairement entre le minimum et le maximum de leur intervalle et les combine ensemble un à un, ce qui forme une grille. Avec cette méthode, les prédictions à haut Reynolds étaient erronées, ce qui réduisait fortement la capacité du réseau de neurones à extrapoler le nombre de puissance. La raison derrière cette observation proviendrait de la dépendance exponentielle du nombre de puissance au nombre de Reynolds. Ainsi, une grille linéaire pourrait facilement échouer à représenter cette corrélation hautement non linéaire. De plus, comme la revue littéraire le démontre, à un certain nombre de Reynolds, le régime du fluide est en transition, passant de laminaire à turbulent. Cette transition dépend grandement de la géométrie de l'agitateur et est difficile à prédire. Pour cette raison, une simple grille ne permettrait pas d'aller vérifier certaines relations plus complexes entre les attributs. La méthode LHS, quant à elle, couple les avantages d'une grille, permettant de représenter le mieux possible l'espace des attributs, et celles d'un échantillonnage aléatoire, permettant d'élargir la variabilité des combinaisons d'attributs. Pour illustrer davantage l'efficacité de la méthode LHS, le réseau de neurones a été entraîné avec une base de données nettement moins volumineuse, contenant seulement 10 000 enregistrements au lieu de 100 000. Une MAPE de 0.6 % a été obtenue, ce qui est une erreur très basse considérant la réduction de la base de données par un facteur 10. Une MAPE de 0.5 % est obtenue avec l'entièreté de la base de données. Ce résultat démontre les

bienfaits de la méthode LHS sur les prédictions faites par le réseau.

Ensuite, le maillage du domaine de calcul, permettant la discrétisation du fluide entourant l'agitateur, joue un rôle important dans la qualité des prédictions du réseau de neurones. Des tétraèdres sont utilisés comme éléments plutôt que des hexaèdres. L'utilisation de tels simplexes permet de générer des maillages plus réguliers, ce qui est avantageux pour des géométries complexes comme les mélangeurs. De plus, comme expliqué dans l'article I, le raffinement autour de l'agitateur est fait de manière proportionnelle à sa géométrie et, pour toutes les simulations, le résidu de la forme faible des équations de Navier-Stokes a convergé en deçà d'une tolérance similaire, soit de 1×10^{-8} . Somme toute, le temps investi à la méthodologie des simulations de CFD a permis d'avoir des données numériques du nombre de puissance consistantes.

5.2 Validation croisée

L'architecture employée pour le réseau de neurones a également eu un impact considérable sur la performance du réseau de neurones. Premièrement, la validation croisée permet de régler les hyperparamètres du réseau, c'est-à-dire les paramètres qui décrivent son architecture. La validation croisée a été faite sur une grille d'hyperparamètres qui teste différentes combinaisons du nombre de neurones ($\in \{40, 50, 60\}$) et du nombre de couches ($\in \{2, 3, 4\}$). Comme indiqué dans l'article, la meilleure architecture, qui donne une erreur moindre sur les prédictions, était un réseau de 3 couches cachées contenant chacune 50 neurones. En utilisant cette architecture, la MSE sur les 3 sous-parties d'entraînement de la base de données possède une moyenne de 8.6×10^{-8} avec un écart-type de 2.4×10^{-8} . Selon ses résultats, il est possible de dire qu'un réseau peu profond n'est pas apte à exploiter la richesse du réseau pour prédire des nombres de puissance cohérents alors qu'un réseau ayant plusieurs neurones est trop difficile à entraîner de par sa complexité élevée. De plus, la validation croisée permet de vérifier si l'entraînement est sensible aux données d'entraînement.

5.3 Effet de la taille des données alimentées au réseau

Le compromis entre le nombre d'itérations et la taille du lot de données à alimenter au réseau pour la mise à jour des poids (ou plus communément appelé le *batch size*) était un autre élément clé de la méthodologie. Durant l'entraînement, un lot plus grand fait en sorte qu'une itération se calcule plus rapidement. Toutefois, malgré le fait qu'un grand lot procure une stabilité à la diminution de la fonction de perte, un grand nombre d'itérations est nécessaire afin de minimiser l'erreur sur les prédictions. Dans le cas contraire, un lot plus

petit exige un coût de calcul plus élevé pour effectuer une itération. Cependant, étant donné l'effet erratique d'un petit lot, un autre minimum peut être rapidement découvert et mener à des prédictions ayant une erreur moindre. La Figure 5.1 illustre la diminution de la fonction de perte selon le lot employé. En employant un petit lot, on remarque que la fonction de perte converge beaucoup plus rapidement. De plus, l'effet erratique est présent, mais il demeure moins prononcé que les autres valeurs de lot, et ce, pour un même taux d'apprentissage et une convergence de la fonction de perte vers une valeur similaire. L'analyse du lot est importante et dépend grandement des données du problème. Dans le cas des mélangeurs, leur nombre de puissance est hautement non linéaire avec le nombre de Reynolds et la géométrie de l'agitateur, faisant en sorte qu'un lot plus petit permet une meilleure exploration de la fonction de perte et de converger vers un minimum assez rapidement.

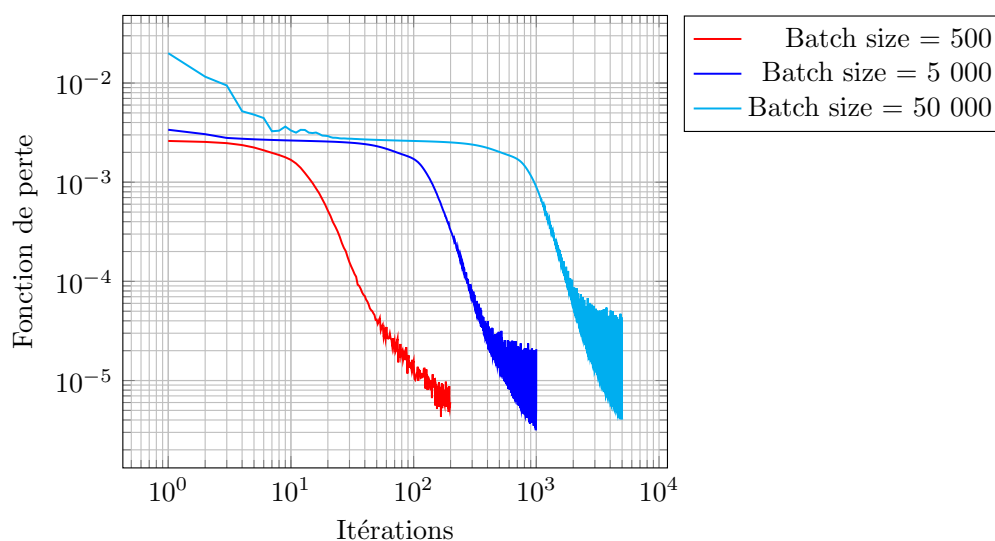


FIGURE 5.1 Analyse du lot sur la base de données du nombre de puissance.

CHAPITRE 6 ARTICLE II. PHYSICS-INFORMED NEURAL NETWORK TO PREDICT KINETICS OF BIODIESEL PRODUCTION IN MICROWAVE REACTORS

Auteur.e.s : Valérie Bibeau, Daria Camilla Boffito et Bruno Blais

Date de soumission : Le 10 juillet 2023

Date de parution : Au moment du dépôt de ce mémoire

Journal : *Chemical Engineering and Processing - Process Intensification* (CEP : PI), dans une revue dédiée à la conférence ISCRE27 qui s'est déroulée à Québec en juin 2023.

Mots-clés : *microwave, reaction kinetics, PINN, digital twin*

6.1 Abstract

Microwaves are a process intensification (PI) method to deliver energy to reactive systems by an alternative energy vector. Microwaves act directly on molecules' dipolar moment, consequently making them spin and generating volumetric heating that causes temperature to rise more quickly, which directly impacts the overall rate of a reaction. The rate of reactions follows non-linear rate laws and therefore are difficult to predict. We use a Physics-informed Neural Network (PINN), a physics-driven model, to identify the reaction kinetics of a biodiesel production process. PINNs perform a regression on very few experimental data points and try to fit the physics at hand. We use a microwave reactor with a constant power output to perform the transesterification reaction, measure the infrared temperature and analyze the concentration of glycerides using GC-FID at different reaction time. We train the PINN to predict the reaction rates with respect to the Arrhenius equation. Results show that the PINN successfully identify the rate constants and the reaction constants, including their temperature dependence. Furthermore, the PINN can extrapolate its predictions to other power outputs without ever seeing the concentration data, making a digital twin of the microwave-assisted reaction.

6.2 Graphical abstract

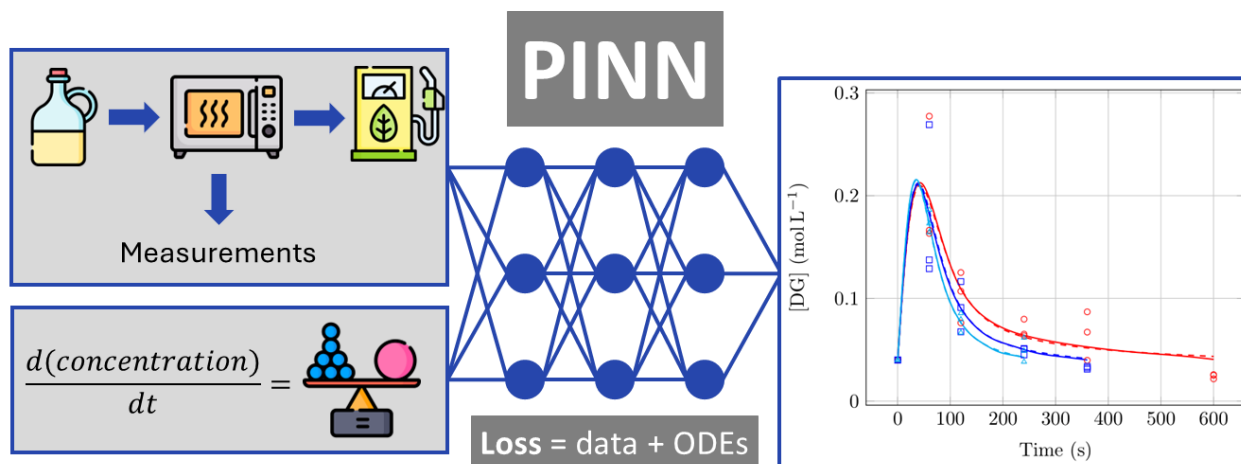


FIGURE 6.1 Graphical abstract of the PINN article.

6.3 Introduction

Engineers mostly design chemical plants based on the "big is best" paradigm to achieve economies of scale [73]. This process design is sequential and operation-orientated, leading to plants with high volume footprint [73,74]. Process intensification (PI) is a paradigm shift that either design new or retrofit existing processes into more compact and more efficient ones [75]. One of PI's approaches is enhancing heat transfer with alternative energy vectors [74]. Microwaves are a form of electromagnetic radiation which generate volumetric heating [76]. They interact directly with the material at the molecular level, more specifically molecules that have a polar moment. Under microwave emission, dipoles forces within molecules try to align the electromagnetic field, making them rotate. This spin results in the dissipation of heat through the internal resistance of rotation [76]. Unlike conductive heating, volumetric heating is more efficient and allows for faster reactions [77,78]. Kinetics modelling measures the contribution of microwaves on reaction rates, which is key to design, optimize and control microwave reactors [79].

Reaction kinetics frequently use rate laws that relate a reaction rate to species concentration (e.g. $[A]$ and $[B]$), stoichiometry and rate constants k :

$$\text{Rate} = k[A]^x[B]^y \quad (6.1)$$

Assuming that reactions are homogeneous, and using a mass balance on the species, we can predict the evolution of concentration over time with a set of ordinary differential equations (ODEs).

Because most reactions are reversible (limited by thermodynamic equilibrium), and include intermediates, closed-form analytical solutions for the set of ODEs can seldom be obtained [80]. Modelling strategies exist to reduce the complexity of reactions and thus the number of rate constants to obtain, such as lumping, quasi-steady state and quasi-equilibrium approximations [81]. On the other hand, a more detailed kinetics approach, such as composition modeling techniques, tries to preserve the specifics of chemicals at the molecular scale [81]. This strategy, although accurate, is difficult to leverage when there are multiple reaction intermediates [82]. Thus, the challenge is to find a compromise between the use of experimental data and the complexity level of the reaction mechanisms considered.

Deep learning is a growing method that is able to find complex patterns between the features of a given problem and its outputs by leveraging an Artificial Neural Network (ANN) [23]. However, it requires a large amount of high-quality labeled data to obtain accurate predictions. Physics-informed Neural Networks (PINNs) combine the advantages of deep learning algorithms and detailed physics models to mitigate this issue [24]. PINNs do not require a large amount of labeled data, reducing the volume of experimental measurements needed [83]. PINNs can evaluate complex transient phenomena such as the stiff kinetics of Burgers’ equation [84] and the point kinetics in the digital twin of a nuclear reactor [85].

In this work, we leverage Physics-informed Neural Networks (PINNs) to identify the reaction kinetics of a multi-step reaction that occurs in a microwave reactor. The reaction we study is the production of biodiesel through transesterification of canola oil. First, we present an overview of PINNs. Then, we present the methodology for extracting the concentration data and the temperature data of the transesterification reaction. We also present the physical model, which is the ODEs and the rate constants that the PINN solves and predicts. We explain in detail the architecture of the PINN that identifies the reaction rates and generalizes to other microwave power values. Finally, we illustrate and discuss the predictions of the PINN and evaluate its capacity to extrapolate to other operating conditions.

