공학박사학위논문

Computationally efficient modeling strategies for chemical reactor system

화학 반응기 시스템을 모델링하기 위한 계산 효율적인 전략

2022 년 2 월

서울대학교 대학원

화학생물공학부

최 솔 지

Computationally Efficient Modeling Strategies for Chemical Reactor System

지도교수 이종민

이 논문을 공학박사 학위논문으로 제출함
2022년 2월

서울대학교 대학원
화학생물공학부
최솔지

최솔지의 공학박사 학위논문을 인준함
2022년 2월

위원장 이원보 (인)

부위원장 이종민 (인)

위원 남재욱 (인)

위원 나종걸 (인)

위원 정동휘 (인)

Abstract

Computationally efficient modeling strategies for chemical reactor system

Solji Choi

School of Chemical & Biological Engineering

The Graduate School of Seoul National University

Keywords: High-fidelity model; Multi-scale modeling; Molecular weight distribution; LDPE; hybrid model; Method of Moment; Kinetic Monte Carlo; Physics-informed neural network; Surrogate model; Computational fluid dynamics; Chemical reactor

Student Number: 2015-22822

Many researchers in chemical engineering have been using analytical and computational models to predict the behaviors of systems and use these models to process optimization, design and control. However, until now, researchers are still forced to compromise on model fidelity and accuracy. Although high fidelity models can improve the model accuracy, simulating these models is usually time-consuming, making it difficult to perform optimization. In this thesis, computationally efficient strategies about two system are introduced which can maintain model fidelity but effectively reduce the calculation time.

Polymer reactor is selected for the first system and we focused on polymer kinetics. A hybrid approach that combines the method of moments and Monte Carlo simulation to predict the molecular weight distribution of low-density polyethylene for a continuous stirred tank reactor system is proposed. A 'Block,' which is repeating reaction group, is introduced for the calculation cost-effective simulation. This model called the 'block Kinetic Monte Carlo' is ~10 to 32 times faster than Neuhaus's model. The model can be applied to any steady state system and provide a calculation cost reduction effect, where one reaction is much faster than others; for example, the propagation reaction. Furthermore, we perform a case study on the effects of the system temperature and initiator concentration on the MWD and reaction rate ratio. Based on the simulation results of 180 case studies, we determine a quantitative guideline for the appearance of shoulder, which is a function of the rate ratio of reactions to the propagation reaction.

Computational fluid dynamics (CFD) based reactor system is selected for the second system. CFD is an essential tool for solving engineering problem that involves fluid dynamics. Especially in chemical engineering, fluid motion usually has extensive effects on system states such as temperature and component concentration. However, due to the critical issue of long computational times for simulating CFD, application of CFD is limited for many real-time problems such as real-time optimization and process control. In this study, we develop the surrogate model of Continuous stirred tank reactor

(CSTR) with Van de Vusse reaction using Physics-informed neural network (PINN) which can train the governing equations of system. We propose PINN architecture that can train every governing equation which chemical reactor system follows and can train multi-reference frame system. Also, we investigate that PINN can resolve the problem of neural network that needs lots of training data, are easily overfitted and cannot contain physical meaning. Furthermore, we modify the original PINN suggested by Raissi in order to solve the memory error and divergence problem with two methods: (1) Mini-batch training; (2) Weighted loss function. We also suggest a similarity based sampling strategy where the accuracy can be improved up to 5 times over the random sampling. This work can provide the guideline for developing the high performance surrogate model of chemical process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research motivation

With the development of computer and sensor technology, the demand for sophisticated control and optimization for high value-added industries is increasing. To achieve this, modeling the industrial level process with high fidelity is essential. However, until now, each unit was mainly modeling as 1 or 2-dimensional model to simulate the entire process. This is because the difference at the small scale can be neglected when simulating industrial level process, but it cannot be ignored in some cases. In particular, the chemical reactor which is one of the most important units in a chemical process the hydrodynamics of internal flow has a large effect on the system behavior in many cases. In a continuous stirred reactor (CSTR) with a stirrer inside, the performance of the reactor varies depending on how well the internal fluid is mixed. In the case of an autoclave reactor, several baffles are installed to divide the reactor into multi-zone and control the flow in order to easily adjust the properties of the product. Since many types of reactors simply cannot reflect their characteristics with 1 or 2 dimensional models, a flow model on a microscopic scale must be considered. However, as the modeling scale gets smaller, the simulation time increases, making it difficult to model industrial level processes and impossible to perform the optimization, control and design. For example, computational fluid dynamics (CFD), which is the only technique that can simulate the hydrodynamics, is difficult for industrial applications because the simulation time takes at least hours to months.

In order to solve this problem, various methodologies have been introduced to reduce the calculation time. One of the most representative method for simulating a chemical reactor is the multi-compartment model. It can significantly reduce the calculation time compared to solve the hydrodynamics (e.g. mass balance, momentum conservation, turbulence model) and kinetic model (e.g. heat conservation, species transport equation) together in mm or μm scale of mesh. This is a method of simplifying the flow by combining several meshes into one compartment and solving kinetic model separately. It allows the simulation of the reactor with a reasonable level of calculation time while considering the fluid flow information. However, the multi-compartment model is not suitable to perform optimization because it still takes an hour-level calculation time, but it requires a new simulation every time the operation condition changes. Another method is to develop a surrogate model that can be reproduced in a short time by learning the simulated results using a machine learning technique. Surrogate model can learn the features of the model by extracting the pattern from the simulation result, and has the advantage of being able to predict the untrained area. However, since most machine learning models are data-driven models that ignore the underlying laws of physics governing the system, it cannot be sure that the result at the untrained area is still physically meaningful.

This calculation time problem occurs frequently not only in hydrodynamics simulation, but also when simulating a kinetic model. For example, when the geometry of a molecule affects the reaction, such as polymer reaction, detail information about the molecule must be considered when simulating kinetics. In order to consider the geometry of the molecule, the order of arrangement of the atoms constituting the molecule must be simulated together, so the simulation scale become smaller in atomic unit and the simulation time is exponentially increased.

Modeling is performed at multiple scales, and it is often necessary to simulate at a microscopic level. However, it takes at least hours to days to perform the simulation with high fidelity. In this thesis, we introduce strategies to reduce the calculation time while maintaining the fidelity in modeling chemical reactor system.

1.2 Research objective

Among several computationally intensive modeling system, this thesis will cover a polymer kinetic model and a CFD based reactor model. For the polymer kinetic model, the target polymer is Low density polyethylene (LDPE), and its properties depend largely on the branches connected to the chain. For this reason, in order to accurately predict the characteristics of LDPE (e.g. Weight average molecular weight, Number average molecular weight, Molecular weight distribution, Polydispersity index , Melt index and etc), the geometry information of the polymer chain must be simulated together. A hybrid approach that combined stochastic approach and deterministic approach was used and a Kinetic Monte Carlo (KMC) was used as a stochastic approach, and a method of moment (MoM) technique was used as a deterministic approach. In order to reduce the calculation time, a new technique called Block KMC will introduced by grouping the reaction series into 7 independent repeating Block. In chapter 2, we will show detailed description of methodology, verification and case study result for predicting molecular weight distribution of LDPE in CSTR. In addition, an analysis of how the newly proposed model can shorten the calculation time compared to the existing method will introduced.

In chapter 3, more robust and precise surrogate model development strategy for CFD-based reactor model is introduced. CFD simulation is always a time consuming step in modeling process that hinders industrial application, so the surrogate model of CFD simulation is necessary when performing process optimization or finding optimal design. As the deep learning technique develops, the performance of learning by extracting patterns from data has improved, and the trained model can implement the original data in a very short time. However, the chemical process is governed by fundamental laws of physics such as mass balance, momentum conservation and heat balance. Most deep learning models ignore the physical laws and learn only patterns of data, so the prediction results in the untrained domain cannot be trusted. Ultimately, in order to perform process optimization, any condition must be simulated in real time. When the surrogate model is used, the prediction result in the domain that has not been learned is inevitably used, and in this domain it is still unsure whether the surrogate model satisfies the physical laws. In order to solve this problem, a physics informed neural network (PINN) that can learn not only the patterns of data but also the fundamental laws of physics was introduced. In the chapter 3, we will introduce the detailed procedure for developing a surrogate model for CSTR with Van de Vusse reaction using PINN. In addition, there will be an analysis of the performance difference between the PINN model and purely data driven model, and how fast the calculation time is compared to the CFD simulation.

1.3 Outline of the thesis

This thesis consists of four chapters. Chapter 1 explains research motivation, objective and thesis outline with associated publications to clarify the relative papers and co-authors. In Chapter 2 and Chapter 3, we introduce methodologies to reduce the calculation time for two different modeling systems, one is polymer kinetic modeling and the other is CFD based chemical reactor system. Chapter 4 deals with the contribution of each project, which summarizes the results and their potential to be applied to other projects and systems. Future work is also suggested to intrigue other researchers and provide a guideline for their further research on the topic treated in this thesis.

Chapter 2

Molecular weight distribution modeling of LDPE in a continuous stirred-tank reactor using coupled deterministic and stochastic approach

2.1 Introduction

Low-density polyethylene (LDPE) is produced under a high-pressure by free-radical polymerization. Unlike high-density polyethylene (HDPE), which has scarce branching, LDPE has high-density long-chain branches owing to the scission and chain transfer reactions. Because of its features, LDPE has a low tensile strength and high ductility, which improves flexibility and transparency. Therefore, LDPE can form various shapes and produce most films and plastic bags.

There are several factors that represent the physical properties of polymers, such as Mn (number average molecular weight), Mw (weight average molecular weight), MI (melt index), molecular weight distribution (MWD), and polydispersity index (PDI). Other properties only represent the average value; whereas, MWD shows the weight ratio of each polymer component. The shape of the MWD allows us to predict the polymer processability and product quality. For example, a narrow MWD enhances the polymer quality and a wide MWD allows for easier processing. The shoulders and tails of the MWD contain information about the density of chain branching. Short chain branching reduces the density of polymer[1] and long-chain branching increases the sensitivity of melt viscosity. [2] To produce a polymer with the desired physical properties and improve the process productivity, an MWD prediction model is essential.

Many studies have been conducted to predict the MWD. MoM is the most popular method for predicting the average molecular weight.[3]–[6] This method has been widely used by many research groups to develop kinetic models for conventional free radical polymerization, coordination polymerization and others for linear and branched polymers.[7]–[11] One of the main advantages of the MoM is that it simplifies the theoretically infinite number of mass balance equations of chain species into a much smaller set of equations. In this method, information about individual chains is ignored for simplification, so only the average properties of the polymer can be calculated. By dividing an entire range of molecular weights into a limited number of intervals, the method of moments is capable of predicting molecular-weight distributions.[12] However, a large number of moments are required to obtain a detailed information on MWD. Well and Ray segregated the polymer distribution into a large number of 'segments,' so that the model can predict the detailed shape of MWD and reveal the kinetic effects for the existence of MWD shoulder.[12] Because the model requires a minimum of 75 segments, at least 400 variables should be considered, which makes it difficult to find a solution. Another approach suggested by Pladis and Kiparissides is based on the 'classes,' in which the total polymer population is discretized according to the number of branches.[13] They assumed that the distribution of classes followed a log-normal distribution, which is not justified. In addition, MoM cannot explicitly implement scission. In an environment where the production conditions exceed 300 °C, the chain scission becomes dominant owing to the high-temperature sensitivity. [14] Because the molecules split differently depending on their topology, moments including scission points must be introduced to simulate the scission. In this case, the number of moments increase exponentially, making the problem difficult to solve. Therefore, in most studies on MoM, a random scission mechanism that does not require the chain topology information has been adopted. In addition, the MoM model requires a closure relationship assumption[15] to solve the scission. The model error accumulates due to the ambiguous implementation of the chain scission.

In contrast to MoM, the Monte Carlo (MC) technique can accurately implement scission based on the topological information. Numerous studies have been conducted to investigate the effect of chain scission on the physical properties of polymers using the MC technique. [16]–[24] Gillespie proposed the Kinetic Monte Carlo (KMC) algorithm, which has been applied to many other studies. [25] However, this method cannot be directly applied to continuous reactors, in which chains with different residence times exist. Tobita first proposed an individually branched polymer chain methodology. He introduced the concept of a primary chain to which all branch points are connected and assumed that the primary chain is severed. [26] It can simulate chains with different residence times and predict the MWD in various types of reactors, such as continuous stirred tank reactors (CSTRs), tubular reactors, and tanks-in-series reactors. [27], [28] However, this approach requires the prior knowledge of the branch density and segment length following the Flory distribution. [29] Moreover, the MC simulation has an issue of exorbitant computational time, which is not suitable for industrial applications.

Studies have been conducted to reduce the computation time by combining deterministic methods with stochastic techniques. [18], [30]–[32] Meimaroglou proposed a kinetic/topology MC algorithm for a multizonal tubular reactor. [18] The topological information of the macromolecule was obtained by the MC algorithm with the kinetic model of the tubular reactor. This method does not require any assumptions about the distributions of live and dead polymers, but the model cannot be extended to CSTR because all macromolecules have the same residence time. [33] Neuhaus developed a hybrid model that coupled deterministic and stochastic approaches that could be applied to the tubular and autoclave reactors. [32] The single macromolecule approach (SMMA) introduced in the model allows the parallel computation of large molecule ensembles and reduces the computation time efficiently. In Eckes's study[32], polymer chains that are combined with the macromolecules via termination by combination are sampled from the previously stored chain data. Although this approach is valid for tubular reactors, it encounters a recursion in the CSTR reactor. This is because the chain to be combined may have a longer residence time than the macromolecules. In addition, to achieve sufficient accuracy, 10 million molecules must be simulated. This requires approximately 1 h even with parallelized computation, which is not practical for industrial applications. [30]

Both the approaches have technical issues, such as MoM cannot implement the topological scission, and MC is computationally intensive. Therefore, this study proposes a hybrid model approach consisting of the MoM and KMC for a CSTR. The proposed model can implement the scission, resolve the recursion, and significantly reduce the calculation time. The model is validated with the model proposed by Neuhaus[30] by comparing its performance and calculation time. In addition, we extensively explored the MWD under various operating conditions and analyzed the effect of each element reaction. Given this, we can determine the effect of each reaction mechanism on the existence of the shoulder in MWD.