6.3.1 Physics-informed Neural Networks

PINNs encode in their training the governing equations that describe the evolution of a physical quantity over time and/or space. Consequently, PINNs approximate partial differential equations (PDEs) or ODEs [83]. To estimate the solution of an ODE specifically, we train a PINN by minimizing a loss function $\mathcal{L}$. The loss combines the mean squared error of the

residual form of the ODE (MSE_f) evaluated at different collocation points in time and the mean squared error of the initial condition and measurements (MSE_u) [24] :

$$\mathcal{L} = \text{MSE}_u + \text{MSE}_f \quad (6.2)$$

where,

$$\text{MSE}_u = \sum_i^{N_u} |u_i - \hat{u}_i|^2 \quad (6.3)$$

$$\text{MSE}_f = \sum_j^{N_f} |f(t_j)|^2 \quad (6.4)$$

with N_u the number of labeled data points, u_i the predicted solution of the ODE, $\hat{u}_i$ the initial condition and the measurements, N_f the number of collocation points, t_j a specific collocation point in time and $f(t_j)$ the evaluation of the residual form of the ODE at the collocation point. The residual form f is :

$$f := u_t + f(u, \lambda) \quad (6.5)$$

with u_t the derivative of the solution over time, λ the parameters (or constants) of the ODE and $f(u, \lambda)$ the right-hand side of the ODE, which could be non-linear with respect to u and λ .

We present a typical PINN's architecture for solving an ODE in Figure 6.2. The PINN predicts u at each temporal collocation point using the neural network. To calculate u_t , we perform automatic differentiation (AD) that calculates the overall derivative by combining known derivatives through the chain rule [86]. Finally, the PINN updates the weights of its network iteratively using the gradient descent algorithm until a targetted loss is reached [23].

PINNs can solve inverse problems where parameters of PDEs or ODEs are unknown. Using gradient descent, PINNs are able to discover the parameters λ from Equation 6.5 during training [24]. For example, a PINN solves a catalytic CO_2 methanation reaction and discover an

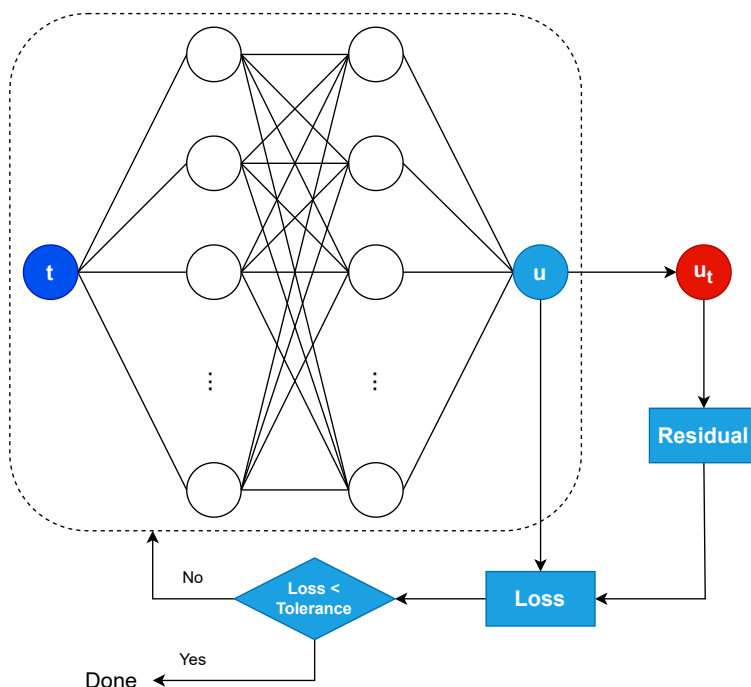


FIGURE 6.2 PINN's architecture.

unknown effectiveness factor involved in the reaction kinetics [87]. In the context of reaction kinetics, a PINN could be used to solve the inverse mass balance problem and identify all the rate constants and their dependence on temperature. If we want to create a digital twin, this PINN should generalize to multiple operating conditions of the microwave reactor. To the best of our knowledge, the PINN has not yet been used to identify the reaction rates of a chemical system and to generalize its predictions to different operating conditions. The present work addresses these issues.

6.4 Methodology

We present the methodology used for the extraction of the data for the discovery of the ODEs describing the reaction rates of the transesterification reaction. First, we get the concentration data at different reaction time from gas chromatography (GC). Second, an infrared (IR) sensor or a ruby ball probe measure the temperature. Finally, we describe the physical model of the reaction, or the ODEs describing the reaction kinetics and the global energy balance.

6.4.1 Concentration and temperature data

We perform the transesterification reaction in an Anton Paar Monowave 400 reactor. The reaction uses canola oil (President’s Choice, available in grocery stores) and methanol (MeOH) with a molar ratio of 1 :6. We choose canola oil for its high content in oleic acid, which produces biodiesel with better cold flow properties, and its low melting point, making the oil fluid at room temperature [88]. The catalyst is sodium hydroxyde (NaOH) and its concentration is 0.8% w/w [89]. We agitate the reagents to avoid any mass transfer limitations and set a constant magnetic agitation of 600 rpm during the reaction [90,91]. We set a constant power microwave output of 4, 5 and 6 Watts until the mixture’s temperature reaches 65 °C, temperature at which the methanol starts to evaporate and the reaction kinetics are optimal [92]. The time it takes to reach 65 °C is approximately 10 min for a power of 4 W, 6 min for 5 W and 4 min for 6 W. We also measure the concentration of the species at multiple times before the temperature of 65 °C is reached. Table 6.1 summarizes the experiments done in the microwave reactor.

Table 6.1 – Power output of the microwave and reaction time of the experiments.

Power output (W)	Reaction time (s)
4	60
4	120
4	240
4	360
4	600
5	60
5	120
5	240
5	360
6	60
6	120
6	240

We extract a sample of the reaction products in order to analyse its components using gas chromatography with flame ionization detection (Scion GC-FID). Following the EN-14105 procedure, we mix the sample with 100 μ L of MSTFA for derivatization and 100 μ L of tricaprins as the internal standard for glycerides analysis [93–95].

With the chromatogram computed by the GC-FID, we evaluate the corrective factor (C_F) of the glycerides, which is the reciprocate of the relative response factor of the detector :

$$C_F = \frac{A_x m_{IS}}{A_{IS} m_x} = \frac{A_x C_{IS} V_{IS}}{A_{IS} m_x} \quad (6.6)$$

with A_x the peak area of the glycerides, A_{IS} the peak area of the internal standard, C_{IS} the concentration of the internal standard (8 mg mL^{-1}), V_{IS} the volume of the internal standard and m_x the mass of the sample [95].

Knowing the total mass m , the molar mass M of the glycerides and the volume of the solution V , it is easy to retrace the molar concentration (in mol L^{-1}) over each reaction time from the corrective factor [95, 96]. The final concentration is :

$$C = C_F \times \frac{mM}{V} \quad (6.7)$$

We produce 3 replicates of the experiments presented in Table 6.1 to evaluate the error of the concentration data and the reproducibility of the methodology.

As for the temperature, the reactor has an infrared (IR) sensor that detects the temperature of the sample inside the vial. The numerical output of the Anton Paar reactor is the temperature in degree Celsius ($^{\circ}\text{C}$) at each second of operation.

6.4.2 Physical model

The transesterification reaction occurs in three consecutive steps where the triglycerides (TG) break down into diglycerides (DG), then into monoglycerides (MG) and finally into glycerol (G). With an excess of methanol (MeOH) and sodium hydroxyde (NaOH) as the catalyst, each step produces methyl esters (ME) or biodiesel. All reactions are reversible :



with k_1 , k_3 and k_5 the rate constants of the direct reactions and k_2 , k_4 and k_6 the rate constants of the inverse reactions [97].

Using rate laws and species balance, we obtain the following set of ODEs :

$$\frac{d[TG]}{dt} = -k_1[TG] + k_2[DG][ME] \quad (6.11)$$

$$\frac{d[DG]}{dt} = -k_2[DG][ME] + k_1[TG] - k_3[DG] + k_4[MG][ME] \quad (6.12)$$

$$\frac{d[MG]}{dt} = -k_4[MG][ME] + k_3[DG] - k_5[MG] + k_6[G][ME] \quad (6.13)$$

$$\frac{d[ME]}{dt} = +k_1[TG] - k_2[DG][ME] + k_3[DG] - k_4[MG][ME] + k_5[MG] - k_6[G][ME] \quad (6.14)$$

$$\frac{d[G]}{dt} = -k_6[G][ME] + k_5[MG] \quad (6.15)$$

$$(6.16)$$

We suppose that all direct reactions are pseudo-first order reactions and we reduce the second-order reactions into first-order. This hypothesis is valid because the methanol is in high excess [97].

Reaction rates are dependant on temperature. They follow the Arrhenius equation :

$$k = A \exp\left(\frac{-E_a}{RT}\right) \quad (6.17)$$

Because the scale of the pre-exponential factor A and the activation energy E_a is usually high, the gradient descent algorithm has difficulty to identify properly these kinetics parameters. To simplify the model, we linearize the Arrhenius equation using a first-order Taylor expansion :

$$k = k' + k''(T - \bar{T}) \quad (6.18)$$

with k' and k'' respectively the Arrhenius equation estimated at the mean temperature $\bar{T}$ of the products during the reaction and the first derivative of the Arrhenius equation evaluated at $\bar{T}$.

Furthermore, for the evolution of the temperature over time, we suppose that it follows a simplified first-degree ODE :

$$\frac{dT}{dt} = \varepsilon P + c_1 T + c_2 \quad (6.19)$$

with P the power of the microwave reactor, ε the efficiency of the microwave, the constants c_1 and c_2 that represents respectively the heat loss of the microwave reactor, that is dependant on the temperature, and the reaction source term.

We use all these previous ODEs as the physical model of the PINN. Having the data and the physics at hand, we present in the next section the final architecture of the PINN.

6.4.3 PINN implementation

Using grid search analysis to optimize the hyperparameters, the final PINN has an input layer of 2 neurons, 3 hidden layers of 64 neurons and an output layer of 6 neurons. We present the architecture of the PINN in Figure 6.3. The input layer includes the time t and the microwave power P . The number of instances, or the length of the features vector, are the number of collocation points discretized in the time domain. The power is a discrete value, either 4, 5 or 6 W. This feature differentiates the temperature evolution and therefore the concentration evolution based on an essential operating condition of the microwave.

As for the outputs, we associate 5 neurons to the species concentrations, which are [TG], [DG], [MG], [G] and [ME]. We associate the last neuron to the temperature. By doing so, the PINN predicts the concentrations and the temperature over time in synchrony.

The loss includes the mean squared errors of the residuals of the ODEs described in the physical model :

$$\text{MSE}_r = \frac{1}{N_r} \sum_i^{N_r} |r(t_i, P_i)|^2 \quad (6.20)$$

$$\text{MSE}_g = \frac{1}{N_g} \sum_i^{N_g} |g(t_i, P_i)|^2 \quad (6.21)$$

with r the residuals of the ODEs describing the concentration evolution of all species over time, g the residual of the ODE describing the temperature evolution over time, N_r and N_g the number of collocation points where we evaluate the ODEs.

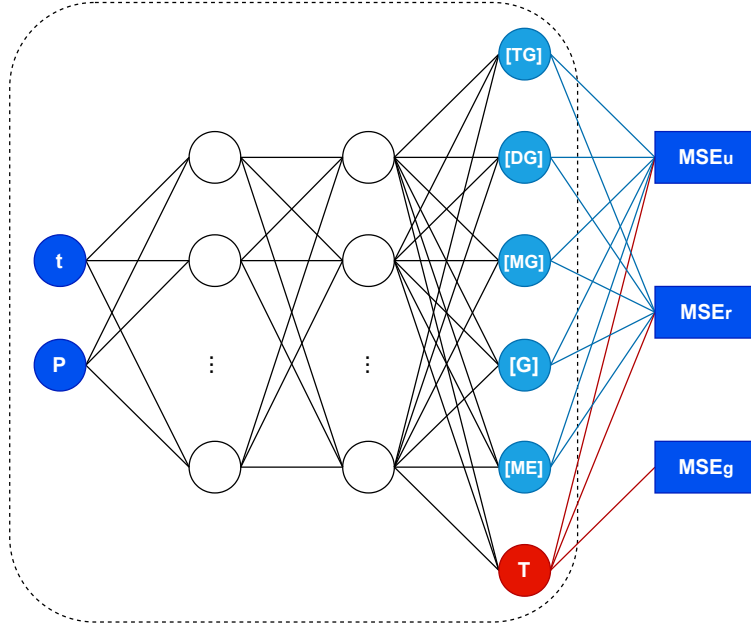


FIGURE 6.3 Reaction kinetics architecture.

Table 6.2 shows the time domain and the number of collocation points (N_r and N_g) for each microwave power. We refine the discretization of the time domain to have a collocation point every half-second in order to better detect the fast transition region of the transesterification reaction [98, 99].

Table 6.2 – Time domain and number of collocation points for each microwave power.