2.2 Methodology

To simulate polymerization in a steady-state reactor, such as CSTR, we developed a model that combined MoM and KMC. First, the bulk polymer concentration was calculated using a mass balance and moment equation. Then, the detailed molecular weight distribution was derived using KMC. The newly proposed model is based on the 'Single Macromolecule Approach' (SMMA) and we named the single macromolecule as 'target chain'. The concept of the target chain model is illustrated in Figure 2.1.

2.2.1 Polymer reaction mechanism

LDPE is produced by free-radical polymerization, and the elementary reactions are presented in Table 2.1. L_n represents a live polymer chain of length n, whose radical is located at the end of the chain. $L_{n,sec}$ represents a secondary radical polymer chain of length n, whose radical is located in the middle of the chain. Finally, D_n represents a dead polymer chain of length n.

To avoid any confusion, we use different reaction names for the same class of reactions depending on the type of molecule that we track. In the case of the chain transfer to polymer, when the target chain is a live polymer, we name it $R_{cp,1}$ while the reaction is named $R_{cp,2}$ when the target chain is a dead polymer. Sometimes, the chain transfer to solvent or chain transfer agent (CTA) and backbiting reactions are also involved in free radical polymerization. [34] However, the effect of the backbiting reaction on the molecular weight distribution is negligible [35], and the chain transfer to solvent or CTA has an effect similar to that of the chain transfer to the monomer. Therefore, these two reactions are not considered in this study. The most important reaction for LDPE to have high-density branches is the chain transfer to polymer. As shown in Figure 2.2, a dead polymer becomes a secondary radical chain by obtaining a radical from a live polymer chain. Then, the propagation and beta scission occur sequentially in the secondary radical chain. When propagation occurs, the chain lengthens and forms long-chain branches. When beta scission occurs, the chain is split into two smaller molecules. Because of these two reactions, the MWD broadens. The beta scission includes two types of pathways: random scission and chain-end scission. Among these, the random scission is selected because it is more dominant in the simulated reaction. The kinetic parameters of each reaction are listed in Table 2.2.

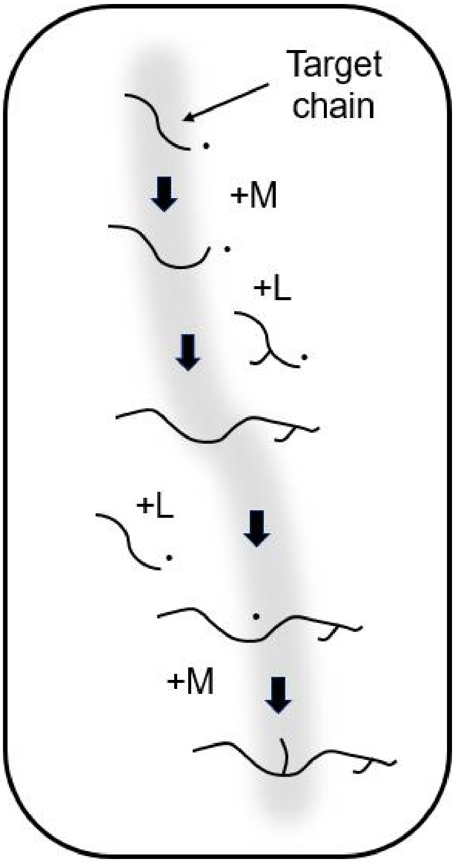


Figure 2.1 Concept of target chain model.

Table 2.1 Elementary reactions of free radical polymerization.

R_i	Initiation	$I \xrightarrow{f*k_d} 2L_0$
$R_{p,1}$	Propagation	$L_m\left(L_{m,sec}\right) + M \xrightarrow{k_p} L_{m+1}$
$R_{p,2}$		
R_t	Termination	$L_m + L_n \xrightarrow{k_t} D_{m+n}$
$R_{cp,1}$	Chain transfer to polymer	$L_m + D_n \xrightarrow{k_{cp}} D_m + L_{n,sec}$
$R_{cp,2}$		
R_{cm}	Chain transfer to monomer	$L_m + M \xrightarrow{k_{cm}} D_m + L_0$
$R_{beta,1}$	Beta scission	$L_{m,sec} \xrightarrow{k_{beta}} L_n + D_{m-n}$
$R_{beta,2}$		

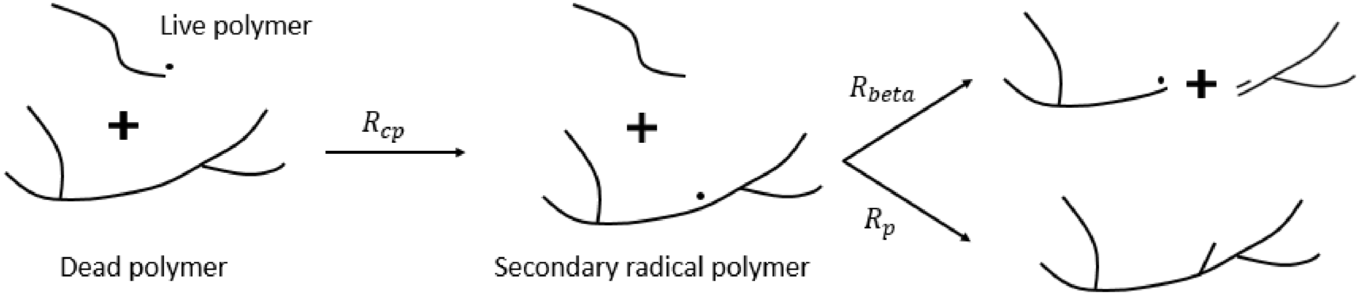


Figure 2.2 Schematic of chain transfer to polymer reaction.

Table 2.2 Kinetic parameters of LDPE.

$$k = A \cdot \exp\left(-\frac{E + P * \Delta V}{RT}\right)$$

	$A(L/mol \cdot s)$	$E/R(K)$	$\Delta V(cm^3/mol)$
k_d	$1.06 * 10^{16}$	17972	2.5077
k_p	$1.14 * 10^{7}$	3584	-19.7717
k_t	$3 * 10^{9}$	1213	13.0443
k_{cp}	$2.92 * 10^{5}$	5580	-20.06
k_{cm}	$1.67 * 10^{6}$	4549	4.3937
k_{beta}	$1.04 * 10^{12}$	10108	-19.27

2.2.2 Reactor model

The most common reactor type for LDPE polymerization in industry is the autoclave reactor, but it is inappropriate for analysis with the proposed model because of the complex fluid dynamics and operation. Therefore, in this research, a model is developed in a series of CSTRs, each of which has one input and one output, and exhibits homogeneous temperatures and pressures. Because the autoclave reactor shows a similar behavior to the system composed of a series of CSTRs, the model can be extended to an autoclave reactor in practice.

The model consists of both deterministic and stochastic parts. In deterministic part, the bulk polymer properties are calculated using MoM. Then, the calculated values are passed on to the stochastic part and are used to calculate the probability of each reaction. Based on these probabilities, the polymer chain length is determined stochastically using the MC technique.

2.2.3 Deterministic part

The steady-state information of the polymer properties in CSTR can be calculated using a set of equations involving the momentum equation and mass balances. The equations for the initiator, monomer, live polymer, dead polymer, and secondary live polymer are presented below:

$$\frac{d[M]}{dt} = \frac{v}{V}[M]_{in} - k_p\lambda_0[M] - k_p\lambda_{sec,0}[M] - k_{cm}\lambda_0[M] - \frac{v}{V}[M] \quad (2\text{-}1)$$

$$\frac{d[I]}{dt} = \frac{v}{V}[I]_{in} - fk_d[I] - \frac{v}{V}[I] \tag{2-2}$$

$$\begin{aligned}\frac{d[L_n]}{dt} \\ &= \delta(n-1)(2fk_d[I] + k_{cm}\lambda_0[M]) \\ &- k_p\big([L_n] - [L_{n-1}] - [L_{sec,n}]\big)[M] - \mathrm{k_t}[L_n]\lambda_0 - k_{cp}[L_n]\mu_1 \\ &- k_{cm}[L_n][M] + k_{beta}\sum_{m=n+1}^{\infty}[\lambda_{sec,m}] - \frac{v}{V}[L_n]\end{aligned} \tag{2-3}$$

$$\begin{aligned}\frac{\mathrm{d}[D_n]}{dt} \\ &= k_t\sum_{m=1}^{n-1}[L_m][L_{n-m}] - k_{cp}\lambda_0 n[D_n] + k_{cp}[L_n]\mu_1 + k_{cm}[L_n][M] \\ &+ k_{beta}\sum_{m=n+1}^{\infty}[\lambda_{sec,m}] - \frac{v}{V}[D_n]\end{aligned} \tag{2-4}$$

$$\begin{aligned}\frac{\mathrm{d}[L_{sec,n}]}{dt} \\ &= -k_p[L_{sec,n}][M] + k_{cp}\lambda_0 n[D_n] - k_{beta}[L_{sec,n}] - \frac{v}{V}[L_{sec,n}]\end{aligned} \tag{2-5}$$

From these equations, the information on the polymer weight distribution is summed up with a couple of moments. From the definition of moment, the 0th, 1st, and 2nd moments of the live polymer, dead polymer, and secondary radical polymer are calculated as in Eqs (2-6)–(2-8).

$$\lambda_{\mathrm{k}} = \sum n^k [L_n] \tag{2-6}$$

$$\mu_{\mathrm{k}} = \sum n^k [D_n] \tag{2-7}$$

$$\lambda_{\mathrm{sec,k}} = \sum n^k [L_{sec,n}] \tag{2-8}$$

Each moment can be derived from Eqs. (2-1)–(2-8) with pseudo-steady state assumption, and the derived equations are given by:

$$\frac{d\lambda_0}{dt} = 2fk_d[I] + k_p\lambda_{sec,0}[M] - k_t\lambda_0^2 - k_{cp}\lambda_0\mu_1 + k_{beta}\lambda_{sec,0} - \frac{v}{V}\lambda_0 \tag{2-9}$$

$$\frac{d\lambda_1}{dt} = 2fk_d[I] + \left(k_{cm} + k_p\right)\lambda_0[M] - k_t\lambda_1\lambda_0 + k_p(\lambda_{sec,0} + \lambda_{sec,1}) - k_{cp}\lambda_1\mu_1 - k_{cm}\lambda_1[M] + k_{beta}\frac{(\lambda_0 + \lambda_1)}{2} - \frac{v}{V}\lambda_1 \tag{2-10}$$

$$\frac{d\lambda_2}{dt} = 2fk_d[I] + k_{cm}\lambda_0[M] + k_p[M](2\lambda_1 + \lambda_0) + k_p\left(\lambda_{sec,2} + 2\lambda_{sec,1} + \lambda_{sec,0}\right)[M] - k_t\lambda_2\lambda_0 - k_{cp}\lambda_2\mu_1 - k_{cm}\lambda_2[M] + k_{beta}\frac{(2\lambda_2 + 3\lambda_1 + \lambda_0)}{6} - \frac{v}{V}\lambda_2 \tag{2-11}$$

$$\frac{d\mu_0}{dt} = \frac{k_t}{2}\lambda_0^2 + k_{cm}\lambda_0[M] + k_{beta}\lambda_{sec,0} - \frac{v}{V}\mu_0 \tag{2-12}$$

$$\frac{d\mu_1}{dt} = k_t(\lambda_0\lambda_1) - k_{cp}\mu_2\lambda_0 + k_{cp}\lambda_1\mu_1 + k_{cm}\lambda_1[M] + k_{beta}\frac{\left(\lambda_{sec,0} + \lambda_{sec,1}\right)}{2} - \frac{v}{V}\mu_1 \tag{2-13}$$

$$\frac{d\mu_2}{dt} = k_t(\lambda_0\lambda_2 + \lambda_1^2) - k_{cp}\mu_3\lambda_0 + k_{cp}\lambda_2\mu_1 + k_{cm}\lambda_2[M] + k_{beta}\frac{\left(2\lambda_{sec,2} + 3\lambda_{sec,1} + \lambda_{sec,0}\right)}{6} - \frac{v}{V}\mu_2 \tag{2-14}$$

$$\frac{d\lambda_{sec,0}}{dt} = -k_p\lambda_{sec,0}[M] + k_{cp}\lambda_0\mu_1 - k_{beta}\lambda_{sec,0} - \frac{v}{V}\lambda_{sec,0} \tag{2-15}$$

$$\frac{d\lambda_{sec,1}}{dt} = -k_p\lambda_{sec,1}[M] + k_{cp}\lambda_0\mu_2 - k_{beta}\lambda_{sec,1} - \frac{v}{V}\lambda_{sec,1} \tag{2-16}$$

$$\frac{d\lambda_{sec,2}}{dt} = -k_p\lambda_{sec,2}[M] + k_{cp}\lambda_0\mu_3 - k_{beta}\lambda_{sec,2} - \frac{v}{V}\lambda_{sec,2} \qquad (2\text{-}17)$$

Given these equations, the problems cannot be closed because of μ_3. Therefore, μ_3 is obtained using the closure technique suggested by Zabisky. [36]

$$\mu_3 = \left(\frac{\mu_2}{\mu_1}\right)^2 \sqrt{2\mu_2\mu_0 - \mu_1^2} \qquad (2\text{-}18)$$

2.2.4 Stochastic part

In this section, we introduce a new stochastic algorithm called 'Block KMC'. This method is based on the conventional KMC, and dramatically reduces the calculation time by introducing the concept of 'Block'. Based on the steady-state assumption, each algorithm step can be simultaneously simulated in block units, and the calculation time can be reduced because the propagation reaction is dominant compared to other reaction components. The detailed algorithm is described in the following subsections.

2.2.4.1 KMC

The typical algorithm of KMC was proposed by Gillespie, and a flowchart is shown in Figure 2.3. [25]

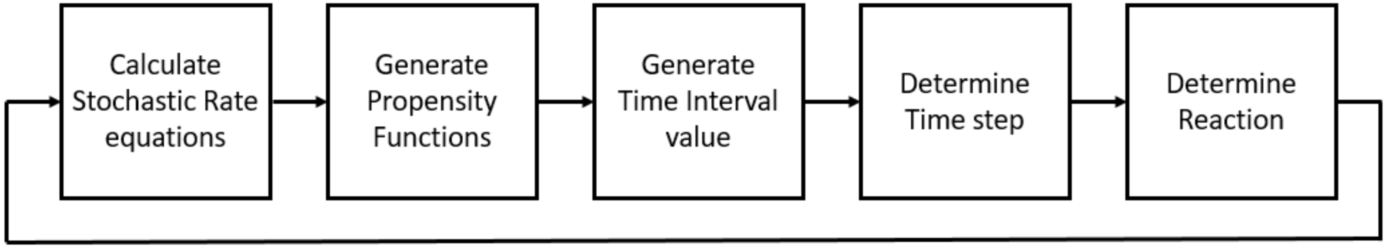

Calculate Stochastic Rate equations
Generate Propensity Functions
Generate Time Interval value
Determine Time step
Determine Reaction

Figure 2.3 Flow chart of KMC algorithm proposed by Gillespie.