Power output (W)	Time domain (s)	Number of collocation points
4	[0,600]	1201
5	[0,360]	721
6	[0,240]	481

Solving an inverse problem where the physics is incomplete, we need to make the loss landscape more conducive to finding a solution. To do so, we evaluate the mean squared errors of the concentration data and the temperature data :

$$\text{MSE}_u = \frac{1}{N_C} \sum_i^{N_C} |C(t_i, P_i) - \hat{C}_i|^2 + \frac{1}{N_T} \sum_i^{N_T} |T(t_i, P_i) - \hat{T}_i|^2 \quad (6.22)$$

with C the concentration predicted by the PINN for a specific time and power (t_i, P_i) , T the temperature predicted by the PINN for the same instance, $\hat{C}_i$ and $\hat{T}_i$ the true values coming from the experiments, N_C the number of concentration data, N_T the number of temperature

data. N_C includes all replicates of the experiments in Table 6.1 and the initial condition of the canola oil. N_T is given by the sampling frequency of the reactor’s IR sensor, which is 1 per second, and includes the replicates of the full time reaction experiments.

By combining the MSE_u , MSE_r and MSE_g , we obtain the final loss function $\mathcal{L}$ of the PINN :

$$\mathcal{L} = \text{MSE}_u + \eta_r \text{MSE}_r + \eta_g \text{MSE}_g \quad (6.23)$$

with η_r the regularization parameter on the loss of the concentrations ODEs and η_g the regularization parameter on the loss of the temperature ODE.

Regularization plays a key role in PINN. It allows to put more emphasis on penalizing the residuals so the PINN enforces the given physical constraints to satisfy the ODEs [100,101]. First, we train the PINN with a low regularization parameter so that the PINN learns mostly from the data. This alleviates the complexity of the loss landscape (the shape of the loss in all features dimensions) caused by the physics. Gradually, we increase the regularization parameter so that the PINN leads to solutions that satisfy the physics [32].

Finally, we train the PINN for 1M epochs on GPUs from Digital Research Alliance of Canada clusters. We use the tanh function to activate the hidden neurons of the PINN and we use the Adam optimizer [27] for the gradient algorithm.

6.5 Results and Discussion

We present the predictions of the PINN and analyze them in two main fashion. First, we demonstrate the impact of the regularization parameter η_r on the PINN’s predictions. Second, we use all data available to identify as closely as possible the rate constants that dictate the reaction kinetics. Finally, we remove some concentration data and try to predict the concentration evolution over time, being closer to a physics-driven model. By doing so, we try to prove that it is possible to extrapolate the PINN’s predictions to other power outputs, making the PINN a digital twin of the microwave reactor.

6.5.1 Regularization

For regularization, we increase η_r from 5, 50, 500 and finally to 5,000 progressively each 25,000 epochs. The moment we change η_r , the PINN continues its training by minimizing a different loss topology that is more physics-based. To understand and evaluate how the PINN behaves with regularization, we take the rate constants k (or λ) predicted by the PINN every

25,000 epochs and we inject them in a fourth-order Runge-Kutta numerical method for the ordinary differential equations [102].

Figure 6.4 shows the PINN predictions of the TG concentration over time when the regularization η_r is increased during the training. The predictions are compared to the numerical ODEs. We see that for a lower regularization parameter value ($\eta_r = 5$) the PINN fits more the data. Indeed, the rate constants predicted by the PINN solve ODEs that are far from the predictions. This means that, at the beginning of the training, the loss landscape is built in a way that the PINN directs its training towards solutions that are closely aligned with the data. For a higher regularization value ($\eta_r = 5000$), the PINN predicts the TG concentration with satisfaction to the physics, meaning that the rate constants predicted by the PINN fits adequately the governing equations encoded within its training.

6.5.2 Identification of the system

In Figure 6.5, we present the predictions of the temperature evolution over time made by the PINN and we compare them to the microwave data. Note that the temperature data has a resolution of 1 °C and is displayed each second of the reaction, creating discontinuous steps in the temperature monitoring. With the ODE describing the overall energy balance of the reactor, the PINN makes the temperature output smoother.

Figure 6.6 shows the comparison between the normalized concentrations predicted by the PINN on one hand and the experiments on the other. We see that the majority of the PINN's predictions are within the error bars of the experiments. Less accurate predictions are made at the end and the beginning of the reaction, when the concentrations measured by the GC-FID carry a higher inherent uncertainty. Because we place more emphasis on penalization the ODEs using a severe regularization, it is expected that the PINN enforces the physics and produces solutions that satisfies the underlying ODEs.

Figure 6.7 shows the evolution of DG concentration over time using the PINN model at the end of its training (1M epochs). We compare the PINN's predictions to the numerical solution of the ODEs for three different microwave power outputs (4, 5 and 6 W). We note that the PINN solves almost perfectly the ODEs while fitting the experimental data as close as possible. Similar results are obtained with TG and MG. Indeed, the straight lines follow the dotted lines, meaning that the adaptative regularization play a major role in the training of the PINN. Without it, the PINN would not discover rate constants that are consistent with the predicted solution of the ODEs.

Finally, the architecture of the PINN influences the predictions. Firstly, adding the tempe-

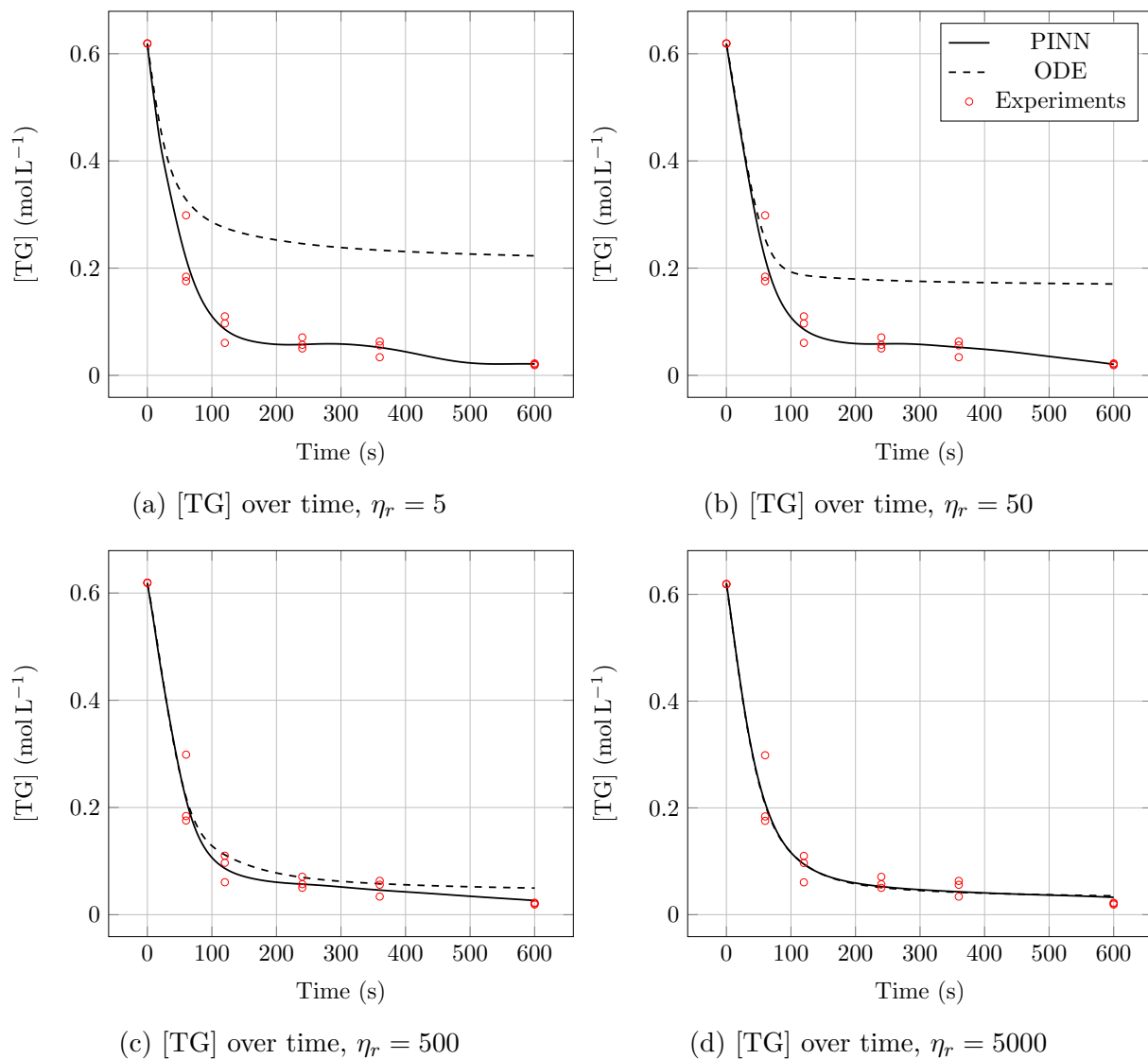


FIGURE 6.4 Impact of the regularization η_r on the predictions of the concentration of triglycerides and on the evaluation of the rate constants.

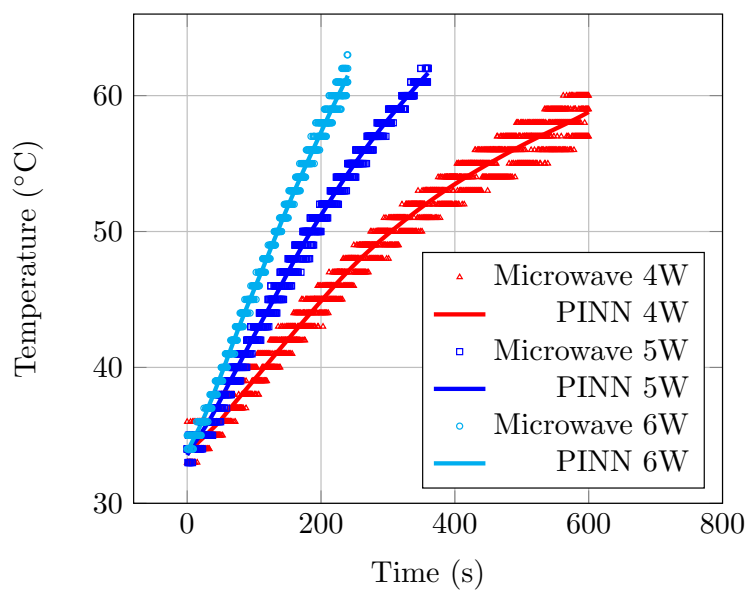


FIGURE 6.5 Temperature predicted by the PINN.

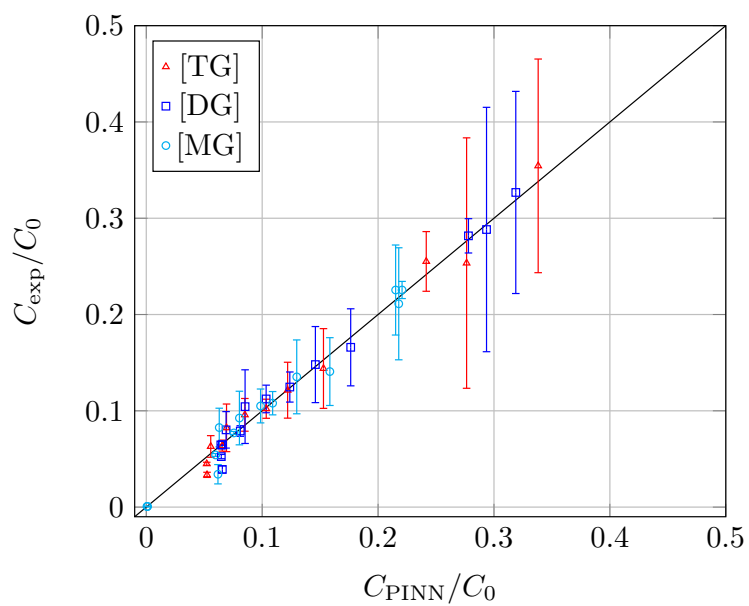


FIGURE 6.6 Parity plot between the concentration of glycerides C predicted by the PINN and the experimental results.

perature as an output impacts the concentration predictions. Indeed, the fact that the loss function includes the temperature data and the residual of the energy balance changes the optimization problem of the PINN. Not only the PINN plays the role of a filter, but it also finds the best fit between the temperature and the concentration by leveraging the Arrhenius equation. Secondly, adding the microwave power as a feature allows the PINN to predict distinct reaction rates depending on the operating condition of the reactor.

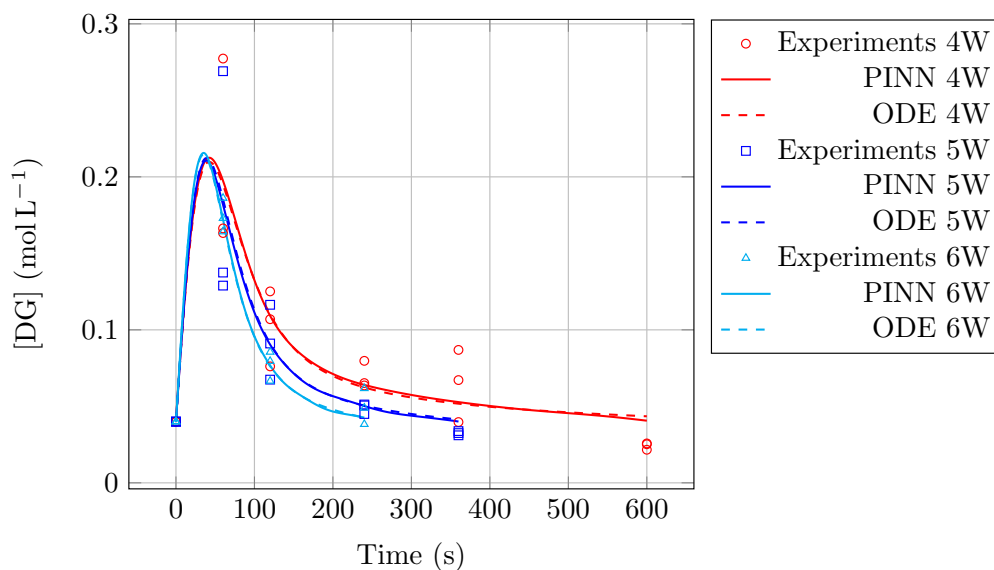


FIGURE 6.7 PINN’s predictions of the concentration of diglycerides over time compared to the numerical method and the experiments.