First, the stochastic rates are calculated based on the bulk polymer concentrations, and propensity functions are generated. Then, one reaction is selected with a random number between 0 and 1 based on these probabilities, and the state is updated with this reaction. Subsequently, the time is updated by marching forward over a single time interval. The simulation is repeated until the total time reaches the residence time.

It is difficult to directly apply this algorithm to our model because each polymer chain has a different residence time in the CSTR. Therefore, we employ the concept of a target chain and combine KMC with MoM. The detailed algorithm of the target chain model is shown in Figure 2.4.

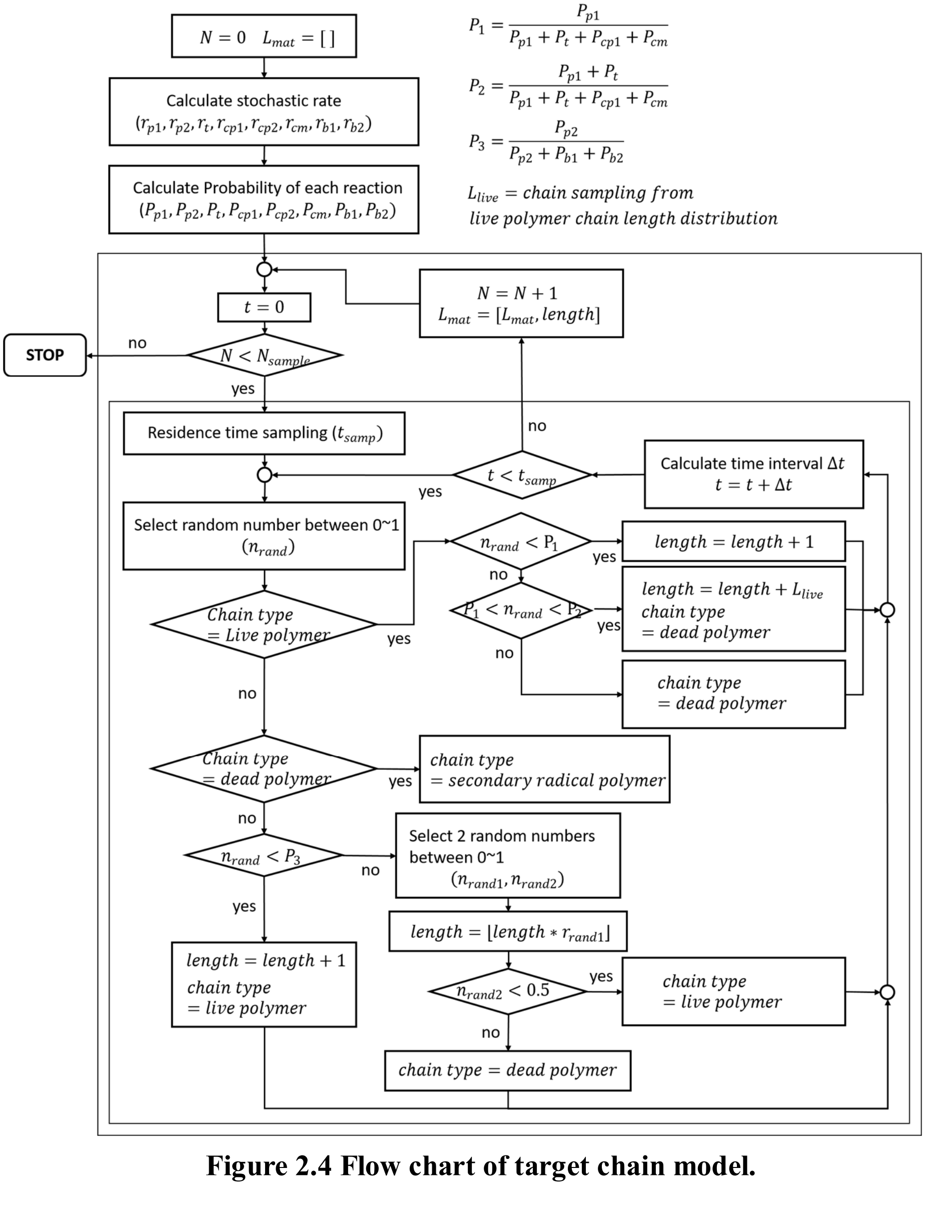

$N = 0 \quad L_{mat} = [\,]$
Calculate stochastic rate
$(r_{p1}, r_{p2}, r_t, r_{cp1}, r_{cp2}, r_{cm}, r_{b1}, r_{b2})$
Calculate Probability of each reaction
$(P_{p1}, P_{p2}, P_t, P_{cp1}, P_{cp2}, P_{cm}, P_{b1}, P_{b2})$
$P_1 = \frac{P_{p1}}{P_{p1} + P_t + P_{cp1} + P_{cm}}$
$P_2 = \frac{P_{p1} + P_t}{P_{p1} + P_t + P_{cp1} + P_{cm}}$
$P_3 = \frac{P_{p2}}{P_{p2} + P_{b1} + P_{b2}}$
$L_{live} = chain\ sampling\ from$
$live\ polymer\ chain\ length\ distribution$
$t = 0$
$N = N + 1$
$L_{mat} = [L_{mat}, length]$
STOP
no
$N < N_{sample}$
yes
no
Residence time sampling (t_{samp})
$t < t_{samp}$
yes
Calculate time interval Δt
$t = t + \Delta t$
Select random number between 0~1
(n_{rand})
$n_{rand} < P_1$
yes
$length = length + 1$
no
$P_1 < n_{rand} < P_2$
yes
$length = length + L_{live}$
$chain\ type$
$= dead\ polymer$
$Chain\ type$
$= Live\ polymer$
yes
no
$chain\ type$
$= dead\ polymer$
no
$Chain\ type$
$= dead\ polymer$
yes
$chain\ type$
$= secondary\ radical\ polymer$
no
$n_{rand} < P_3$
no
Select 2 random numbers
between 0~1
(n_{rand1}, n_{rand2})
yes
$length = \lfloor length * r_{rand1} \rfloor$
$length = length + 1$
$chain\ type$
$= live\ polymer$
$n_{rand2} < 0.5$
yes
$chain\ type$
$= live\ polymer$
no
$chain\ type = dead\ polymer$

Figure 2.4 Flow chart of target chain model.

The residence time distribution in CSTR is known as:

$$E(\theta) = \frac{1}{T} exp(-\theta) \tag{2-19}$$

$$\theta = \frac{\tau}{T} \tag{2-20}$$

The stochastic rates of each reaction are given as:

$$r = k \; C_a C_b \quad \rightarrow \quad r = k_{stochastic} \; N_a N_b, \tag{2-21}$$

$$k_{stochastic} = k \quad \text{for monomolecular reaction} \tag{2-22}$$

$$k_{stochastic} = {}^{k}\!/_{VN_{Avo}} \quad \text{for bimolecular reaction between similar species} \tag{2-23}$$

$$k_{stochastic} = {}^{2k}\!/_{VN_{Avo}} \quad \text{for bimolecular reaction between different species} \tag{2-24}$$

$$r_i = 2k_i \; N_{initiator} \tag{2-25}$$

$$r_p = {}^{2k_p}\!/_{VN_{Avo}} \; N_L N_M \tag{2-26}$$

$$r_t = {}^{2k_{tc}}/_{VN_{Avo}} \quad N_L N_L \tag{2-27}$$

$$r_{cm} = {}^{2k_{cm}}/_{VN_{Avo}} \quad N_L N_M \tag{2-28}$$

$$r_{cp} = {}^{2k_{cp}}/_{VN_{Avo}} \quad N_L N_D \tag{2-29}$$

$$r_b = {}^{2k_{beta}}/_{VN_{Avo}} \quad N_{L_{sec}} \tag{2-30}$$

The same type of reaction may have different stochastic rates depending on the type of chain. For example, the chain transfer to polymer can occur when the target chain is a live or dead polymer. The number of possible radical sites is the total length of all the dead polymer chains when the target chain is a live polymer. Hence, N_D becomes μ_1/λ_0. The number of possible radical sites is the length of the target chain when the target chain is a dead polymer because the target chain receives a radical from the surrounding live chain. Therefore, the probability of each reaction is determined, as described in Table 2.3.

Table 2.3 List of possible reactions depending on the chain type and their probability.

Chain type	Possible Reaction	Probability
L (Live Polymer)	• $R_{p,1}$	• $P_{p,1} = \frac{r_{p,1}}{r_{p,1}+r_t+r_{cp,1}+r_{cm}}$
	• R_t	• $P_t = \frac{r_t}{r_{p,1}+r_t+r_{cp,1}+r_{cm}}$
	• $R_{cp,1}$	• $P_{cp,1} = \frac{r_{cp,1}}{r_{p,1}+r_t+r_{cp,1}+r_{cm}}$
	• R_{cm}	• $P_{cm} = \frac{r_{cm}}{r_{p,1}+r_t+r_{cp,1}+r_{cm}}$
D (Dead Polymer)	• $R_{cp,2}$	• $P_{cp,2} = 1$
L_{sec} (Secondary radical Polymer)	• $R_{p,2}$	• $P_{p,2} = \frac{r_{p,2}}{r_{p,2}+r_{b,1}+r_{b,2}}$
	• $R_{beta,1}$	• $P_{b,1} = \frac{r_{b,1}}{r_{p,2}+r_{b,1}+r_{b,2}}$
	• $R_{beta,2}$	• $P_{b,2} = \frac{r_{b,2}}{r_{p,2}+r_{b,1}+r_{b,2}}$

The reaction is determined based on the probabilities with random sampling, and the target chain is updated with this reaction. The only reactions that change the chain length are propagation and termination by combination. In the case of propagation, the length is increased by one, and in the case of termination by combination, the length is increased by the length of the surrounding live polymer. The molecular weight distribution of live polymers is assumed to follow the Schultz-Zimm form, which is the simplest form that can represent the bimodal distribution. [36] Parameters y and h are inversely estimated from $[L]$ and λ_1. Therefore, the live polymer chain that is combined with the target chain is sampled from the Scultz-Zimm distribution of the live polymer chain.

$$\bar{W}(n) = \frac{ny}{h} y^h n^{h-1} \exp(-yn) / \Gamma(h) \tag{2-39}$$

Time interval value is calculated by the reaction rate and random sampling. [25]

$$\Delta \mathrm{t} = \mathrm{Q}_{\mathrm{k}}^{-1} \cdot \ln(\frac{1}{n_{rand}}) \tag{2-40}$$

$$\mathrm{Q}_{\mathrm{live\ polymer}} = R_{p,1} + R_t + R_{cp,1} + \mathrm{R}_{\mathrm{cm}} \tag{2-41}$$

$$\mathrm{Q}_{\mathrm{dead\ polymer}} = R_{cp,2} \tag{2-42}$$

$$\mathrm{Q}_{\mathrm{secondary\ radical\ polymer}} = R_{p,2} + R_{beta,1} + R_{beta,2} \tag{2-43}$$

2.2.4.2 Block KMC

The chain grown by the KMC algorithm has a certain pattern in the series of reactions. Because the possible reaction types vary depending on the chain type, seven types of blocks are repeated. These blocks are found by binding the reactions, whose chain type is reverted to the live polymer. The types of blocks and their appearance probabilities are listed in Table 2.4. A randomly selected sample from the series of reactions is shown in Figure 2.5. These reaction series can be combined into the blocks listed in Table 2.4. Each block is independent, and the appearance probability is constant. As, the number of blocks follows a binomial distribution, it is possible to directly determine the number of blocks in the whole series. The algorithm is shown in Figure 2.6.

Table 2.4 Representation of Blocks and their appearance probabilities.

Block	Chain type	Probability
B1) $\mathrm{R}_{\mathrm{p},1}$	$L \rightarrow L$	$P_{p,1} = \dfrac{r_{p,1}}{r_{p,1} + r_t + r_{cp,1} + r_{cm}}$
B2 − 1) $R_{\mathrm{t}} - \left(R_{cp,2} - R_{beta,2}\right)^n - R_{cp,2} - R_{p,2}$ B2 − 2) $R_{\mathrm{t}} - \left(R_{cp,2} - R_{beta,2}\right)^n - R_{cp,2} - R_{beta,1}$	$L \rightarrow (D \rightarrow L_{sec})^n \rightarrow L$	$P_t = \dfrac{r_t}{r_{p,1} + r_t + r_{cp,1} + r_{cm}}$
B3 − 1) $R_{\mathrm{cp},1} - \left(R_{cp,2} - R_{beta,2}\right)^n - R_{cp,2} - R_{p,2}$ B3 − 2) $R_{\mathrm{cp},1} - \left(R_{cp,2} - R_{beta,2}\right)^n - R_{cp,2} - R_{beta,1}$	$L \rightarrow (D \rightarrow L_{sec})^n \rightarrow L$	$P_{cp,1} = \dfrac{r_{cp,1}}{r_{p,1} + r_t + r_{cp,1} + r_{cm}}$
B4 − 1) $R_{\mathrm{cm}} - \left(R_{cp,2} - R_{beta,2}\right)^n - R_{cp,2} - R_{p,2}$ B4 − 2) $R_{\mathrm{cm}} - \left(R_{cp,2} - R_{beta,2}\right)^n - R_{cp,2} - R_{beta,1}$	$L \rightarrow (D \rightarrow L_{sec})^n \rightarrow L$	$P_{cm} = \dfrac{r_{cm}}{r_{p,1} + r_t + r_{cp,1} + r_{cm}}$

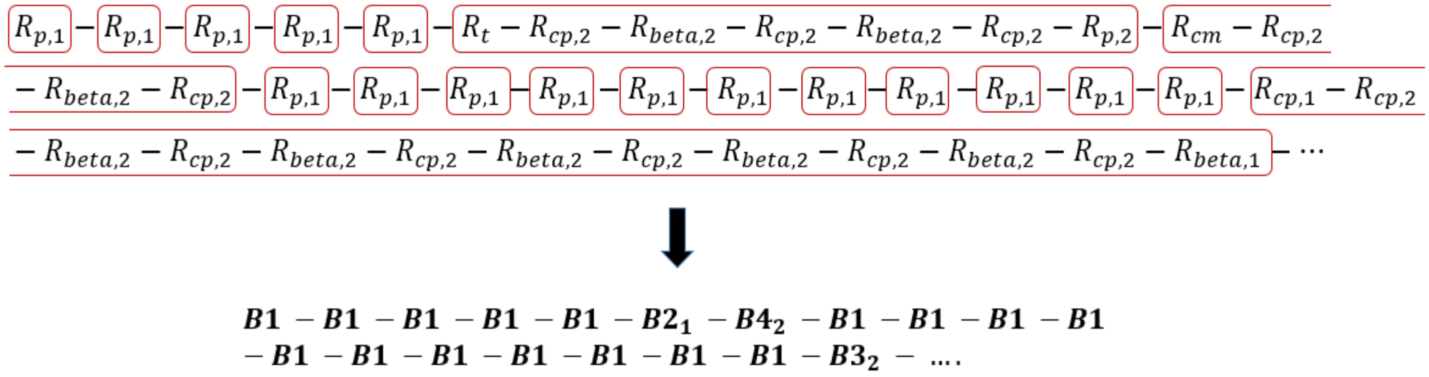

$R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_t - R_{cp,2} - R_{beta,2} - R_{cp,2} - R_{beta,2} - R_{cp,2} - R_{p,2} - R_{cm} - R_{cp,2}$
$- R_{beta,2} - R_{cp,2} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{p,1} - R_{cp,1} - R_{cp,2}$
$- R_{beta,2} - R_{cp,2} - R_{beta,2} - R_{cp,2} - R_{beta,2} - R_{cp,2} - R_{beta,2} - R_{cp,2} - R_{beta,2} - R_{cp,2} - R_{beta,1} - \cdots$
$\boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B2_1} - \boldsymbol{B4_2} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1}$
$- \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B1} - \boldsymbol{B3_2} - \ldots.$

Figure 2.5 Example of reaction series and the series replaced by Block.

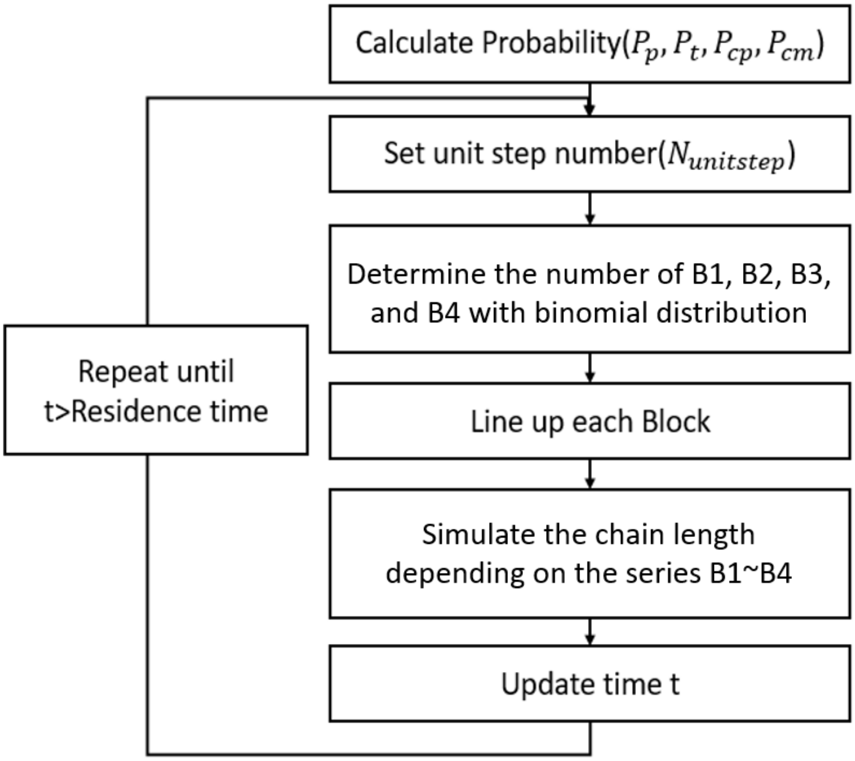

Calculate Probability(P_p, P_t, P_{cp}, P_{cm})
Set unit step number($N_{unitstep}$)
Determine the number of B1, B2, B3, and B4 with binomial distribution
Line up each Block
Simulate the chain length depending on the series B1~B4
Update time t
Repeat until t>Residence time

Figure 2.6 Flow chart of Block KMC algorithm.

.

To determine the number of blocks, we need to know the total number of blocks. However, it is impossible to know the total number of blocks until the end of simulation, because the time interval depends on random numbers. Therefore, we introduce a unit step number to determine the number of blocks. The simulation is repeated by the unit step number until the time reaches the total residence time of the target chain. In this system, B1 is dominantly selected because the propagation is much faster than other reactions. This means that most of the unit step number is the number of B1 blocks. Because of these features, we can simulate much faster than the conventional MC algorithm.

To calculate the length of the target chain, the number of blocks and the order of blocks must be determined. The algorithm used to set the order of blocks in the target chain model is as follows:

1) Generate N_{B2}+N_{B3}+N_{B4}+1 random numbers.

2) Normalize the random numbers so that the sum becomes 1, and multiply each number by N_{B1}.

3) Determine the position of remaining Blocks that will come to the site between the B1 groups based on their probabilities.

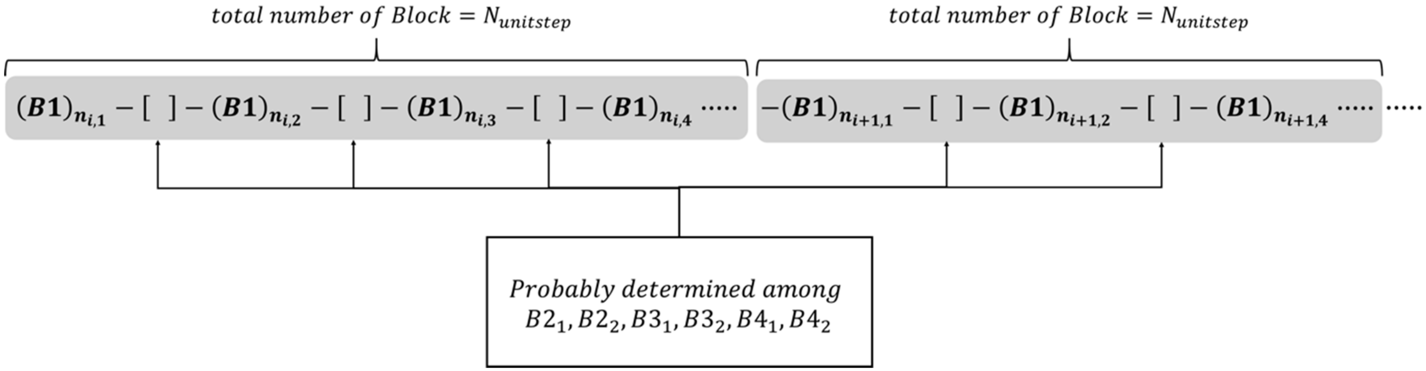

$total\ number\ of\ Block = N_{unitstep}$
$total\ number\ of\ Block = N_{unitstep}$
$(\boldsymbol{B1})_{n_{i,1}} - [\ \ ] - (\boldsymbol{B1})_{n_{i,2}} - [\ \ ] - (\boldsymbol{B1})_{n_{i,3}} - [\ \ ] - (\boldsymbol{B1})_{n_{i,4}} \cdots\cdots$
$-(\boldsymbol{B1})_{n_{i+1,1}} - [\ \ ] - (\boldsymbol{B1})_{n_{i+1,2}} - [\ \ ] - (\boldsymbol{B1})_{n_{i+1,4}} \cdots\cdots \cdots\cdots$
$Probably\ determined\ among$
$B2_1, B2_2, B3_1, B3_2, B4_1, B4_2$

Figure 2.7 Representation of Block series derived from Block KMC method.

Because the number of B1 is much greater than that of the other blocks, the series of blocks takes the form of the remaining blocks between the B1 groups, as presented in Figure 2.8. By the first and second stages of the algorithm, we can determine the number of B1 blocks belonging to the B1 groups, which are represented as $n_{i,1}, n_{i,2}, n_{i,3}, n_{i,4}, \ldots$ in Figure 2.7. In the third stage, the remaining blocks are placed between the B1 groups.

Let us assume that t_1 is the average computation time per step, which consists of random sampling and updating the chain length. The conventional KMC technique requires approximately $N \cdot t_1$ computation time to simulate a total of N reaction series. In the Block KMC model, when the number of propagation reactions is N_p, the first and second stages take $(N - N_p + 1) \cdot t_1$, and the third stage takes approximately $(N - N_p) \cdot t_1$ computation time. Hence, the total computation time is approximately $(2 \cdot (N - N_p) + 1) \cdot t_1$. As expected, because N_p is close to N, $(2 \cdot (N - N_p) + 1) \cdot t_1$ is much smaller than the computation time of the conventional model $N \cdot t_1$.

2.3 Result

2.3.1 Verification

To verify the Block KMC model, we compared the conventional KMC results with those of the proposed model. These are the results of three different cases by varying the temperature and amount of initiator. The ‘Unit Step Number’ is fixed at 50.

Table 2.5 Operation conditions of Case1, Case2, and Case3.

	Case 1	Case 2	Case 3
Temperature (K)	480	490	520
Initiator (mol/L)	10^{-7}	10^{-6}	10^{-7}

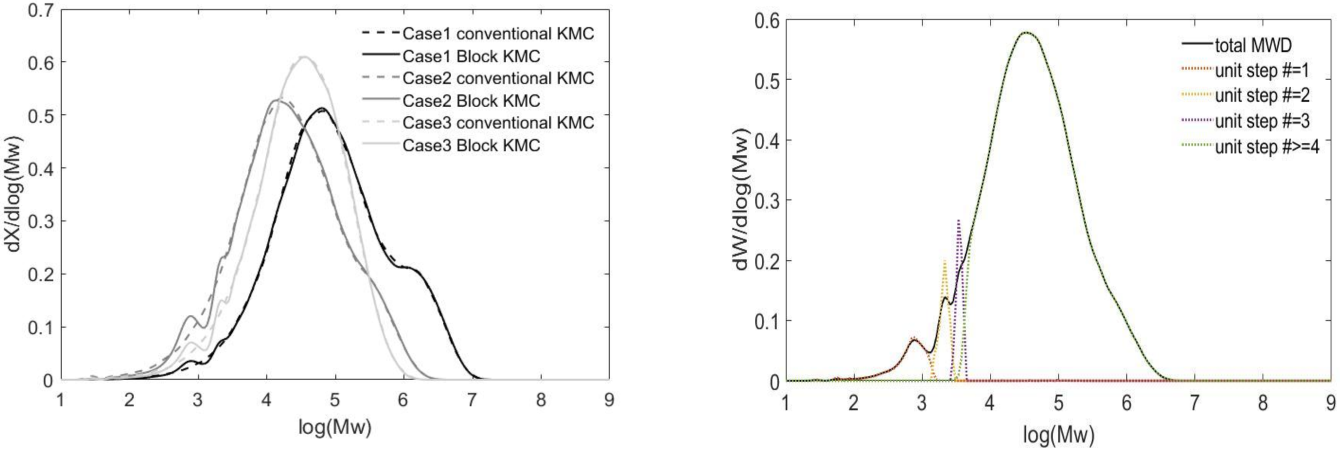


Figure 2.8 a) MWD results from conventional KMC and Block KMC models for each case b) MWD results depending on iteration number.

From the simulation results, the target model shows over- and under-peaks in the region of small molecular weights. To determine the reason, we classify the chain according to the iteration numbers of the algorithm in Figure 2.6. In Figure 2.8, the red dotted line represents the iteration number of 1, the yellow and purple dotted lines represent the MWD when the chain iterates twice and thrice, respectively, and the green dotted line corresponds to the MWD when the chain iterates four or more times. The red, yellow, and purple lines exactly match the point at which the small peaks appear in the entire MWD. This indicates that an error occurs when the number of repetitions is small. This is because the algorithm shows a little oscillation when the total number of Blocks is smaller than the 'Unit Step Number.' The error appears until the number of repetitions is 3 in case the 'Unit Step Number' is 50. To address this issue, when the target chain is short, we set a small 'Unit Step Number' and increase it as the chain length increases. In this way, the model accuracy can be improved while maintaining the calculation time. The detailed setting of the unit step number is shown in Eq. (2-44). As a result, the modified model removed the small peaks and agreed with the conventional model, as shown in Figure 2.9.

$$\begin{cases} \text{Unit step number} = 1 & \text{target chain length} < 50 \\ \text{Unit step number} = 20 & 50 \leq \text{target chain length} < 500 \\ \text{Unit step number} = 100 & 500 \leq \text{target chain length} < 5000 \\ \text{Unit step number} = 1000 & 5000 \leq \text{target chain length} \end{cases} \quad (2\text{-}44)$$

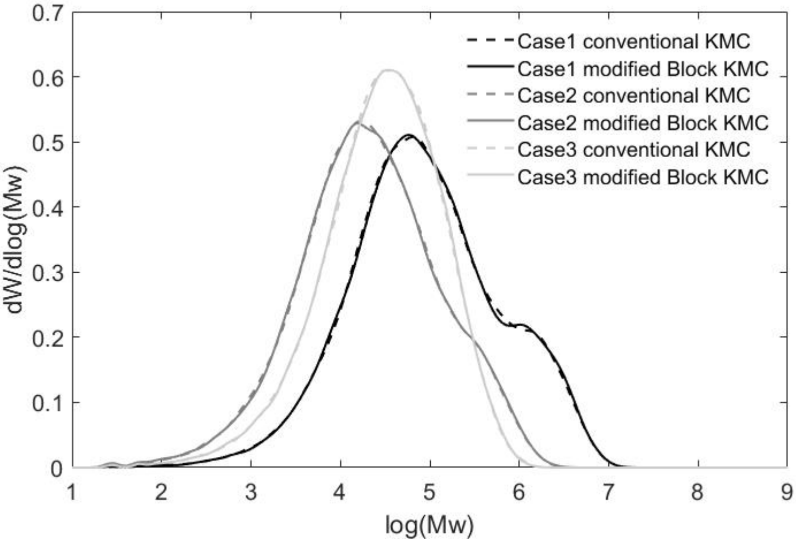


Figure 2.9 MWD results of Block KMC model in which 'Unit Step Number' is adjusted and compared with traditional KMC model.