6.5.3 Extrapolation to conditions not trained for

We want to assess the capacity of the PINN to predict the rates of reactions that have unknown concentrations and have been performed under diverse operating conditions. To do so, we perform two different tests. For both tests, we train the 4, 5 and 6 W reactions by evaluating the loss of the ODEs describing the species balance (at the temporal collocation points), the loss of the ODE describing the energy balance (at the temporal collocation points) and the loss of the microwave temperature data (at each second of the reaction). For the first test, we add to the loss only the MSE of the 4 W concentration data, whereas for the second test, we add only the MSE of the 6 W concentration data.

Figure 6.8 shows the predictions of the DG concentration over time for the microwave power outputs of 5 and 6 W after training the PINN only with concentrations from the 4 W experiments. Figure 6.9 shows the same, but for the microwave powers of 4 and 5 W after training

only with concentrations from the 6 W experiments.

PINN's predictions from Figure 6.8 are almost all within the error bars of the experiments, without ever seeing the concentration data. PINN's predictions from Figure 6.9 are less consistent with the experiments. Similar results are obtained for TG and MG in both cases. The maximum reaction time of the 6 W experiments is 240 s, the time when the temperature reaches 65 °C and methanol starts to evaporate. Because no concentration data are available after this time, the PINN only fits the ODEs at the collocation points. The PINN finds easily and rapidly a local minima, where the concentration derivative is zero. That is why, from 240 s, all concentrations reach a plateau and do not predict accurately the experiments. Overall, if the PINN knows concentration data that cover all the time domain of the ODEs, the PINN successfully extrapolates, otherwise, it gets stuck in a local minima.

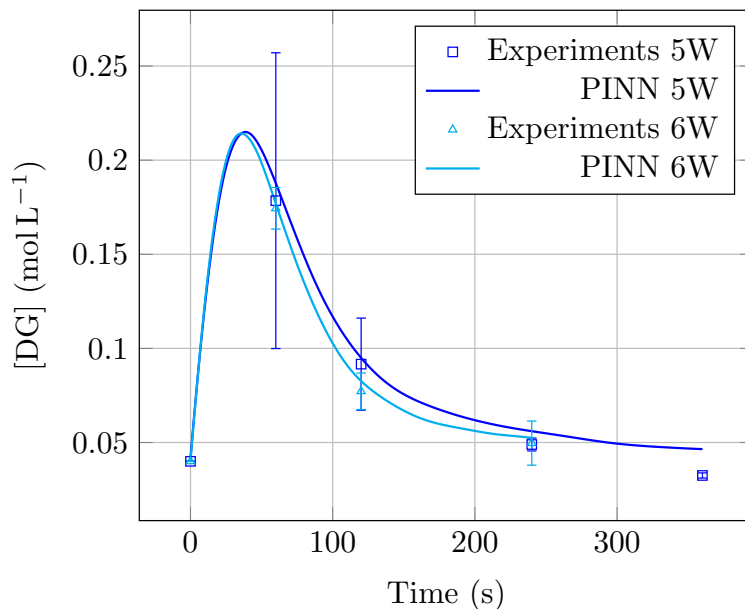


FIGURE 6.8 PINN's predictions of the concentration of diglycerides over time with only data from 4 W are known during training.

6.6 Conclusion

In this work, we propose a PINN algorithm, which combines a mathematical model with experimental measurements, to predict the reaction rates of a biodiesel process using microwaves. The mathematical model consists of a system of ODEs, which incorporates rate laws, species balance and a linearized form of the Arrhenius equation. Experimental measurements involve obtaining the products temperature from the IR sensor of the microwave reactor and

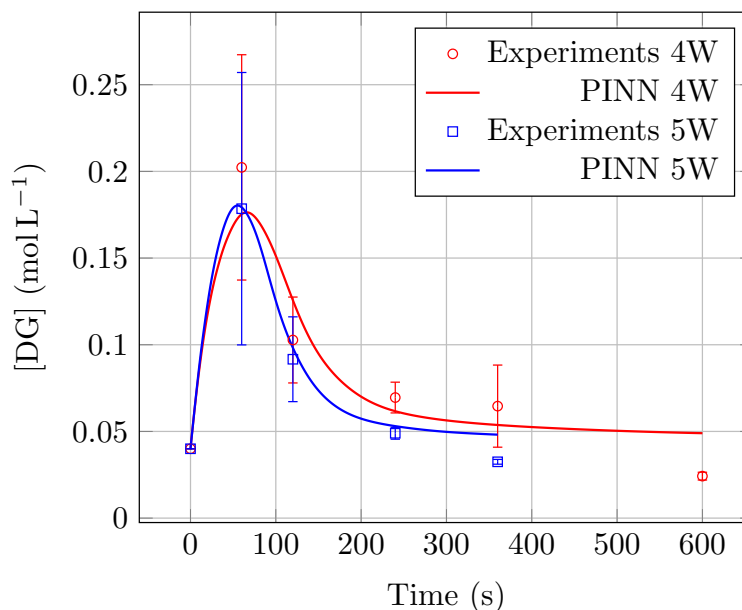


FIGURE 6.9 PINN’s predictions of the concentration of diglycerides over time with only data from 6 W are known during training.

the glycerides concentration using GC-FID. We train the PINN using an adaptive regularization parameter so its predictions are close to the data while respecting the underlying ODEs.

Results show that the PINN discovers rate constants k that satisfy the system of ODEs, predicts the species concentration as closely as possible to the experimental data and filters low-resolution temperature data. We also evaluate the ability of the PINN to extrapolate to operating conditions where concentration data are unknown. Results show that the PINN is able to predict concentrations within the error range of experiments that were not part of the training dataset. Based on these outcomes, we can conclude that with a set of ODEs, accessible temperature data and only few concentration data of non-negligible errors, a PINN is able to identify the reaction kinetics of a biodiesel process and extend its predictions to other microwave powers.

Future work includes the addition of other features that have an impact on the reaction kinetics such as : catalysis concentration, agitation speed and reaction volume. By adding more parameters to the design of experiments and to the inputs of the model, the PINN could formulate a full digital twin of the microwave reactor. We note that PINNs can be easily adapted to any reactions by altering the ODEs within the loss function. Consequently, we strongly believe that this approach is viable to rapidly constitute physics/data-driven hybrid model for intensified chemical reactors.

6.7 Acknowledgements

Valérie Bibeau would like to acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Canada Graduate Scholarships-Master's Program. Bruno Blais would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the RGPIN-2020-04510 Discovery Grant. This research was undertaken, in part, thanks to funding from the Canada Research Chair Program. The authors would also like to acknowledge technical support and computing time provided by the Digital Research Alliance of Canada and Calcul Québec. Computations were made on clusters Beluga managed by the Digital Research Alliance of Canada. The authors would like to acknowledge the efficient support received from Scion Instruments analysts for the GC-FID maintenance.

CHAPITRE 7 ARTICLE II. DISCUSSION

Ce présent chapitre a pour but de revenir sur certains éléments clés à retenir de l'article II ou d'approfondir certains concepts qui ont été superficiellement effleurés dans celui-ci. Pour ce faire, la discussion revoit les grands thèmes décrivant la méthodologie derrière le développement du réseau de neurones informé de la physique, soit le protocole expérimental permettant la génération des données de concentration et de température, le choix des neurones d'entrée et de sortie du réseau, l'influence de la régularisation sur les prédictions et finalement, des résultats supplémentaires démontrant le respect des bilans molaires.

7.1 Génération des données expérimentales

Le protocole expérimental demande que le flacon contenant l'huile et le méthanol doive être vigoureusement agité avant d'être placé dans la cavité du réacteur. La raison pour laquelle cette étape est suivie provient de l'immiscibilité importante des deux espèces chimiques. La tension de surface entre les deux phases est si importante que le transfert de masse entre le catalyseur solubilisé dans le méthanol et les glycérides de l'huile est fortement affecté. La Figure 7.1 démontre bien la différence entre le début de la réaction et sa fin. Les deux moments sont captés par la caméra du réacteur à micro-ondes. L'emplacement et la vitesse de l'agitateur magnétique influencent le nombre de Froude, ce qui impacte la formation du vortex et donc ajoute une certaine énergie cinétique permettant la rupture de la tension de surface. La température possède également un impact considérable sur la miscibilité des deux phases. Plus la température augmente, moins le biodiesel est visqueux, facilitant le mélange. Ainsi, au fur et à mesure que la réaction évolue, l'état du mélange change également, de la manifestation du vortex, à l'apparition de gouttelettes de méthanol tombant dans l'huile puis au mélange homogène. Ainsi, afin de rendre les expériences plus consistantes, tous les facteurs affectant le transfert de masse ont été écartés par une agitation préalable à la réaction.

7.2 Neurones d'entrée et de sortie

L'une des contributions importantes de l'article II est l'ajout d'un neurone d'entrée qui est une variable manipulée du système. Comme la revue de littérature le démontre, les PINNs n'utilisent généralement que les variables indépendantes des EDOs ou des EDPs (i.e, le temps et les coordonnées de l'espace du domaine à l'étude). L'ajout d'un neurone supplémentaire à l'entrée du réseau permet d'ajouter un attribut aux données de concentration. Cet attri-

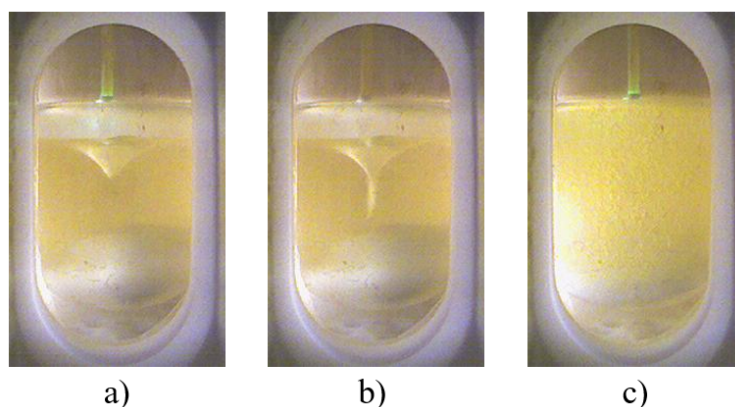


FIGURE 7.1 Évolution du mélange. a) Formation du vortex. b) Gouttelettes. c) Mélange homogène.

but, qui est la puissance du réacteur à micro-ondes, marque une différence évidente entre la cinétique des réactions faites à 4, 5 et 6 W. Si on enlève le neurone de puissance, le réseau établit exactement la même relation entre les concentrations et le temps, et ce, pour toutes les puissances. Cette observation est physiquement fausse, puisque la température évolue différemment selon la puissance et affecte la cinétique de la réaction de par l'équation d'Arrhenius. Même si la puissance n'est pas directement liée aux concentrations via les EDOs, le réseau est apte à établir une fonction hypothétique cohérente entre les deux variables. Bref, la puissance permet d'effectuer une régression non linéaire entre les concentrations et une condition d'opération importante du réacteur.

Un autre aspect important de l'architecture du réseau est le neurone de température. Plusieurs itérations sur la connexion du neurone au réseau ont été faites afin de déterminer l'architecture optimale. La Figure 7.2 démontre les endroits où la température a été ajoutée au réseau. Tout d'abord, il a été tenté de remplacer le neurone de puissance par le neurone de température. Étant donné que l'évolution de la concentration dépend de la constante cinétique, qui elle dépend de la température, l'ajout de celle-ci en tant qu'attribut semblait plausible. Cependant, l'entraînement du réseau était hasardeux. La fonction de perte avait de la difficulté à converger et les prédictions n'avaient aucun sens physique. Cette observation pourrait s'expliquer de par la relation hautement non linéaire entre les concentrations et la température, rendant la descente du gradient laborieuse. Une autre tentative était de connecter la température au réseau en entier. On disait alors que la température était un neurone d'entrée « flottant » au réseau. Elle n'était donc pas considérée comme un attribut du problème. Ses valeurs, à chaque point de collocation temporel, étaient connues par l'algorithme. Cette façon de faire amenait des prédictions plausibles et physiques. Toutefois, l'architecture

de l'article II, qui propose de connecter la température à la couche de sortie du réseau, réalise des prédictions plus proches des données expérimentales. L'ajout de la température en sortie exige par défaut l'ajout de fonctions de perte qui évaluent l'accord entre les prédictions, les données du capteur infrarouge du réacteur et le bilan d'énergie. La fonction de perte globale possède alors un paysage complètement différent et le problème d'optimisation change également. Bref, dans l'article II, les neurones d'entrée contiennent les variables d'opération du réacteur, soit le temps de réaction et la puissance des micro-ondes, alors que les neurones de sortie sont les mesures des variables d'état, soit la concentration et la température. Cette façon plus logique d'agencer les neurones semble amener à des prédictions plus cohérentes.