2.3.2 Reduction in calculation time

To quantify the time saved by the proposed model, we measured the average time required to simulate one target chain for each model. Figure 2.10 shows the average calculation time per chain according to the chain length. The blue dots are the Block KMC model results, and the red dots are the conventional KMC model results. Although the time difference between the two models is small in the short chain, the Block KMC model becomes much faster in the long chain. As discussed in the Block KMC part in Methodology section, although the conventional method takes $\mathrm{N} \cdot \mathrm{t}_1$ calculation time, the proposed algorithm only takes $(2 \cdot (\mathrm{N} - \mathrm{N}_\mathrm{p}) + 1) \cdot \mathrm{t}_1$. This implies that the new algorithm is $1/(2 \cdot \left(1 - \frac{\mathrm{N}_\mathrm{p}}{\mathrm{N}}\right) + \frac{1}{\mathrm{N}})$ times faster than the conventional KMC. This is because the unit step number is set much larger in the long chain section. LDPE typically has a molecular weight of less than 50,000 $\mathrm{g/mol}$, which translates to 1,700 mol^{-1} in terms of the number of monomers in the chain. Therefore, at least 10 to 32 times reduction effect can be expected when the chain length is approximately 1700. As shown in Figure 2.10, although the calculation time remains constant along the average chain length in the Block KMC model, the calculation time of the conventional method increases in proportion to the average chain length. This indicates that the proposed method has an advantage in simulating polymers with a high Mw.

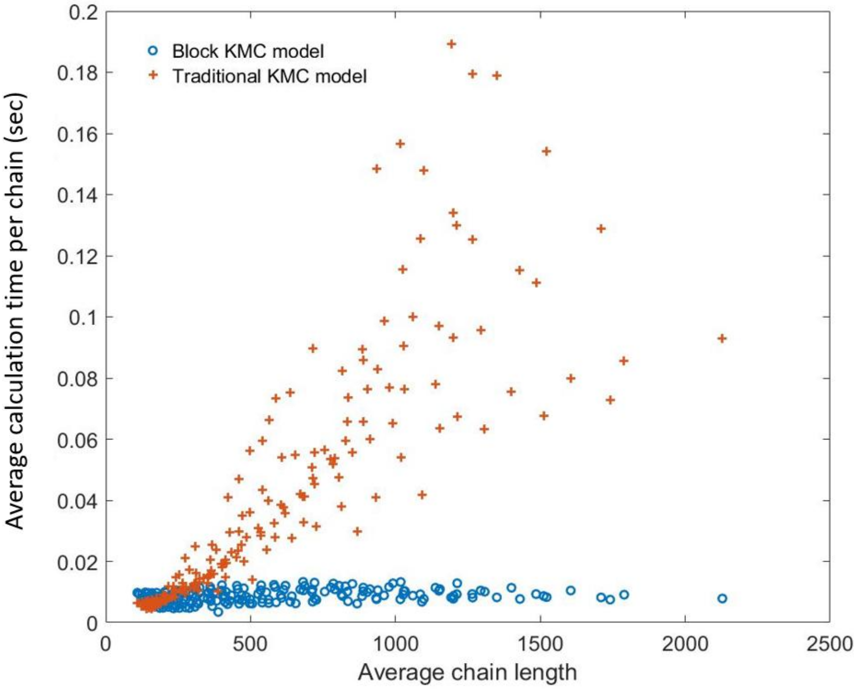


Figure 2.10 Average calculation time per one chain according to the average chain length for each model.

2.3.3 Case study

Figure 2.11 shows the dependence of the molecular weight distribution on the temperature. As the temperature decreases, the formation of the shoulder is observed owing to the generation of more long chains.

Figure 2.12 shows the reaction ratio according to the temperature at various initiator concentrations. The rate ratio of termination to propagation reaction decreases with increasing temperature because the kinetic constant of termination is less sensitive to temperature, and the live polymer concentration decreases as the temperature increases.

$$\frac{\mathrm{R_t}}{R_p} = \frac{k_t \lambda_0^2}{k_p \lambda_0 [M]} = \frac{\mathrm{k_t} \lambda_0}{k_p [M]} \tag{2-45}$$

The rate ratio of the chain transfer to polymer to propagation increases as the temperature increases. This is because the kinetic constant of the chain transfer to polymer reactions is more sensitive, and the dead polymer weight concentration increases with temperature.

$$\frac{\mathrm{R_{cp}}}{R_p} = \frac{k_{cp} \lambda_0 \mu_1}{k_p \lambda_0 [M]} = \frac{k_{cp} \mu_1}{k_p [M]} \tag{2-46}$$

The rate ratio of beta scission to propagation increases as the temperature increases. This is because the kinetic constant of the beta scission is more sensitive, and the secondary radical concentration increases with temperature, while the live polymer concentration decreases.

$$\frac{\mathrm{R_b}}{R_p} = \frac{k_b \lambda_{sec,0}}{k_p \lambda_0 [M]} \tag{2-47}$$

The rate ratio of chain transfer to monomer increases as the temperature increases because the rate constant of the chain transfer to monomer is more sensitive to temperature, and the ratio is constant even if the concentration changes.

$$\frac{\mathrm{R_{cm}}}{R_p} = \frac{k_{cm} \lambda_0 [M]}{k_p \lambda_0 [M]} = \frac{k_{cm}}{k_p} \tag{2-48}$$

The results of MWD according to the temperature can be interpreted by the ratio results of each reaction. MWD becomes narrower as the ratio of the chain transfer to polymer to the propagation increases with increasing temperature. This does not agree with the fact that the chain transfer to polymer broadens MWD. This means that the broadness of MWD is not determined only by the chain transfer to polymer, but also by the termination and beta scission. At low temperatures, as the termination reaction becomes dominant, the chains are combined to increase the proportion of long chains. At high temperatures, the beta scission predominates, and occurs better than the propagation in the secondary radical chain generated from the chain transfer to polymer. Therefore, the chain does not become a higher branched chain but breaks into smaller chains, which causes the left shift of MWD.

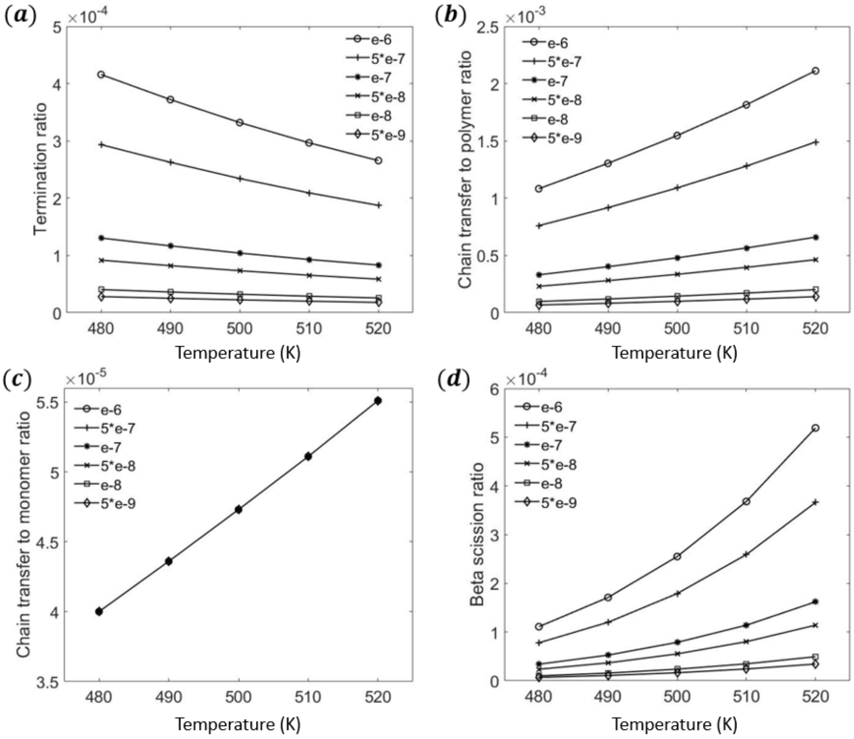


Figure 2.11 Reaction ratio according to temperature at various initiator concentrations (a) $\frac{\mathrm{R_t}}{R_p}$, (b) $\frac{\mathrm{R_{cp}}}{R_p}$, (c) $\frac{\mathrm{R_{cm}}}{R_p}$, and (d) $\frac{\mathrm{R_b}}{R_p}$.

Table 2.6 Temperature sensitivity of reaction mechanism.

Reaction mechanism	Activation energy J/mol	Temperature sensitivity
Initiation decomposition	149,347	Highest
beta scission	83,997	↑
Chain transfer to polymer	46,369	
Chain transfer to monomer	37,802	
Propagation	29,783	
Termination	10,080	Lowest

Figure 2.12 shows the dependence of the molecular weight distribution on the initiator concentration. As the initiator concentration increases, the formation of the shoulder is observed and the average chain length decreases.

The rate ratio of termination to propagation increases with increasing initiator concentration because the live polymer concentration increases as the initiator concentration increases. The rate ratio of the chain transfer to polymer to propagation also increases as the initiator concentration increases, because the dead polymer weight concentration increases as the initiator concentration increases. The rate ratio of beta scission to propagation increases as the temperature increases because the secondary radical concentration increases faster than the live polymer concentration. The rate ratio of the chain transfer to monomer is constant as the initiator concentration increases because the rate constant of the chain transfer to monomer does not depend on the concentration.

As the initiator concentration decreases, the MWD becomes narrower and shifts to the right. The plot shows a broad distribution because both long and short chains are produced as the termination and beta scission become faster. In addition, because the termination is more dominant than the beta scission under a low initiator concentration, the distribution plot moves to the right and the values of Mn and Mw also increase.

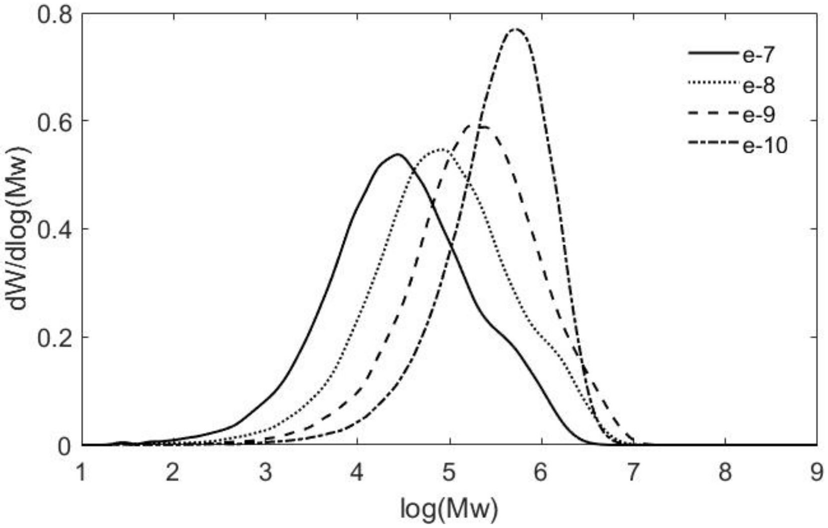


Figure 2.12 Effect of initiator concentration on MWD.

Table 2.7 Calculated properties of products at various initiator concentrations.

Initiator concentration	Mn	Mw	PDI	μ_1, mol/L	λ_0, mol/L	$\lambda_{sec,0}$, mol/L
$5 * 10^{-7}$	$1.27 * 10^5$	$6.97 * 10^5$	5.51	$1.80 * 10^{-3}$	$8.01 * 10^{-9}$	$6.39 * 10^{-12}$
$5 * 10^{-8}$	$3.35 * 10^5$	$1.67 * 10^6$	5.14	$5.55 * 10^{-4}$	$2.51 * 10^{-9}$	$6.12 * 10^{-13}$
$5 * 10^{-9}$	$5.67 * 10^5$	$2.36 * 10^6$	4.17	$1.64 * 10^{-4}$	$7.72 * 10^{-10}$	$5.57 * 10^{-14}$
10^{-10}	$6.34 * 10^5$	$1.39 * 10^6$	2.19	$1.78 * 10^{-5}$	$8.53 * 10^{-11}$	$6.54 * 10^{-16}$

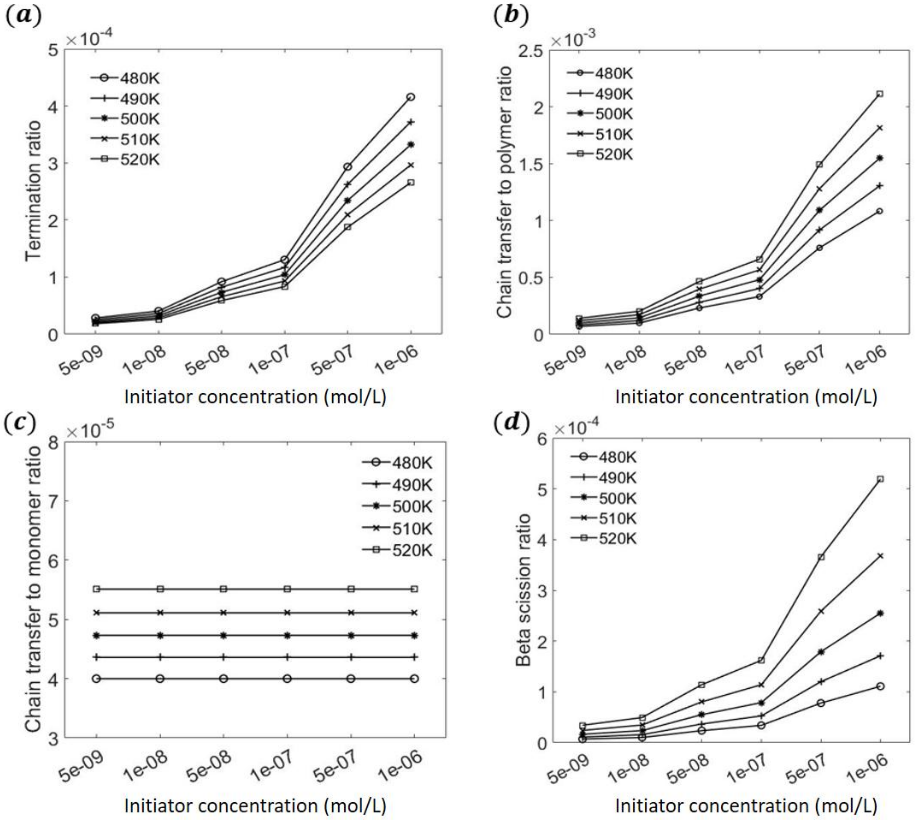


Figure 2.13 Reaction ratio according to initiator concentration at various temperatures (a) $\frac{R_t}{R_p}$, (b) $\frac{R_{cp}}{R_p}$, (c) $\frac{R_{cm}}{R_p}$, and (d) $\frac{R_b}{R_p}$.