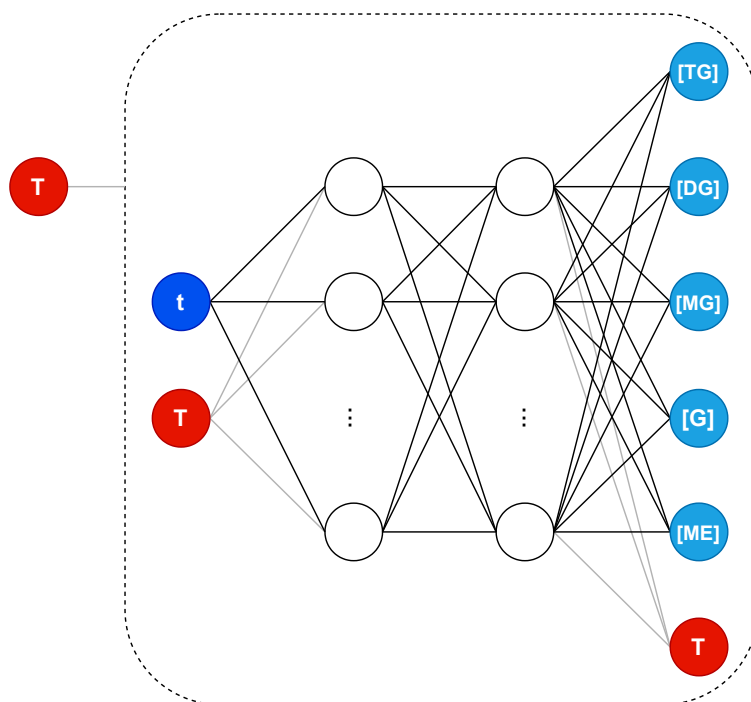


FIGURE 7.2 Emplacements possibles du neurone de température dans l'architecture.

7.3 Régularisation

L'article II évalue en profondeur l'influence du paramètre de régularisation sur les prédictions de concentration. D'une part, le paramètre de régularisation était assez sévère, ce qui permet d'obtenir des prédictions logiques, suivant le sens physique donné par les EDOs. D'autre part, la régularisation a été faite de manière progressive durant l'entraînement, permettant une descente de gradient plus facile en début d'entraînement. Toutefois, le changement brusque du paramètre de régularisation fait en sorte que le paysage de la fonction de perte change de manière draconienne d'une itération à l'autre. La Figure 7.3 permet de visualiser le changement

brusque des fonctions de perte. Celle qui évalue l'accord entre les prédictions et les données expérimentales augmente, puis prend un certain temps avant de diminuer à nouveau. La fonction de perte qui évalue le résidu des EDOs chute excessivement. Il faut un certain nombre d'itérations, environ 25 000, avant que la fonction de perte des données expérimentales commence à redescendre jusqu'à retrouver son ancienne valeur (précédant le changement de η_r). À ce moment, il est possible d'augmenter à nouveau le paramètre de régularisation. Afin d'arriver à des prédictions plausibles, la progression se fait assez lentement et l'entraînement demande 1 million d'itérations, ce qui prend environ 4 heures à rouler sur un GPU de la grappe de calcul de Béluga. Ce temps de calcul n'est pas négligeable.

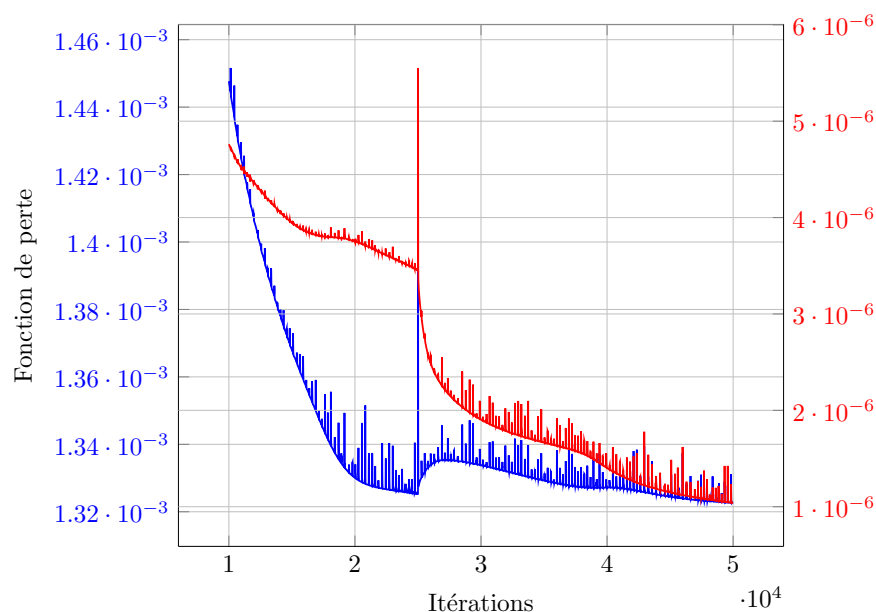


FIGURE 7.3 Fonction de perte des données de concentration (en **bleu**) et des EDOs (en **rouge**).

L'article II n'examine pas du tout l'influence de la régularisation sur le bilan d'énergie. En effet, η_r est un paramètre qui s'applique seulement sur la fonction de perte du résidu des EDOs décrivant les bilans molaires. On discute ici la raison pour laquelle aucune régularisation n'a été appliquée à l'EDO décrivant le bilan d'énergie. La Figure 7.4 compare l'évolution de température de la réaction et celle de l'huile sans catalyseur (où aucune réaction n'a lieu). On remarque qu'au début de la réaction, la température augmente puis, autour de 300 s, elle se stabilise et évolue de la même manière que l'huile. L'hypothèse derrière cette augmentation de température plus marquée provient de la nature exothermique de la réaction. Lorsque les glycérides ont atteint l'équilibre chimique, la température continue d'augmenter grâce à la radiation des micro-ondes, mais son évolution est très similaire à l'huile. Le bilan

d'énergie implémenté dans l'article II suppose une évolution quadratique de la température. Elle ne considère pas la thermodynamique de la réaction. C'est pour cette raison qu'aucune régularisation n'est utilisée pour le bilan d'énergie, puisque son EDO est fortement simplifié.

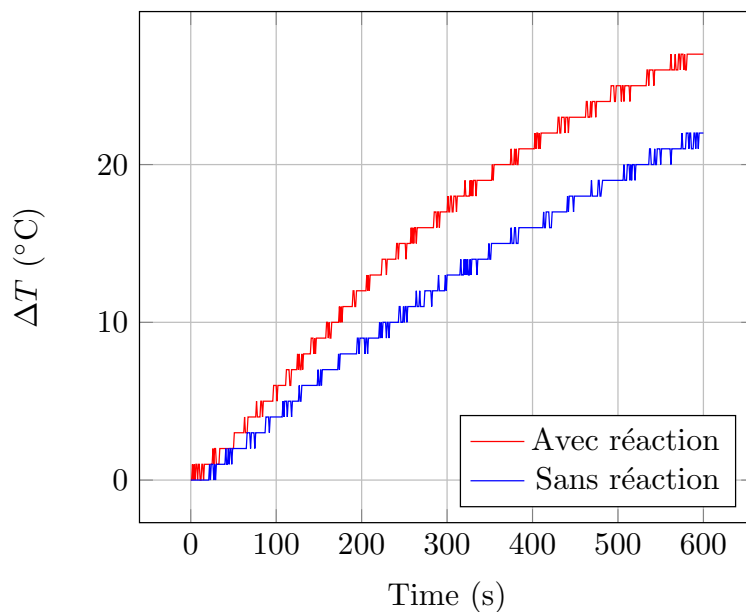


FIGURE 7.4 Évolution de la variation de température selon le temps de la réaction de transestérification et de l'huile de canola (sans réaction).

La Figure 7.5 montre les prédictions du PINN, la résolution de l'EDO par Runge-Kutta et les données du capteur infrarouge. On remarque que les prédictions du PINN sont plus proches des données et divergent légèrement de l'EDO. L'évolution quadratique de la température ne permet pas de représenter la température réelle du réacteur et ignore l'effet exothermique de la réaction. En régularisant le bilan d'énergie avec moins de sévérité que les bilans molaires, la température est plus flexible à épouser les données et permet au PINN de capter l'augmentation soudaine et marquée de la température. De cette manière, on obtient une relation plus réaliste et physique entre les données de concentration et celles de température.

7.4 Respect des bilans molaires

Finalement, certains résultats ont été omis dans l'article II. Les concentrations du glycérol ([G]) et du biodiesel ([ME]) sont prédites par le PINN, mais ne sont pas discutées. Il a été décidé que le glycérol et le biodiesel n'allaient pas être analysés. Étant donné que le glycérol est extrêmement visqueux, il était plus difficile de le solubiliser dans un solvant pour son passage dans la colonne du GC. De plus, même en suivant les protocoles standards d'analyse

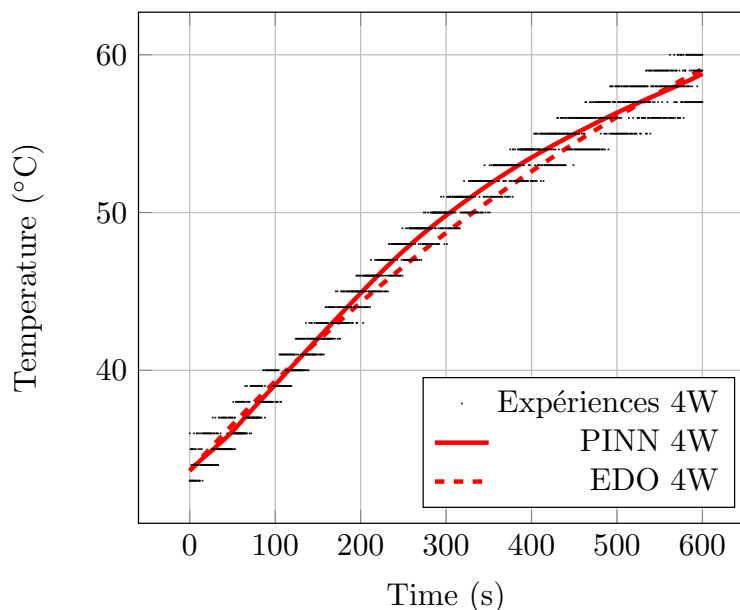


FIGURE 7.5 Évolution de la température selon le temps prédite par le PINN, résolue par Runge-Kutta et décrite par les données expérimentales du réacteur.

du glycérol et du biodiesel, les mesures de concentration présentaient de grandes erreurs. Ainsi, le PINN est entraîné considérant seulement leur condition initiale et leur EDO. La Figure 7.6 montre les prédictions du PINN pour les concentrations de G et ME avec une régularisation faible. La cinétique n'est pas respectée dans ce cas. La Figure 7.7, quant à elle, montre un meilleur accord entre les prédictions et l'EDO encodé dans l'entraînement.

Une dernière analyse, rapide et simple, a été réalisée afin de vérifier si les concentrations de G et ME prédites par le PINN respectaient le bilan molaire global de la réaction de transestérification. On sait que la concentration initiale des triglycérides est d'environ 0.6 mol L^{-1} , que sa concentration finale prédite par le PINN est d'environ 0.03 mol L^{-1} et que la réaction globale de la réaction s'écrit comme étant :



Alors, 1 mole de TG donne 1 mole de G et 3 moles de ME. Si l'avancement de la réaction selon TG est de 0.57 mol L^{-1} , on s'attend à ce que, en fin de réaction, la concentration de G soit autour de 0.57 mol L^{-1} et que la concentration de ME soit autour de 1.7 mol L^{-1} . Les prédictions du PINN sont proches de ces valeurs, à 7 % et 1 % d'erreur pour G et ME

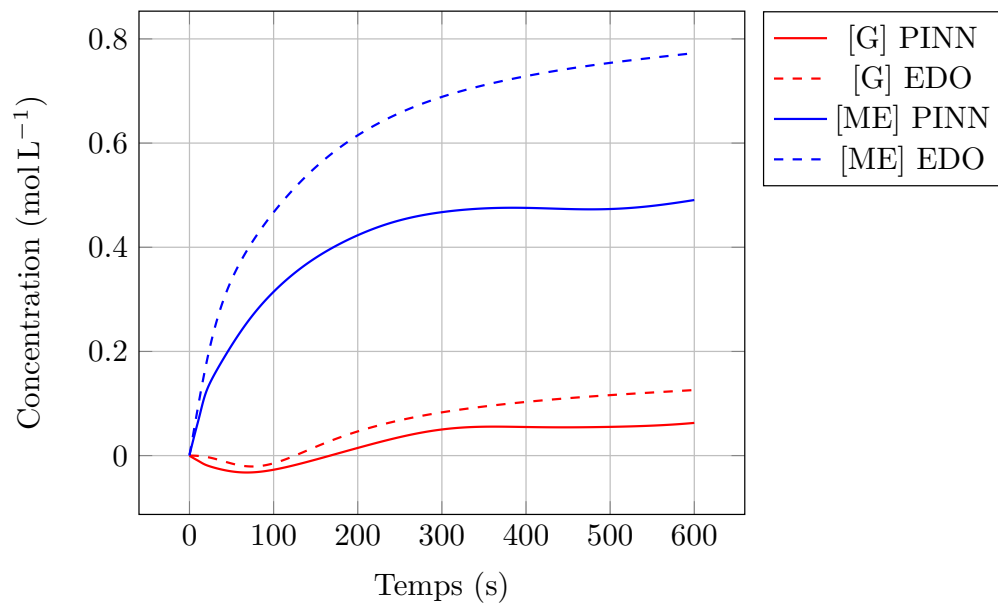


FIGURE 7.6 Concentration de G et ME selon le temps prédite par le PINN et résolue par Runge-Kutta pour une régularisation $\eta_r = 5$.