2.3.4 Shouldering condition

Because each reaction affects the MWD in a complex manner, it is impossible to determine the condition in which a shoulder of the MWD is generated by only one reaction. Therefore, 180 cases were simulated by varying the temperature, initiator amount, and average residence time to determine the condition of the shoulder appearing. The range of operation conditions is listed in Table 2.8. In Figure 2.14, each reaction ratio is set as an axis. Blue and red dots represent the shouldered and non-shouldered cases, respectively. To formulate the reaction conditions in which the shoulders are generated, a flat plane is found that separates the blue and red dots through a linear support vector machine (SVM) technique. The hyperplane equation is given by Eq. (2-49), and the model classification accuracy is 89%.

Table 2.8 Range of operation condition

Operation condition	Range
Temperature	$480 – 520\ \text{K}$
Initiator concentration	$5 * 10^{-9} – 10^{-6}\ \text{mol/L}$
Average residence time	$10 – 60\ \text{sec}$

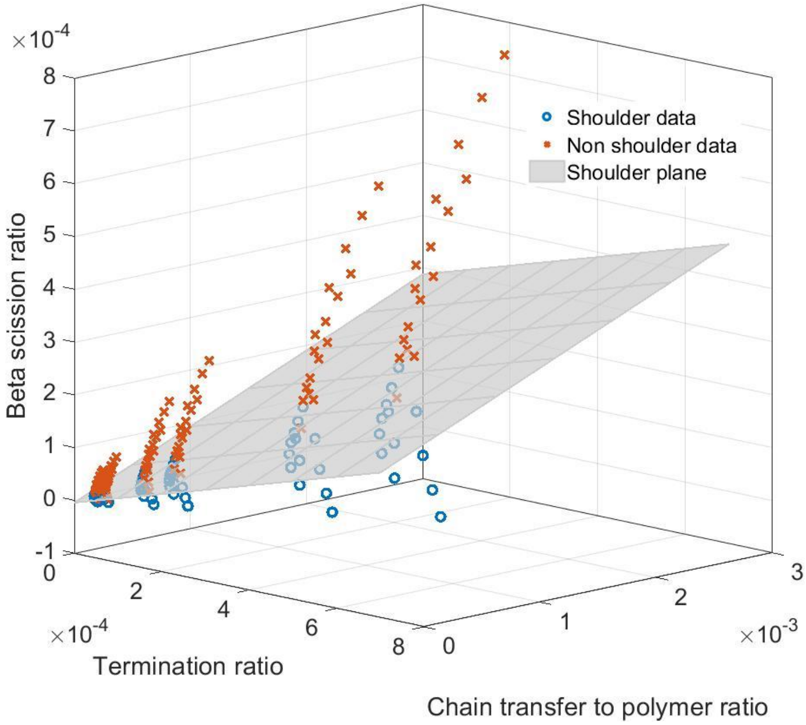


Figure 2.14 Shoulder plane for LDPE in CSTR reactor ($\mathbf{F_{sh}} = \mathbf{0}$ from equation (49)).

Under the condition that the F_{sh} value is positive, MWD does not have a shoulder. When F_{sh} is negative, the MWD has a shoulder. The faster the termination and the chain transfer to polymer and the slower the beta scission, the higher the probability that MWD will have a shoulder, which is consistent with a previous study. [12], [22], [32], [37] As discussed in the case study on temperature, it is confirmed that the shoulder can be generated by the termination and beta scission, despite the decrease in the the chain transfer to polymer reaction ratio.

$$F_{sh} = -0.585 * \phi_t - 0.223 * \phi_{cp} + 2.286 * \phi_{beta} + 0.6 * 10^{-5} \quad (2\text{-}49)$$

$$\phi_t = \text{Rate ratio of termination to propagation}, R_t/R_p$$

$$\phi_{cp} = \text{Rate ratio of the chain transfer to polymer to propagation}, R_{cp}/R_p$$

$$\phi_{beta} = \text{Rate ratio of beta scission to propagation}, R_{beta}/R_p$$

$$\text{if } \mathrm{F_{sh}} > 0, \quad \text{nonshouldered MWD} \quad (2\text{-}50)$$

$$\text{else } \mathrm{F_{sh}} < 0, \quad \text{shouldered MWD}$$

2.4 Conclusions

In this study, we proposed an MWD prediction model that coupled deterministic and stochastic methods. We introduced a new concept of ‘Block,’ which is a group of repeating reaction series. It reveals that the model is approximately 10 to 32 times faster than the model proposed by Neuhaus[30], with almost the same performance. There are two reasons for the model to reduce the computation time effectively. First, the number of blocks follows a binomial distribution, which makes it possible to determine each reaction quickly. Second, the propagation is much faster than other reactions. Because of this feature, the model can effectively reduce the calculation time for ordering the reactions. This reveals that the larger the average molecular weight of the polymer, the greater is the time reduction effect. Therefore, the model can be effectively applied to HDPE or higher-molecular-weight polymers.

We simulated the MWD and calculated the rate ratio of other reactions to the propagation according to the temperature and initiator concentration. The MWD was finally determined as the kinetic effects of each reaction that interact with each other. Some studies have dealt with the effects of operation conditions[12], [38], [39], but it is still challenging to analyze how the reactions comprehensively affect the MWD. In particular, the shouldering phenomenon remains a poorly understood topic, as it does not necessarily correlate with the average molecular weight. This makes it difficult to obtain the desired product, while making the operation of the reaction system difficult. Therefore, in this study, we provided a quantitative guideline for the appearance of the shoulder with 89% accuracy by simulating 180 cases for the operating conditions shown in Table 2.9.

The model was developed using a CSTR reactor. For a practical reactor, the viscosity of the fluid in the reactor increases as the polymerization proceeds, hindering perfect mixing and heat transfer. Furthermore, the reaction system was controlled with several input flows by regulating the temperature or initiator concentration. Therefore, industrial polymer reactors exhibit inhomogeneous behavior. The conventional KMC technique takes too long to combine with the flow data; whereas, the proposed model is applicable because of its reduced computation time. Therefore, the proposed model can be applied to an industrial-scale reactor for real-time control and optimization.

2.5 Notations

C_a	concentration of molecule a, mol/L
D_n	dead polymer chain of length n
f	effectiveness factor for initiator decomposition, dimensio
F_{sh}	shouldering measurement, dimensionless
k	kinetic constant
k_b	kinetic constant of beta scission reaction, s^{-1}
k_{cp}	kinetic constant of chain transfer to polymer reaction, L $\cdot s)^{-1}$
k_{cm}	kinetic constant of chain transfer to monomer reaction, $\cdot s)^{-1}$
k_d	kinetic constant of initiator decomposition reaction, s^{-1}
k_p	kinetic constant of propagation reaction, $L(mol \cdot s)^{-1}$

k_t	kinetic constant of termination by combination reaction $\cdot$ s$)^{-1}$
L_n	live polymer chain of length n
$L_{n,sec}$	secondary radical polymer chain of length n
L_{live}	chain length sampled from the live polymer chain leng
L_{mat}	the matrix where chain length data is stored
M	ethylene monomer
N	number of simulated taget chain
N_a	number of molecules
N_{AVO}	Avogadro number
N_{B1}	number of B1 blocks
N_{B2}	number of $\mathrm{B2}-1$ and $\mathrm{B2}-2$ blocks
N_{B3}	number of $\mathrm{B3}-1$ and $\mathrm{B3}-2$ blocks
N_{B4}	number of $\mathrm{B4}-1$ and $\mathrm{B4}-2$ blocks
n_{rand}	random number between 0– 1
N_{samp}	total number of taget chains to be simulated
P_{p1}, P_{p2}	Probabilities of propagation reaction
P_t	Probabilities of termination by combination reaction
P_{cp1}, P_{cp2}	Probabilities of chain transfer to polymer reaction
P_{cm}	Probabilities of chain transfer to monomer reaction
P_{b1}, P_{b2}	Probabilities of beta scission reaction
r	stochastic reaction rate
R_i	initiator decomposition reaction
R_{p1}, R_{p2}	propagation reaction

R_t	termination by combination reaction
R_{cp1}, R_{cp2}	chain transfer to polymer reaction
R_{cm}	chain transfer to monomer reaction
R_{beta1}, R_{bet}	beta scission reaction
r_{p1}, r_{p2}	stochastic rate of propagation reaction, $(\mathrm{mol} \cdot \mathrm{s})^{-1}$
r_t	stochastic rate of termination by combination reaction, ($\cdot\, \mathrm{s})^{-1}$
r_{cp1}, r_{cp2}	stochastic rate of chain transfer to polymer reaction, (m $\cdot\, \mathrm{s})^{-1}$
r_{cm}	stochastic rate of chain transfer to monomer reaction, ($\cdot\, \mathrm{s})^{-1}$
r_{b1}, r_{b2}	stochastic rate of beta scission reaction, $(\mathrm{mol} \cdot \mathrm{s})^{-1}$
T	Average residence time in CSTR reactor, sec
t	current simulation time, sec
t_{samp}	residence time of target chain, sec
V	total volume of reactor system, L
v	volumetric flow rate of input flow, L/s
$[D]$	concentration of dead polymer, mol/L
$[D_n]$	concentration of the dead polymer of length n, mol/L
$[I]$	concentration of initiator, mol/L
$[L]$	concentration of live polymer, mol/L
$[L_n]$	concentration of the live polymer of length n, mol/L
$[L_{sec}]$	concentration of secondary radical polymer, mol/L

$[L_{sec,n}]$	concentration of the secondary radical polymer of lengt /L
$[M]$	concentration of monomer, mol/L
λ_k	$\mathrm{k^{th}}$ moment of live polymer chain
$\lambda_{sec,k}$	$\mathrm{k^{th}}$ moment of secondary radical polymer chain
μ_k	$\mathrm{k^{th}}$ moment of dead polymer chain

2.6 Abbreviations

CSTR, continuous stirred-tank reactor; CTA, chain transfer agent; HDPE, high density polyethylene; KMC, Kinetic Monte Carlo; LDPE, low density polyethylene; MC, Monte Carlo; MI, melt index; MoM, method of moment; MWD, molecular weight distribution; PDI, polydispersity index; SMMA, single macromolecule approach; SVM, support vector machine

Chapter 3

Physics-informed deep learning for data-driven solutions of computational fluid dynamics

3.1 Introduction

Computational fluid dynamics (CFD) is a powerful tool that uses numerical approaches to solve engineering problems involving fluid motion and heat transfer. CFD has been employed in a wide range of research fields, such as aerodynamics [40], weather dynamics [41], environmental engineering, and industrial system modeling [42]. However, a critical issue of CFD simulations is their long computational times. Owing to this disadvantage, the application of CFD in analyzing the real-time behavior of systems and predicting non-simulated scenarios has been inherently limited.

Recent developments in machine learning have fundamentally transformed many classical engineering approaches and provided suitable alternative methods to solve complex tasks, including the simulation of complex fluid motion and heat transfer. Motivated by the potential of machine learning for speeding up numerical simulations without compromising the accuracy, a number of studies have focused on developing a surrogate model that aims to replace or accelerate CFD simulations. The most widely-used surrogate models in the field of chemical process engineering include the polynomial approximations [43], Gaussian process (GP) [44], and deep learning models [45]. The polynomial approximation is the most commonly used model;

however, it is simple and its applications are limited to less complex underlying models and local regions. GP is a nonparametric Bayesian approach that works well on small datasets and can provide a measure of uncertainty for predictions. GP is suitable for modeling small datasets where some prior knowledge of the generative process exists. Deep learning techniques have been widely used to develop surrogate models for flu-id-flow systems owing to their scalability with a large amount of data and hierarchical feature learning, which are suitable for large-scale and highly nonlinear systems, respectively. Ling et al. (2016) [46] used a deep neural network (NN) with embedded invariance for Reynolds-averaged turbulence modeling. Na et al. (2018) [47] proposed a non-linear surrogate model for toxic gas release scenarios using a variational autoencoder with deep convolutional layers and a deep NN. To obtain a reliable model, these deep learning approaches require a large amount of data to avoid the problem of overfitting and undue extrapolation. Although the feature extraction characteristic of deep learning is known to be advantageous, these extracted features are often inconsistent with physical implications. Hence, these purely data-based surrogate models cannot be reliably used for inference in untrained spaces [48].

To overcome these limitations, several approaches to combine scientific knowledge and deep learning models have been developed. Karpatne et al. (2017) [48] proposed a physics-guided NN that uses scientific knowledge as physics-based loss functions in the learning objective of NNs in order to obtain a physically consistent model. Raissi et al. [49] proposed a more advanced model called the PINN, in which the constraint term of the physical law described by general non-linear partial differential equations is incorporated into NN training. PINN is suitable for systems where the first principles are given in the form of partial differential equations, such as fluid flow systems governed by mass and momentum conservation laws. PINN has numerous advantages over purely data-driven models: it can be effectively trained with small datasets; it shows high performance in regions without observable data because it can be trained in any region to satisfy the governing equation without actual data. It can also solve the inverse problem of finding unknown parameters that converge toward their true values during training. Owing to these advantages, the PINN approach has been applied to several physical systems such as high-speed aerodynamic flows described by the Euler equation [50], wall modeling in large-eddy simulations [51], and the incompressible Navier-Stokes equation [52]. Previous studies have mainly dealt with fluid flow systems only, and none of them have solved reaction kinetics and heat transfer together, which is essential in the model development of chemical engineering systems.