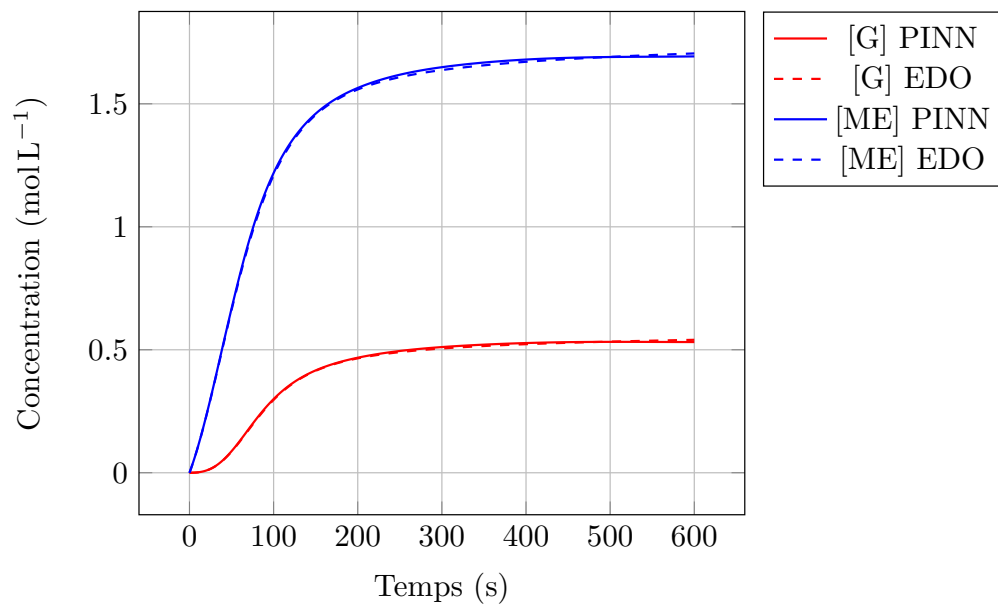


FIGURE 7.7 Concentration de G et ME selon le temps prédite par le PINN et résolue par Runge-Kutta pour une régularisation $\eta_r = 5000$.

respectivement. Il est possible de dire que, grâce à une forte régularisation sur les bilans molaires, le bilan global de la réaction est respecté, et ce, même si les données de G et ME ne font pas partie de la méthodologie ni de l'entraînement du PINN.

CHAPITRE 8 CONCLUSION

8.1 Synthèse des travaux

Ce projet de maîtrise visait à développer des outils d'apprentissage profond afin de résoudre des problèmes d'ingénierie concrets tels que la détermination du rendement énergétique d'un mélangeur industriel et l'identification de la cinétique chimique d'une réaction se déroulant dans un réacteur à micro-ondes. Dans les deux cas, la méthodologie requiert trois étapes cruciales, soit la génération des données, qu'elles soient numériques ou expérimentales, l'implémentation des réseaux de neurones et l'évaluation de la performance de ces derniers suite à leur entraînement.

En ce qui a trait au premier objectif, les données numériques du nombre de puissance des mélangeurs ont été obtenues via des simulations numériques, où le logiciel libre Lethe a été employé pour résoudre les équations de dynamique des fluides par la méthode des éléments finis. Au total, 100 000 simulations ont roulé sur des grappes de calcul appartenant à l'Alliance de recherche numérique du Canada. Ainsi, avec un logiciel de CFD performant, une vérification de la sensibilité du maillage et un échantillonnage robuste, le sous-objectif **S.O.1** est réussi. Ensuite, le logiciel informatique qui implémente le réseau de neurones a été développé avec la librairie TensorFlow. Afin de déterminer les différents hyperparamètres du réseau, une validation croisée a été effectuée afin d'implémenter une architecture optimale et de contrôler efficacement la descente du gradient. Cette méthodologie permet de prédire le nombre de puissance avec une excellente métrique de performance. L'article I présente les résultats obtenus avec l'utilisation du réseau de neurones. Ce dernier est capable de prédire le nombre de puissance avec davantage de précision que les corrélations algébriques présentes dans la littérature. Il est capable de généraliser ses prédictions à plusieurs géométries différentes (ne faisant pas partie de l'entraînement) et ce, à une erreur relative de 0.5 %. Ainsi, les sous-objectifs **S.O.2** et **S.O.3** ont été atteints.

En ce qui concerne le deuxième objectif, les données étaient générées par des expériences en laboratoire. La réaction de transestérification de l'huile de canola pour la production de biodiesel a été reproduite à différents temps (de 1 à 10 minutes) et à quelques conditions d'opération distinctes (à une puissance des micro-ondes de 4, 5 et 6 W). Les concentrations de glycérides ont été analysées par une méthode de chromatographie gazeuse (GC-FID). La température, quant à elle, a été mesurée via le capteur infrarouge du réacteur. Chaque expérience a été réalisée de sorte à avoir un triplicata. De cette manière, il était possible d'évaluer la reproductibilité des expériences. Ainsi, le protocole expérimental robuste permettant la

génération des données en concentration et en température, nécessaire à la base de données du PINN, répond au sous-objectif **S.O.1**. Ayant les données en main, le PINN pouvait être implémenté. Cette fois-ci, la librairie PyTorch a été employée afin de développer l'algorithme du PINN. Les fonctions de perte ont été programmées de sorte à évaluer la forme résiduelle des EDOs décrivant les bilans molaires des glycérides et le bilan d'énergie du réacteur. La fonction de perte qui évalue l'erreur entre les prédictions du PINN et les données expérimentales a également été implémentée. Une attention particulière a été mise sur l'entraînement du PINN. Utilisant une régularisation progressive, le PINN était en mesure de s'adapter plus facilement aux EDOs encodés dans l'algorithme. L'article II montre les résultats démontrant la capacité du PINN à identifier la cinétique chimique d'une production de biodiesel. Le PINN est tout aussi apte à prédire des concentrations proches des expériences qu'à respecter la physique du problème à l'étude. De plus, l'article II démontre la capacité du PINN à généraliser à d'autres puissances de micro-ondes, ce qui fait de lui un bon candidat pour représenter le jumeau numérique du réacteur. Grâce à ces résultats, il est possible de dire que les sous-objectifs **S.O.2** et **S.O.3** ont été réussis.

Somme toute, ces deux projets tentent de rapprocher, que ce soit d'un petit pas (ou même de deux), la recherche en intelligence artificielle à sa concrétisation en génie chimique. Que ce soit avec des données volumineuses ou des données ayant des erreurs non négligeables, des données provenant de simulations numériques ou des données expérimentales en laboratoire, un modèle demandant plusieurs équations physiques ou aucune, l'apprentissage profond peut facilement s'adapter, à quelques exceptions près.

8.2 Limitations de la solution proposée

Les réseaux de neurones sont relativement faciles à implémenter, mais demandent plusieurs réglages avant d'avoir un algorithme performant. La difficulté derrière l'apprentissage profond se cache derrière les données qui sont alimentées au réseau. Comme il a été démontré avec les mélangeurs, un soin particulier doit être porté aux étapes d'échantillonnage, du maillage et de la résolution numérique par CFD, ce qui n'est pas une tâche facile à accomplir. Une minutie doit être apportée dans la production des données et cela n'est pas toujours évident à exécuter en industrie. Les PINNs, quant à eux, ne demandent pas une quantité énorme de données pour son entraînement. Par contre, comme il a été discuté à la suite de l'article II, les données de température étaient difficilement exploitables en tant qu'attributs $\mathbf{x}$ à la fonction hypothétique f . Cela signifie que, potentiellement, les PINNs éprouvent de grandes difficultés à exploiter la richesse d'information d'un procédé en temps réel. Finalement, il se peut que les méthodes classiques de régression non linéaire performant aussi bien que les PINNs. Une

vérification plus approfondie aurait permis de confirmer ou d’infirmer la capacité des PINNs à surpasser les méthodes traditionnelles.

8.3 Améliorations futures

Dans l’article I, on démontre que les réseaux de neurones possèdent une grande capacité de généralisation . Avec une certaine expertise en CFD, il serait possible de générer des données numériques d’autres problèmes de mécanique des fluides en génie chimique, comme l’écoulement d’un liquide ou d’un gaz au sein d’un lit de particules ou l’écoulement d’un fluide non newtonien. Par exemple, dans le groupe de recherche de l’URPEI, un réseau de neurones est utilisé afin de prédire l’influence des forces entre deux particules, et ce, pour différents régimes d’écoulement. De plus, il serait pertinent de vérifier l’application du réseau de neurones à des opérations unitaires qui impliquent d’autres phénomènes d’échange tel le transfert de chaleur.

Dans l’article II, l’architecture du PINN possède la puissance des micro-ondes comme attribut. Cela permettait de différencier l’évolution des concentrations des glycérides en fonction d’une condition d’opération du réacteur. D’autres conditions ont un impact majeur sur la cinétique du biodiesel, comme la concentration initiale du catalyseur ou la rotation de l’agitateur. Une première amélioration serait d’ajouter d’autres attributs afin de vérifier si le PINN est réellement apte à extrapoler à plusieurs conditions différentes et de prouver davantage son potentiel de jumeau numérique. De plus, les données expérimentales présentaient une erreur non négligeable. Étant donné la rapidité de la réaction en présence de micro-ondes, l’analyse des glycérides était assez difficile, ce qui rendait les expériences plus ou moins reproductibles. Il aurait été intéressant d’ajouter à l’algorithme du PINN une composante bayésienne qui prend en charge l’écart-type des données. La recherche sur les PINNs demeure encore embryonnaire. À ce jour, peu d’applications des PINNs sur des problématiques concrètes sont mises en œuvre. En théorie, les PINNs s’entraînent sur des EDOs ou des EDPs. On ne manque pas de ce type d’équations en ingénierie. Ainsi, les PINNs ne manquent qu’à être validés davantage à plusieurs types de problèmes différents.

8.4 Mot de la fin

Comme la plupart des étudiants et étudiantes à Polytechnique, j’ai décidé de poursuivre mes études aux cycles supérieurs dans le but de m’embarquer dans des projets motivants qui pourraient ébranler, juste assez, la façon dont on conçoit nos industries. Le dynamisme de la nouvelle génération d’ingénieurs et d’ingénieures à vouloir changer les choses est palpable, ce

qui me motive à utiliser ma créativité pour voir le génie chimique de demain d'un œil différent. Les fondements du génie chimique sont solides, personne ne peut le nier. Toutefois, l'industrie change, elle évolue, les données se multiplient et une évolution s'amorce, qu'on le veuille ou non. Utilisons les meilleurs outils qui soient afin de nous propulser plus loin vers l'avant. C'est donc avec fierté, un peu de fébrilité et beaucoup d'enthousiasme que je vous présente ce mémoire. Je souhaite fortement que le travail accompli puisse ouvrir certaines portes, petites soient elles, vers une industrie 4.0 profitable et qui respecte notre environnement si fragile.

RÉFÉRENCES

- [1] E. L. Paul, V. A. Atiemo-Obeng et S. M. Kresta, *Handbook of industrial mixing*. Wiley Online Library, 2004.
- [2] J. Rushton, “Power characteristics of mixing impellers part 1,” *Chem. Eng. Prog.*, vol. 46, p. 395–404, 1950.
- [3] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin et D. Wells, “The deal.II finite element library : Design, features, and insights,” *Computers & Mathematics with Applications*, vol. 81, p. 407–422, 2021. [En ligne]. Disponible : <https://arxiv.org/abs/1910.13247>
- [4] H. S. Fogler, *Elements of chemical reaction engineering*. Pearson Boston, 2020.
- [5] H. Li, Z. Xu, G. Taylor, C. Studer et T. Goldstein, “Visualizing the loss landscape of neural nets,” *Advances in neural information processing systems*, vol. 31, 2018.
- [6] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang et L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, n°. 6, p. 422–440, 2021.
- [7] W. L. McCabe, J. C. Smith et P. Harriott, *Unit operations of chemical engineering*. McGraw-Hill New York, 1993, vol. 5.
- [8] F. P. Incropera, D. P. DeWitt, T. L. Bergman, A. S. Lavine *et al.*, *Fundamentals of heat and mass transfer*. Wiley New York, 1996, vol. 6.
- [9] S. M. Hall, *Rules of Thumb for Chemical Engineers (6th Edition)*. Elsevier, 2018. [En ligne]. Disponible : <https://app.knovel.com/hotlink/toc/id:kpRTCEE013/rules-thumb-chemical/rules-thumb-chemical>
- [10] B. A. Patel et C. S. Pereira, “Process intensification at scale : An industrial perspective,” *Chemical Engineering and Processing-Process Intensification*, vol. 181, p. 109098, 2022.
- [11] V. Venkatasubramanian, “The promise of artificial intelligence in chemical engineering : Is it here, finally ?” *AIChE Journal*, vol. 65, n°. 2, p. 466–478, 2019.
- [12] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.
- [13] D. G. Harkut, *Artificial intelligence-Scope and limitations*. IntechOpen, 2019.
- [14] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel et M. Harnisch, “Industry 4.0 : The future of productivity and growth in manufacturing industries,” *Boston consulting group*, vol. 9, n°. 1, p. 54–89, 2015.