In this study, we investigate the feasibility of applying PINN to a chemical reactor system, which is a continuous stirred tank reactor (CSTR) with a Van de Vusse reaction. A method for a training system, including mass conservation equations, momentum conservation equations, heat conservation equations, and species transport equations, is suggested. In particular, because the system requires multi-reference frames owing to the mixing domain, we present how the model framework is conceived for implementation. Further, as the system operates by several different types of physical equations and the number of system variables increases, reliable performance cannot be obtained with existing PINN methodology. Therefore, we suggest three strategies to improve model performance. First, a large number of training data points are needed to train the system; thus, mini-batch training is introduced to prevent memory errors and increase model accuracy. Second, because empirical and physics errors for each governing equation are of different scales and hinder proper learning, we adopt the concept of the weighted loss function. Third, a similarity-based sampling strategy is introduced, which can improve the performance of the model when a sudden change in the system state occurs.

3.2 PINN

PINN, first proposed by Raissi et al. [49], is effective in finding a robust solution from data derived from a system whose governing equations are in the form of partial differential equations. PINN embraces a partial differential form of physical law in machine learning, making it possible to learn not only the patterns of the data but also their underlying physical laws. PINN is based on the same algorithm as that of the feedforward NN, except for the loss function. While the gradients of the output value errors are backpropagated to update the weights and bias in the feedforward NN, additional loss terms regarding the physical law determined using automatic differentiation are included in PINN. Therefore, the loss function consists of an empirical error term (MSE_y) and a physics error term (MSE_f), as presented in Eq. (3-1).

$$MSE = MSE_y + MSE_f \tag{3-1}$$

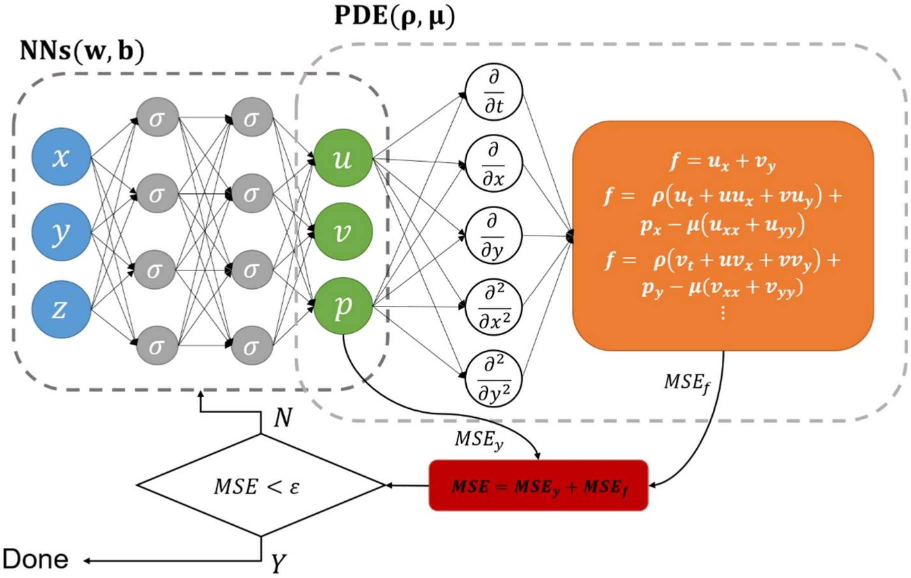
As presented in Eq. (3-2), the empirical error term is defined as the mean squared error of the output variables, where $\{x_y^i, y^i\}$ $(i = 1, \dots, N_y)$ denotes the training data on $y(x)$, N_y is the number of training data points, x_y^i is mostly defined as a set of independent space and time variables, and y^i are the system state variables such as velocity components and pressure. The physics error term in Eq. (3-3) is the mean squared error of the governing equations, which can be calculated from NNs through automatic differentiation, where $\{x_f^i\}$ specifies the collocation points for $f(x)$, N_f is the number of collocation points, and f is a set of physical equations.

$$MSE_y = \frac{1}{N_y}\sum_{i=1}^{N_y}\left|y(x_y^i) - y^i\right|^2 \tag{3-2}$$

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|f(x_f^i)\right|^2 \tag{3-3}$$

As shown in Figure 3.1, the model outputs from the NNs are fed into the first-principles model, and the error in the output variables and the error in the physical equation are summed and backpropagated to update the hyperparameters of the NNs.

PINNs are suitable for simulating a chemical reactor system, which is represented by a set of partial differential equations, such as mass and momentum balances, heat conservation equations, and species transport equations. However, because of the properties of the reactor system, where multiple reference frames exist and are governed by various types of physical equations, PINN structures need to be modified, and several strategies are needed to improve model performance. In the subsequent section, we suggest a loss function and model structure relevant to the target system.



$\mathbf{PDE(\rho, \mu)}$
$\mathbf{NNs(w, b)}$
x
y
z
σ
u
v
p
$\frac{\partial}{\partial t}$
$\frac{\partial}{\partial x}$
$\frac{\partial}{\partial y}$
$\frac{\partial^2}{\partial x^2}$
$\frac{\partial^2}{\partial y^2}$
$f = u_x + v_y$
$f = \rho(u_t + uu_x + vu_y) + p_x - \mu(u_{xx} + u_{yy})$
$f = \rho(v_t + uv_x + vv_y) + p_y - \mu(v_{xx} + v_{yy})$
⋮
MSE_f
MSE_y
$MSE = MSE_y + MSE_f$
$MSE < \varepsilon$
N
Y
Done

Figure 3.1 Schematic of PINN.

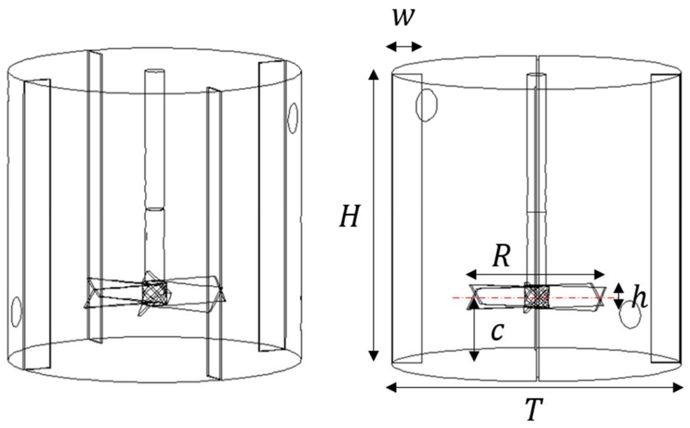
3.3 Model description

In this section, we introduce the modeling method of a CSTR with a Van de Vusse reaction using a model framework based on CFD and PINNs. The CSTR with the Van de Vusse reaction has been widely used as a benchmark problem owing to its nonlinearity [53]. There are two types of reactions commonly found in the literature: isothermal [54], [55] and non-isothermal [56], [57]. A non-isothermal reactor is used in this study to consider fluid flow, heat transfer, and reaction kinetics together.

3.3.1 CFD modeling

Figure 3.2 depicts the reactor geometry, which is a flat-bottomed, fully baffled tank, with a diameter of 0.288 m and a height of 0.288 m. It has one inlet, one outlet, and centrally located pitched-blade impellers that rotate at a speed of 10 rpm.

Van de Vusse reactions are referred to by Ridlehoover et al. [58], and the kinetic parameters that we used for the target system are presented in Table 3.1. The model component properties and the initial and boundary conditions are also listed in Table 3.1.



$H = 0.288\ m$
$T = 0.288\ m$
$c = 0.055\ m$
$w = 0.03\ m$
$h = 0.021\ m$
$R = 0.144\ m$

Figure 3.2 Reactor system geometry and tank specifications.

Table 3.1 Reaction kinetics, component properties, and initial and boundary conditions.

Reactions	Kinetic parameters	
	$A\ (1/s)$	$E_a\ (J/mol)$
$\mathrm{A} \overset{k_1}{\rightarrow} \mathrm{B}$	$3.575 * 10^9$	$8.109 * 10^7$
$\mathrm{B} \overset{k_2}{\rightarrow} \mathrm{C}$	$3.575 * 10^9$	$8.109 * 10^7$
$2\mathrm{A} \overset{k_3}{\rightarrow} \mathrm{D}$	$2.512 * 10^9$	$7.113 * 10^7$
Component properties		
ρ	$934.6\ \mathrm{kg/m^3}$	
μ	$1.0 * 10^{-3}\ \mathrm{kg/m \cdot s}$	
C_p	$3{,}010\ \mathrm{J/K \cdot kg}$	
K	$0.6\ \mathrm{W/m \cdot K}$	
Mw_A	$155.3\ \mathrm{kg/mol}$	
D	$2.88 * 10^{-5}\ m^2/s$	
Initial conditions		
C_{a_0}	$2.23 * 10^{-3}\ \mathrm{mol/L}$	
C_{b_0}	$1.04 * 10^{-3}\ \mathrm{mol/L}$	
C_{c_0}	$0.91 * 10^{-3}\ \mathrm{mol/L}$	
C_{d_0}	$0.92 * 10^{-3}\ \mathrm{mol/L}$	
T_0	$79.591\ ℃$	
T_f	$104.9\ ℃$	
$C_{a,f}$	$5.1 * 10^{-3}\ \mathrm{mol/L}$	
T_k	$77.5\ ℃$	
Q_f	$3.89 * 10^{-2}\ \mathrm{L/sec}$	

The reactor is divided into two domains: the inner and outer domains. The inner domain is assigned around the impeller, and the outer domain is the rest of the domain. The inner domain refers to the moving reference frame, which rotates in the opposite direction of the impeller, and the outer domain refers to the stationary frame. Therefore, the governing equation for each domain varies according to the reference frame. A set of governing equations that satisfy each domain are explained in the section on the governing equation, and the PINN model architecture that integrates two different domains into one model is introduced in the section on the PINN architecture.

The mesh structure is composed of 0.3M cells. We have performed a mesh convergence test on 8 different meshes. The graph on the left in the Fig. 3.3 shows the average velocity value at the outlet according to the number of meshes and the graph on the right shows the average mole fraction of d at the outlet according to the number of meshes. When the number of meshes is more than 0.2M, it can be seen that the simulation results converge. We used ANSYS-Fluent v19.2 to generate the model training data calculated through CFD. After the simulations were completed, the values of the state variables (velocity components, pressure, and temperature) and the mole fraction of each component according to time and position were exported.

3.3.2 Governing equations

The target system is governed by the mass, momentum, and energy conservation laws and the species transport equations. We used a laminar model

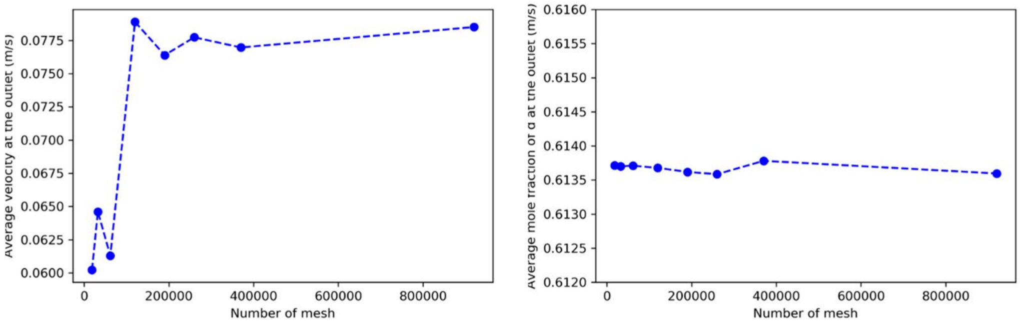


Figure 3.3 Mesh convergence test results.

for the viscous model and assumed a compressible flow. Specifically, for our system, we consider two domains with different reference frames, and thus each domain follows a different form of physics law. The velocity vector is converted from a moving reference frame to an absolute frame using

$$\vec{v} = \overrightarrow{v_r} + (\vec{w} \times \vec{r}). \tag{3-4}$$

The term $\vec{v}$ is the absolute velocity vector, $\overrightarrow{v_r}$ is the relative velocity vector, $\vec{w}$ is the rotating angular velocity, and $\vec{r}$ is the position vector from the origin of the moving system. The governing equations for fluid flow in an absolute frame can be written as follows:

The equation for the conservation of mass is

$$\frac{\delta\rho}{\delta t} + \nabla \cdot (\rho\vec{v}) = 0, \tag{3-5}$$

where ρ is the density.

The equation for the conservation of momentum is

$$\frac{\delta}{\delta t}(\rho\vec{v}) + \nabla \cdot (\rho\vec{v}\vec{v}) = -\nabla p + \nabla \cdot (\bar{\bar{\tau}}), \tag{3-6}$$

where $\bar{\bar{\tau}}$ is the stress tensor.

The equation for the conservation of energy is

$$\frac{\delta}{\delta t}(\rho E) + \nabla \cdot \big(\vec{v}(\rho E + p)\big) = \nabla \cdot (k\nabla T + \bar{\bar{\tau}} \cdot \vec{v}), \tag{3-7}$$

where E is the internal energy.

The equation for species transport is

$$\frac{\delta}{\delta t}(\rho Y_i) + \nabla \cdot (\rho \vec{v} Y_i) = -\nabla(\rho D \nabla Y_i) + R_i, \tag{3-8}$$

where Y_i is the local mass fraction of the i^{th} species, and R_i is the net rate of the production of species i by a chemical reaction.

For the relative velocity formulation, the governing equations of the fluid flow in a moving reference frame can be written as follows:

The equation for the conservation of mass is

$$\frac{\delta \rho}{\delta t} + \nabla \cdot (\rho \overrightarrow{v_r}) = 0. \tag{3-9}$$

The equation for the conservation of momentum is

$$\frac{\delta}{\delta t}(\rho \overrightarrow{v_r}) + \nabla \cdot (\rho \overrightarrow{v_r} \overrightarrow{v_r}) + \rho(2\vec{w} \times \overrightarrow{v_r} + \vec{w} \times \vec{w} \times \vec{r}) \\ = -\nabla p + \nabla \cdot (\bar{\bar{\tau}}_r). \tag{3-10}$$

The momentum equation contains two additional acceleration terms. The $(2\vec{w} \times \overrightarrow{v_r})$ term is the Coriolis acceleration, and the $(\vec{w} \times \vec{w} \times \vec{r})$ term is the centripetal acceleration.