- [15] Y. Liao, F. Deschamps, E. d. F. R. Loures et L. F. P. Ramos, “Past, present and future of industry 4.0-a systematic literature review and research agenda proposal,” *International journal of production research*, vol. 55, n^o. 12, p. 3609–3629, 2017.
- [16] T. Kumaresan et J. B. Joshi, “Effect of impeller design on the flow pattern and mixing in stirred tanks,” *Chemical engineering journal*, vol. 115, n^o. 3, p. 173–193, 2006.
- [17] I. Houcine, E. Plasari et R. David, “Effects of the stirred tank’s design on power consumption and mixing time in liquid phase,” *Chemical Engineering & Technology : Industrial Chemistry-Plant Equipment-Process Engineering-Biotechnology*, vol. 23, n^o. 7, p. 605–613, 2000.
- [18] A. Constantinides et N. Mostoufi, *Numerical Methods for Chemical Engineers with MATLAB Applications*. Prentice Hall PTR, 1999.
- [19] E. Goursat, *A course in mathematical analysis*. Ginn, 1904, vol. 1.
- [20] A. Fortin, *Analyse numérique pour ingénieurs*. Presses inter Polytechnique, 2008.
- [21] J. N. Reddy, *Introduction to the finite element method*. McGraw-Hill Education, 2019.
- [22] O. Campesato, *Artificial intelligence, machine learning, and deep learning*. Mercury Learning and Information, 2020.
- [23] Y. LeCun, Y. Bengio et G. Hinton, “Deep learning,” *nature*, vol. 521, n^o. 7553, p. 436–444, 2015.
- [24] M. Raissi, P. Perdikaris et G. E. Karniadakis, “Physics-informed neural networks : A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, vol. 378, p. 686–707, 2019.
- [25] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : Concepts, tools, and techniques to build intelligent systems*. O’Reilly Media, Inc., 2019.
- [26] X. Glorot et Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” dans *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, p. 249–256.
- [27] D. P. Kingma et J. Ba, “Adam : A method for stochastic optimization,” *arXiv preprint arXiv :1412.6980*, 2014.
- [28] D. A. Pratama, R. R. Abo-Alsabeh, M. A. Bakar, A. Salhi et N. F. Ibrahim, “Solving partial differential equations with hybridized physic-informed neural network and optimization approach : Incorporating genetic algorithms and l-bfgs for improved accuracy,” *Alexandria Engineering Journal*, vol. 77, p. 205–226, 2023.

- [29] Y. Bengio, E. Laufer, G. Alain et J. Yosinski, “Deep generative stochastic networks trainable by backprop,” dans *International Conference on Machine Learning*. PMLR, 2014, p. 226–234.
- [30] U. Michelucci, *Advanced applied deep learning : convolutional neural networks and object detection*. Springer, 2019.
- [31] A. K. Tyagi et A. Abraham, *Recurrent Neural Networks : Concepts and Applications*. CRC Press, 2022.
- [32] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby et M. W. Mahoney, “Characterizing possible failure modes in physics-informed neural networks,” *Advances in Neural Information Processing Systems*, vol. 34, p. 26 548–26 560, 2021.
- [33] E. Kharazmi, Z. Zhang et G. E. Karniadakis, “Variational physics-informed neural networks for solving partial differential equations,” *arXiv preprint arXiv :1912.00873*, 2019.
- [34] L. Yang, X. Meng et G. E. Karniadakis, “B-pinns : Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data,” *Journal of Computational Physics*, vol. 425, p. 109913, 2021.
- [35] A. D. Jagtap et G. E. Karniadakis, “Extended physics-informed neural networks (xpinns) : A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations.” dans *AAAI spring symposium : MLPS*, vol. 10, 2021.
- [36] L. Sabat et C. K. Kundu, “History of finite element method : a review,” *Recent Developments in Sustainable Infrastructure : Select Proceedings of ICRDSI 2019*, p. 395–404, 2020.
- [37] T. G. Grossmann, U. J. Komorowska, J. Latz et C.-B. Schönlieb, “Can physics-informed neural networks beat the finite element method?” *arXiv preprint arXiv :2302.04107*, 2023.
- [38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow : Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv :1603.04467*, 2016.
- [39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch : An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.
- [40] H. Dai, X. Peng, X. Shi, L. He, Q. Xiong et H. Jin, “Reveal training performance mystery between tensorflow and pytorch in the single gpu environment,” *Science China Information Sciences*, vol. 65, p. 1–17, 2022.

- [41] V. Bibeau, “lethe-cfd/mixing-ann,” 2022. [En ligne]. Disponible : <https://github.com/lethe-cfd/mixing-ann>
- [42] V. Bibeau, “lethe-cfd/bio-pinn,” 2022. [En ligne]. Disponible : <https://github.com/lethe-cfd/bio-pinn>
- [43] B. Blais, L. Barbeau, V. Bibeau, S. Gauvin, T. El Geitani, S. Golshan, R. Kamble, G. Mirakhori et J. Chaouki, “Lethe : An open-source parallel high-order adaptative cfd solver for incompressible flows,” *SoftwareX*, vol. 12, p. 100579, 2020.
- [44] T. El Geitani, S. Golshan et B. Blais, “Toward high-order cfd-dem : Development and validation,” *Industrial & Engineering Chemistry Research*, vol. 62, n^o. 2, p. 1141–1159, 2023.
- [45] C.-A. Daunais, L. Barbeau et B. Blais, “An extensive study of shear thinning flow around a spherical particle for power-law and carreau fluids,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 311, p. 104951, 2023.
- [46] L. Barbeau, S. Étienne, C. Béguin et B. Blais, “Development of a high-order continuous galerkin sharp-interface immersed boundary method and its application to incompressible flow problems,” *Computers & Fluids*, vol. 239, p. 105415, 2022.
- [47] J. Joachim, C.-A. Daunais, V. Bibeau, L. Heltai et B. Blais, “A parallel and adaptative nitsche immersed boundary method to simulate viscous mixing,” *Journal of Computational Physics*, vol. 488, p. 112189, 2023.
- [48] B. Blais et al., 2020. [En ligne]. Disponible : <https://github.com/lethe-cfd/lethe>
- [49] B. Blais, M. Lassaigue, C. Goniva, L. Fradette et F. Bertrand, “A semi-implicit immersed boundary method and its application to viscous mixing,” *Computers & Chemical Engineering*, vol. 85, p. 136–146, 2016.
- [50] R. B. Bird, W. E. Stewart et E. N. Lightfoot, *Transport phenomena*. John Wiley & Sons, Inc., 2007.
- [51] R. L. Bates, P. L. Fondy et R. R. Corpstein, “Examination of some geometric parameters of impeller power,” *Industrial & Engineering Chemistry Process Design and Development*, vol. 2, n^o. 4, p. 310–314, 1963.
- [52] S. Nagata, *Mixing : principles and applications*. Kodansha, 1975.
- [53] S. Nagata, T. Tokoyama et H. Maeda, “Studies on the power requirement of paddle agitators in cylindrical vessels,” *Kagaku Kogaku*, vol. 20, n^o. 11, p. 582–592, 1956.
- [54] S. Hiraoka, N. Kamei, Y. Kato, Y. Tada, H.-G. Cheon et T. Yamaguchi, “Power correlation for pitched blade paddle impeller in agitated vessels with and without baffles,” *Kagaku Kogaku Ronbunshu*, vol. 23, n^o. 6, p. 969–975, 1997.

- [55] Y. Kato, Y. Tada, Y. Takeda, Y. Hirai et Y. Nagatsu, "Correlation of power consumption for propeller and pfaudler type impellers," *Journal of Chemical Engineering of Japan*, vol. 42, n^o. 1, p. 6–9, 2009.
- [56] W. Suewatanakul, "A comparison of fault detection and classification using artificial neural networks with traditional methods," Thèse de doctorat, University of Texas, 1993.
- [57] J. C. MacMurray et D. M. Himmelblau, "Modeling and control of a packed distillation column using artificial neural networks," *Computers & chemical engineering*, vol. 19, n^o. 10, p. 1077–1088, 1995.
- [58] M. Rostamizadeh, D. C. Boffito, G. S. Patience et A. Taeb, "Neural network modeling of methanol to propylene over p-zsm-5 in a fluidized bed," *J Chem Proc Engg*, vol. 1, p. 1–8, 2014.
- [59] K. K. Singh, K. T. Shenoy, A. K. Mahendra et S. K. Ghosh, "Artificial neural network based modelling of head and power characteristics of pump-mixer," *Chemical engineering science*, vol. 59, n^o. 14, p. 2937–2945, 2004.
- [60] R. Hemrajani et G. Tatterson, *Mechanically Stirred Vessels*. Wiley Online Library, 01 2004, p. 345–390.
- [61] D. Chapple, S. M. Kresta, A. Wall et A. Afacan, "The effect of impeller and tank geometry on power number for a pitched blade turbine," *Chemical Engineering Research and Design*, vol. 80, n^o. 4, p. 364–372, 2002.
- [62] K. Rutherford, S. M. Mahmoudi, K. C. Lee et M. Yianneskis, "The influence of rush-ton impeller blade and disk thickness on the mixing characteristics of stirred vessels," *Chemical engineering research & design*, vol. 74, n^o. 3, p. 369–378, 1996.
- [63] C. Geuzaine et J.-F. Remacle, "Gmsh : A 3-d finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, n^o. 11, p. 1309–1331, 2009.
- [64] B. Delacroix, A. Bouarab, L. Fradette, F. Bertrand et B. Blais, "Simulation of granular flow in a rotating frame of reference using the discrete element method," *Powder technology*, vol. 369, p. 146–161, 2020.
- [65] M. D. McKay, R. J. Beckman et W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, n^o. 1, p. 55–61, 2000.
- [66] M. Zakaria, M. Al-Shebany et S. Sarhan, "Artificial neural network : a brief overview," *International Journal of Engineering Research and Applications*, vol. 4, n^o. 2, p. 7–12, 2014.

- [67] A. Gökhan, C. O. Güzeller et M. T. Eser, “The effect of the normalization method used in different sample sizes on the success of artificial neural network model,” *International Journal of Assessment Tools in Education*, vol. 6, n^o. 2, p. 170–192, 2019.
- [68] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn : Machine learning in python,” *the Journal of machine Learning research*, vol. 12, p. 2825–2830, 2011.
- [69] W. L. Oberkamp et C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.
- [70] H. Furukawa, Y. Kato, Y. Inoue, T. Kato, Y. Tada et S. Hashimoto, “Correlation of power consumption for several kinds of mixing impellers,” *International Journal of Chemical Engineering*, vol. 2012, 2012.
- [71] F. R. Hama, J. D. Long et J. C. Hegarty, “On transition from laminar to turbulent flow,” *Journal of Applied Physics*, vol. 28, n^o. 4, p. 388–394, 1957.
- [72] P. A. Tanguy et F. Thibault, “Power consumption in the turbulent regime for a coaxial mixer,” *The Canadian Journal of Chemical Engineering*, vol. 80, n^o. 4, p. 601–603, 2002.
- [73] A. Stankiewicz et J. A. Moulijn, “Process intensification,” *Industrial & Engineering Chemistry Research*, vol. 41, n^o. 8, p. 1920–1924, 2002.
- [74] A. I. Stankiewicz, J. A. Moulijn *et al.*, “Process intensification : transforming chemical engineering,” *Chemical engineering progress*, vol. 96, n^o. 1, p. 22–34, 2000.
- [75] E. Pahija, S. Golshan, B. Blais et D. C. Boffito, “Perspectives on the process intensification of co2 capture and utilization,” *Chemical Engineering and Processing-Process Intensification*, vol. 176, p. 108958, 2022.
- [76] E. Thostenson et T.-W. Chou, “Microwave processing : fundamentals and applications,” *Composites Part A : Applied Science and Manufacturing*, vol. 30, n^o. 9, p. 1055–1071, 1999.
- [77] S. Nomanbhay et M. Y. Ong, “A review of microwave-assisted reactions for biodiesel production,” *Bioengineering*, vol. 4, n^o. 2, p. 57, 2017.
- [78] C. S. Lee, E. Binner, C. Winkworth-Smith, R. John, R. Gomes et J. Robinson, “Enhancing natural product extraction and mass transfer using selective microwave heating,” *Chemical Engineering Science*, vol. 149, p. 97–103, 2016.
- [79] H. C. Foley, *Introduction to Chemical Engineering Analysis Using Mathematica : For Chemists, Biotechnologists and Materials Scientists*. Academic Press, 2021.

- [80] D. Himmelblau, C. Jones et K. Bischoff, "Determination of rate constants for complex kinetics models," *Industrial & Engineering Chemistry Fundamentals*, vol. 6, n^o. 4, p. 539–543, 1967.
- [81] L. P. de Oliveira, D. Hudebine, D. Guillaume et J. J. Verstraete, "A review of kinetic modeling methodologies for complex processes," *Oil & Gas Science and Technology—Revue d'IFP Energies nouvelles*, vol. 71, n^o. 3, p. 45, 2016.
- [82] W. H. Green, "Moving from postdictive to predictive kinetics in reaction engineering," *MIT Open Access Articles*, 2020.
- [83] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi et F. Piccialli, "Scientific machine learning through physics-informed neural networks : where we are and what's next," *Journal of Scientific Computing*, vol. 92, n^o. 3, p. 88, 2022.
- [84] W. Ji, W. Qiu, Z. Shi, S. Pan et S. Deng, "Stiff-pinn : Physics-informed neural network for stiff chemical kinetics," *The Journal of Physical Chemistry A*, vol. 125, n^o. 36, p. 8098–8106, 2021.
- [85] K. Prantikos, L. H. Tsoukalas et A. Heifetz, "Physics-informed neural network solution of point kinetics equations for a nuclear reactor digital twin," *Energies*, vol. 15, n^o. 20, p. 7697, 2022.
- [86] A. G. Baydin, B. A. Pearlmutter, A. A. Radul et J. M. Siskind, "Automatic differentiation in machine learning : a survey," *Journal of Machine Learning Research*, vol. 18, p. 1–43, 2018.
- [87] S. I. Ngo et Y.-I. Lim, "Solution and parameter identification of a fixed-bed reactor model for catalytic co₂ methanation using physics-informed neural networks," *Catalysts*, vol. 11, n^o. 11, p. 1304, 2021.
- [88] A. J. Folayan, P. A. L. Anawe, A. E. Aladejare et A. O. Ayeni, "Experimental investigation of the effect of fatty acids configuration, chain length, branching and degree of unsaturation on biodiesel fuel properties obtained from lauric oils, high-oleic and high-linoleic vegetable oil biomass," *Energy Reports*, vol. 5, p. 793–806, 2019.
- [89] F. Qiu, Y. Li, D. Yang, X. Li et P. Sun, "Biodiesel production from mixed soybean oil and rapeseed oil," *Applied Energy*, vol. 88, n^o. 6, p. 2050–2055, 2011.
- [90] O. S. Stamenković, M. Lazić, Z. Todorović, V. Veljković et D. Skala, "The effect of agitation intensity on alkali-catalyzed methanolysis of sunflower oil," *Bioresource technology*, vol. 98, n^o. 14, p. 2688–2699, 2007.
- [91] D. Frascari, M. Zuccaro, A. Paglianti et D. Pinelli, "Optimization of mechanical agitation and evaluation of the mass-transfer resistance in the oil transesterification reaction

- for biodiesel production,” *Industrial & Engineering Chemistry Research*, vol. 48, n°. 16, p. 7540–7549, 2009.
- [92] D. Y. Leung, X. Wu et M. K. H. Leung, “A review on biodiesel production using catalyzed transesterification,” *Applied energy*, vol. 87, n°. 4, p. 1083–1095, 2010.
 - [93] G. Knothe, “Analyzing biodiesel : standards and other methods,” *Journal of the American Oil Chemists’ Society*, vol. 83, n°. 10, p. 823–833, 2006.
 - [94] E. C. for Standardization (CEN), “En 14105 : Fat and oil derivatives. fatty acid methyl esters (fame). determination of free and total glycerol and mono-, di-, triglyceride contents (reference method),” 2011.
 - [95] H. Laajimi, F. Galli, G. S. Patience et D. Schieppati, “Experimental methods in chemical engineering : Gas chromatography—gc,” *The Canadian Journal of Chemical Engineering*, vol. 100, n°. 11, p. 3123–3144, 2022.
 - [96] D. Zeng, R. Li, T. Jin et T. Fang, “Calculating the thermodynamic characteristics and chemical equilibrium of the stepwise transesterification of triolein using supercritical lower alcohols,” *Industrial & Engineering Chemistry Research*, vol. 53, n°. 17, p. 7209–7216, 2014.
 - [97] F. Trejo-Zárraga, F. de Jesús Hernández-Loyo, J. C. Chavarría-Hernández et R. Sotelo-Boyás, “Kinetics of transesterification processes for biodiesel production,” *Biofuels-State of development*, p. 149–179, 2018.
 - [98] M.-C. Hsiao, P.-H. Liao, N. V. Lan et S.-S. Hou, “Enhancement of biodiesel production from high-acid-value waste cooking oil via a microwave reactor using a homogeneous alkaline catalyst,” *Energies*, vol. 14, n°. 2, p. 437, 2021.
 - [99] B. Likozar et J. Levec, “Transesterification of canola, palm, peanut, soybean and sunflower oil with methanol, ethanol, isopropanol, butanol and tert-butanol to biodiesel : Modelling of chemical equilibrium, reaction kinetics and mass transfer based on fatty acid composition,” *Applied energy*, vol. 123, p. 108–120, 2014.
 - [100] S. Wang, Y. Teng et P. Perdikaris, “Understanding and mitigating gradient flow pathologies in physics-informed neural networks,” *SIAM Journal on Scientific Computing*, vol. 43, n°. 5, p. A3055–A3081, 2021.
 - [101] Y. Yang et P. Perdikaris, “Adversarial uncertainty quantification in physics-informed neural networks,” *Journal of Computational Physics*, vol. 394, p. 136–152, 2019.
 - [102] U. M. Ascher et L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. Siam, 1998, vol. 61.

ANNEXE A VÉRIFICATION DE CODE AVEC DES DONNÉES ARTIFICIELLES

Avant de se lancer dans l'implémentation du PINN avec les données du réacteur à micro-ondes, des tests préliminaires ont été faits avec des données artificielles afin de vérifier l'implémentation du code en PyTorch. Une réaction chimique fictive, comportant deux étapes et étant chacune réversible, a été posée :



avec A, B, C et D des espèces chimiques, k_1 et k_3 les constantes cinétiques des réactions directes ainsi que k_2 et k_4 les constantes des réactions inverses. Les constantes ont été posées à 1.5 s^{-1} pour k_1 , $0.5 \text{ M}^{-1} \text{ s}^{-1}$ pour k_2 , 1 s^{-1} pour k_3 et $0.1 \text{ M}^{-1} \text{ s}^{-1}$ pour k_4 .

L'objectif du PINN est d'approximer la concentration de chaque espèce selon le temps et d'ajuster les constantes k , a priori inconnues au PINN, afin que celles-ci respectent le plus possible les EDOs. Pour ce faire, les EDOs doivent être définies et les données d'entraînement doivent être générées.

En imposant un bilan de matières sur chacune des espèces chimiques, on obtient le système d'équations différentielles ordinaires suivant :

$$\frac{d[A]}{dt} = -k_1[A] + k_2[B][C] \quad (\text{A.3})$$

$$\frac{d[B]}{dt} = k_1[A] - k_2[B][C] \quad (\text{A.4})$$

$$\frac{d[C]}{dt} = k_1[A] - k_2[B][C] - k_3[C] + k_4[D] \quad (\text{A.5})$$

$$\frac{d[D]}{dt} = k_3[C] - k_4[D] \quad (\text{A.6})$$

Les données artificielles ont été générées par la méthode de Runge-Kutta, précise en temps au quatrième ordre. La méthode numérique résout le système d'EDO sur une plage de temps de 10 s. Ensuite, 6 données de concentration (incluant la condition initiale) de chaque espèce

ont été ajouté à l’entraînement du PINN. Ainsi, l’erreur entre ces 6 données et les prédictions faites par le PINN à 0, 2, 4, 6, 8 et 10 s est évaluée. La fonction de perte calcule également le résidu des EDOs sur 100 points de collocations, discrétisés uniformément sur le domaine de temps. Le PINN est finalement entraîné pendant 20 000 itérations, ce qui correspond à environ 1 minute d’exécution machine. Après l’entraînement, le PINN atteint une erreur quadratique moyenne de 2×10^{-7} .

La Figure A.1 montre l’évolution des concentrations de chaque espèce selon le temps prédite par le PINN. Le Tableau A.1 montre l’erreur relative entre les constantes cinétiques prédites par le PINN et leur valeur réelle imposée plus haut. Il est possible de conclure que le PINN est en mesure de retrouver les constantes cinétiques avec une erreur relative en deçà de 1%. Cela explique pourquoi les prédictions du PINN épousent quasi parfaitement la méthode numérique qui génère les données artificielles. Le code implémenté avec PyTorch permet de créer un algorithme de résolution d’équations différentielles ordinaires en plus d’un outil de régression non linéaire pour retrouver les paramètres inconnus.

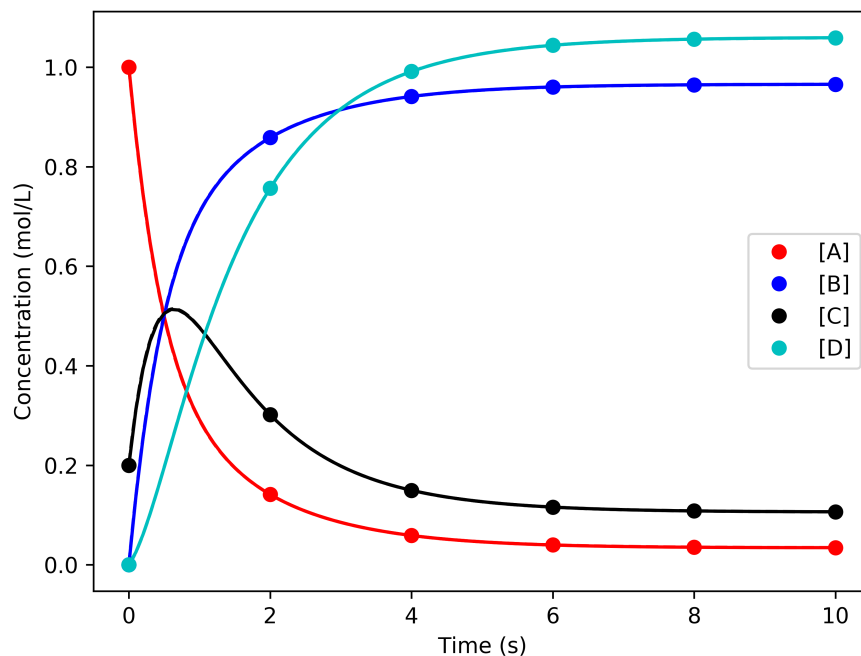


FIGURE A.1 Prédictions de la concentration des espèces chimiques par le PINN (—) comparées aux données artificielles (- - -) et aux données d’entraînement (●).

Cependant, dans la réalité, les données ne suivent jamais parfaitement la physique. Elles

TABLEAU A.1 – Erreur relative des prédictions des constantes cinétiques avec des données artificielles.

Constante	Erreur relative (%)
k_1	0.3
k_2	0.6
k_3	0.06
k_4	0.04

comportent, dans la très grande majorité du temps, des erreurs (ou du bruit). Ainsi, le PINN a été testé davantage avec les mêmes données, mais cette fois-ci bruitées par une erreur de 20% qui suit une loi normale. Le PINN a été entraîné de nouveau avec les données bruitées pendant 20 000 itérations et a atteint une erreur quadratique moyenne de 2×10^{-4} .

La Figure A.2 montre une nouvelle fois l'évolution des concentrations prédites par le PINN. Le Tableau A.2 montre l'erreur relative des constantes cinétiques. Malgré le bruit des données, le PINN reste en mesure de retrouver des vitesses de réaction semblables à la méthode numérique de référence. De plus, une erreur maximale de 16% sur les constantes cinétiques démontre la propagation de l'erreur des données, ce qui est attendu.

TABLEAU A.2 – Erreur relative des prédictions des constantes cinétiques avec des données artificielles bruitées.

Constante	Erreur relative (%)
k_1	1.0
k_2	2.9
k_3	8.9
k_4	16

Le PINN développé dans l'article II utilise le même code que celui vérifié dans cette présente section, à l'exception de quelques ajouts à l'architecture, de certains ajustements à l'entraînement et de l'utilisation de données expérimentales pour l'entraînement.

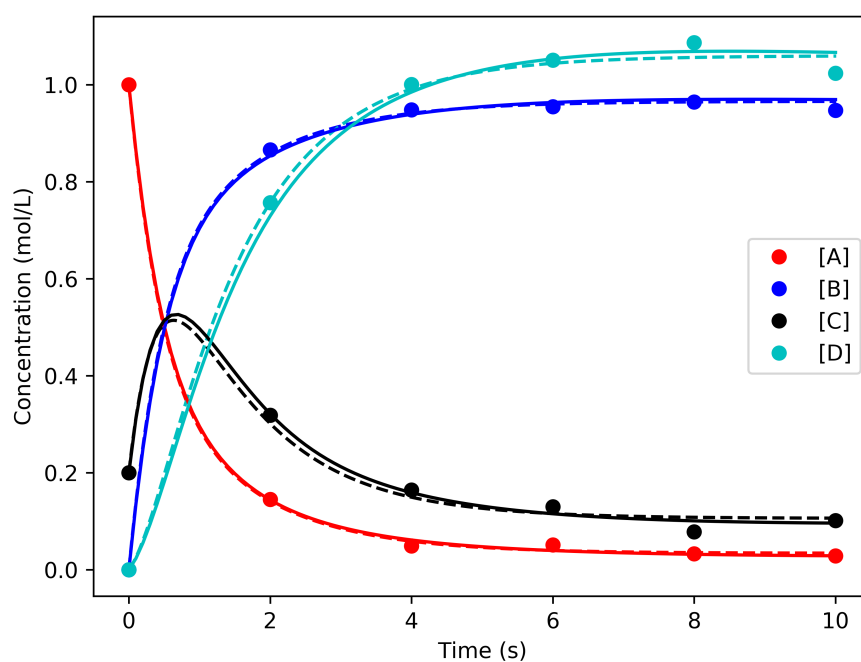


FIGURE A.2 Prédications de la concentration des espèces chimiques par le PINN (—) comparées aux données artificielles (- - -) et aux données d'entraînement bruitées (●).