The equation for the conservation of energy is

$$\frac{\delta}{\delta t}(\rho E_r) + \nabla \cdot (\rho \vec{v}_r H_r) = \nabla \cdot (k \nabla T + \bar{\bar{\tau}}_r \cdot \overrightarrow{v_r}), \tag{3-11}$$

where E_r is the relative internal energy and H_r is the relative total enthalpy.

The equation for species transport is

$$\frac{\delta}{\delta t}(\rho Y_i) + \nabla \cdot (\rho \vec{v}_r Y_i) = -\nabla(\rho D \nabla Y_i) + R_i. \tag{3-12}$$

These equations will contribute to PINN training.

3.3.3 PINN architecture

We design the PINN to infer all the states of interest, i.e., velocity components (u, v, w), temperature (T), and mole fractions of a, b, c, and d (f_a, f_b, f_c, f_d). $\boldsymbol{x} := (x, y, z, t)$ is introduced as a set of independent space and time variables and $\boldsymbol{y} := (u, v, w, T, f_a, f_b, f_c, p)$ represents the set of state variables. Based on the PINN framework [49], the output variable y is approximated using a feed-forward multi-layer NN:

$$\boldsymbol{y}(\boldsymbol{x}) \approx \mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}), \tag{3-13}$$

where $\mathcal{N}$ is an $\mathcal{L}$-layer NN with a set of input variables $\boldsymbol{x}$ and a network parameter $\boldsymbol{\theta}$. This network represents a compositional function, which is

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}) = \Sigma^{\mathcal{L}} \circ \Sigma^{\mathcal{L}-1} \circ \cdots \circ \Sigma^{1}(\boldsymbol{x}) \tag{3-14}$$

$$\Sigma^{\lambda}(z^{\lambda-1}) := z^{\lambda} = \sigma^{\lambda}(\boldsymbol{W}^{\lambda} \cdot z^{\lambda-1} + \boldsymbol{b}^{\lambda}) \qquad \text{where } \lambda = 1, \ldots, \mathcal{L} \tag{3-15}$$

where the symbol $\circ$ is the composition operation, $\boldsymbol{\lambda}$ is the layer number, $z^0 := \boldsymbol{x}$ is the input to the network, $z^{\mathcal{L}} := \boldsymbol{y}$ is the output of the network, $\boldsymbol{W}^{\boldsymbol{\lambda}}$ and $\boldsymbol{b}^{\boldsymbol{\lambda}}$ are the network parameters of layer λ known as weight and bias, and σ^{λ} is the activation function.

The architecture of a PINN consists of 11 layers, including 100 neurons for each layer, whose numbers are determined empirically by simulating several cases. Too many layers prevent a model from being generalized, and a small number of layers are not sufficient to represent the system. Fig. 3.4 shows training and test loss according to the learning Epoch for various number of layer. The number of epochs means how many times the entire dataset is passed forward and backward through the neural network. As shown in Figure 3.4, when the number of layer exceeds 11, model accuracy does not improve anymore. Training and test data loss oscillate more when the number of layer is 13, reducing model stability. Each node is fully connected to the nodes of the previous layer and contributes to the next layer's nodes with the tangent hyperbolic activation function.

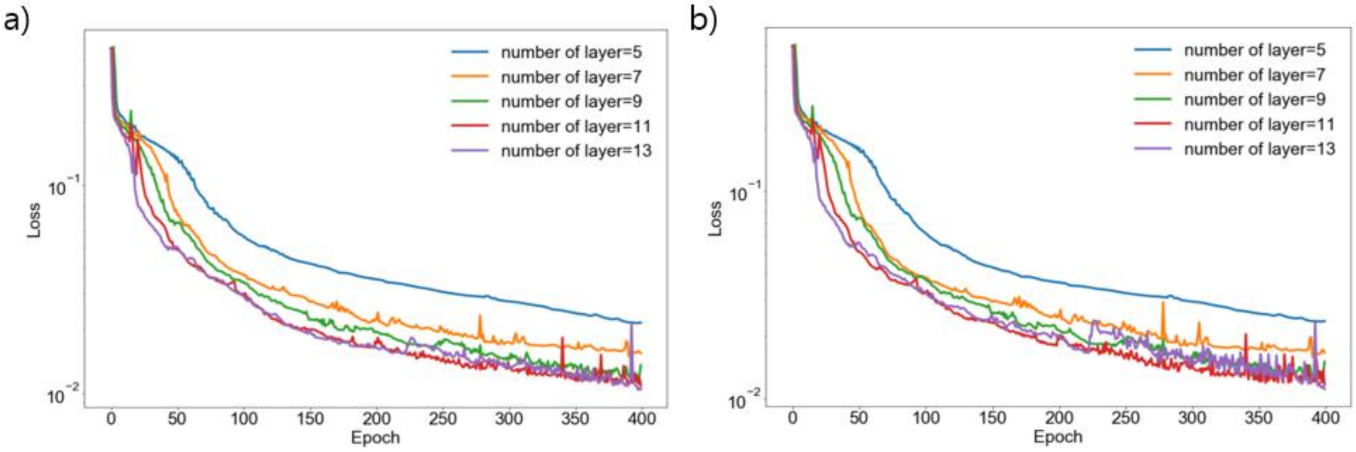


Figure 3.4 Loss graph for various number of layer: a) training data loss, b) test data loss.

As mentioned in the previous section, the loss function consists of an empirical loss term, which is the mean squared error between the real and predicted values of the output variables, and a physics loss term, which is the model error of the governing equations consisting of derivative terms calculated through automatic differentiation from the networks. A weighted loss function is used here. In particular, for our system, the total loss of the network, MSE, includes a loss term for the output variables $(MSE_u, MSE_T$ and $MSE_Y)$ and the physics equations $(MSE_{f_{cont}}, MSE_{f_u}, MSE_{f_k}$ and $MSE_{f_{spec}})$.

$$\begin{aligned} &MSE \\ &= w_{emp}(MSE_u + MSE_T + MSE_Y) + w_{phy_1} MSE_{f_{cont}} \\ &+ w_{phy_2} MSE_{f_u} + w_{phy_3} MSE_{f_k} + w_{phy_4} MSE_{f_{spec}}, \end{aligned} \tag{3-16}$$

where the empirical loss term is evaluated at training points $\{x_u^i, y_u^i, z_u^i, t_u^i, u^i, v^i, w^i, T^i, f_a^i, f_b^i, f_c^i\}$ $i = 1, \dots, N_u$, and the physics loss term is evaluated at a collocation point $\{x_f^i, y_f^i, z_f^i, t_f^i\}$ $i = 1, \dots, N_f$. The empirical loss terms are expressed as

$$MSE_u = \frac{1}{N_u} \sum_{i=1}^{N_u} (|u(\boldsymbol{x})^i - u^i|^2 + |v(\boldsymbol{x})^i - v^i|^2 + |w(\boldsymbol{x})^i - w^i|^2) \tag{3-17}$$

$$MSE_T = \frac{1}{N_u} \sum_{i=1}^{N_u} (|T(\boldsymbol{x})^i - T^i|^2) \tag{3-18}$$

$$MSE_Y = \frac{1}{N_u} \sum_{i=1}^{N_u} (|f_a(\boldsymbol{x})^i - f_a^i|^2 + |f_b(\boldsymbol{x})^i - f_b^i|^2 + |f_c(\boldsymbol{x})^i - f_c^i|^2) \tag{3-19}$$

The physics loss terms differ depending on whether the collocation point belongs to the inner or outer domains. The loss term for the mass conservation equation is

$$MSE_{f_{cont}} = \frac{1}{N_f}\sum_{i=1}^{N_f}(\left|f_{cont}^{i}\right|^2). \tag{3-20}$$

The loss term for the momentum conservation equation is

$$MSE_{f_u} \tag{3-21}$$
$$= \begin{cases} \frac{1}{N_{in}}\sum_{i=1}^{N_{in}}\left(\left|f_{u_{in}}^{i}\right|^2 + \left|f_{u_{in}}^{i}\right|^2 + \left|f_{w}^{i}\right|^2\right) & \{x_f^i, y_f^i, z_f^i, t_f^i\} \in \boldsymbol{X}_{in} \\ \frac{1}{N_{out}}\sum_{i=1}^{N_{out}}\left(\left|f_{u_{out}}^{i}\right|^2 + \left|f_{u_{out}}^{i}\right|^2 + \left|f_{w}^{i}\right|^2\right) & \{x_f^i, y_f^i, z_f^i, t_f^i\} \in \boldsymbol{X}_{out} \end{cases},$$

where $\boldsymbol{X_{in}}$ denotes the inner domain and $\boldsymbol{X_{out}}$ denotes the outer domain.

The loss term for the energy conservation equation is

$$MSE_{f_k} = \begin{cases} \frac{1}{N_{in}}\sum_{i=1}^{N_{in}}\left(\left|f_{k_{in}}^{i}\right|^2\right) & \{x_f^i, y_f^i, z_f^i, t_f^i\} \in \boldsymbol{X}_{in} \\ \frac{1}{N_{out}}\sum_{i=1}^{N_{out}}\left(\left|f_{k_{out}}^{i}\right|^2\right) & \{x_f^i, y_f^i, z_f^i, t_f^i\} \in \boldsymbol{X}_{out} \end{cases}. \tag{3-22}$$

The loss term for the species transport equation is

$$MSE_{f_{spec}} = \begin{cases} \frac{1}{N_{in}} \sum_{i=1}^{N_{in}} \left(\left| f_{spec_{a_{in}}}^{i} \right|^{2} + \left| f_{spec_{b_{in}}}^{i} \right|^{2} + \left| f_{spec_{b_{in}}}^{i} \right|^{2} \right) & \{x_f^i, y_f^i, z_f^i, t_f^i\} \in \boldsymbol{X}_{in} \\ \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} \left(\left| f_{spec_{a_{out}}}^{i} \right|^{2} + \left| f_{spec_{b_{out}}}^{i} \right|^{2} + \left| f_{spec_{b_{out}}}^{i} \right|^{2} \right) & \{x_f^i, y_f^i, z_f^i, t_f^i\} \in \boldsymbol{X}_{out} \end{cases} \tag{3-23}$$

The detailed forms of all governing equations are listed in the Appendix. The network is trained by minimizing the total loss using the backpropagation algorithm, where every weight is updated by calculating the gradient in each weight with respect to a loss function with Eq. (24). It is repeated until the total loss is smaller than the stopping criteria. The model was trained by the optimization package, AdamOptimizer in TensorFlow.

$$w_{ij}^{*} = w_{ij} + \alpha * \frac{\partial MSE}{\partial w_{ij}} \tag{3-24}$$

where the w_{ij} is the weight for neuron j in layer i in the current training iteration, w_{ij}^{*} is the updated weight for neuron j in layer i, and α is the learning rate.

As shown in Fig. 3.5, the entire training data set is composed of $\mathrm{D_{emp}}$ and $\mathrm{D_{phy}}$. $\mathrm{D_{emp}}$ is the training data set used to calculate the empirical error which is the set of the input and output variables of the network. $\mathrm{D_{phy}}$ is the collocation data set used to calculate the physics error which is the set of the only input variables of the network. Collocation data are divided into two domains, and the physics losses are calculated separately using Eqs. (20)–(23). These physics losses are summed and combined with the empirical loss calculated by Eqs. (17)–(19) to determine the total loss, and the network is trained by minimizing the total loss using the backpropagation algorithm. A total of $2 * 10^5$ data points were used for training data and $2 * 10^5$ data points were used for collocation points. The PINNs were trained using Tensorflow v1.10.0 with a NVIDIA GTX Titan GPU.

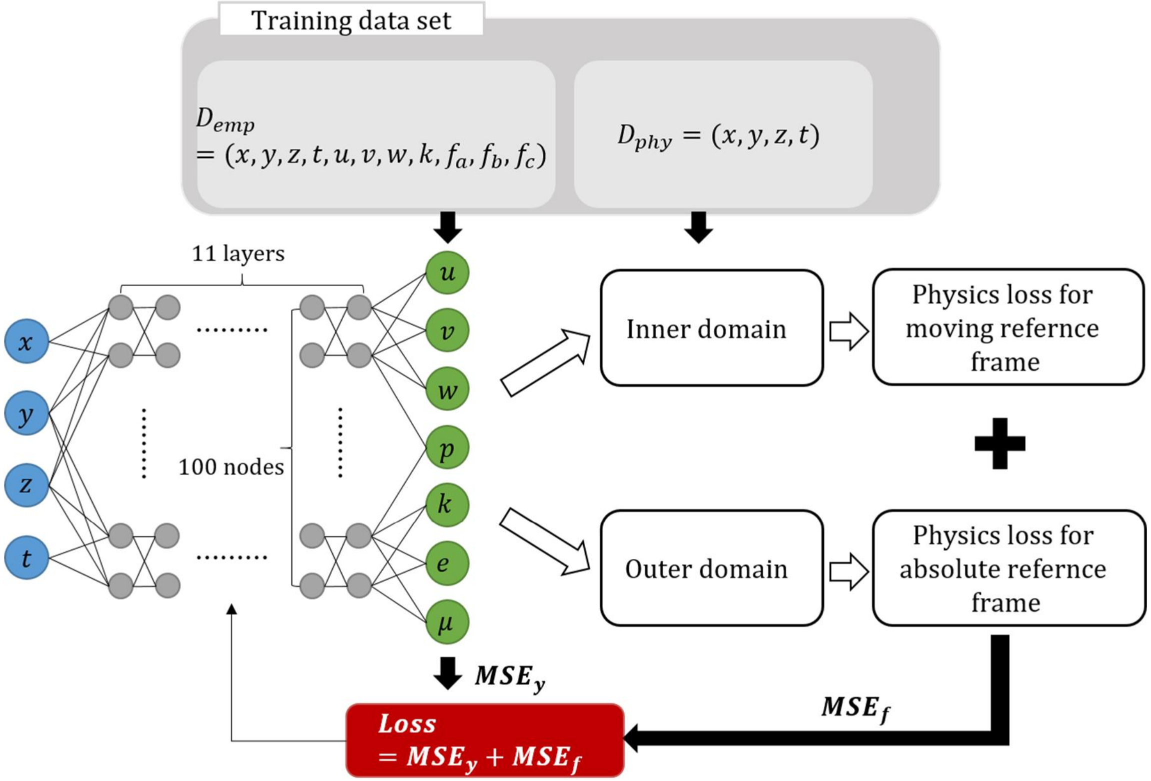


Figure 3.5 PINN model structure for CSTR with Van de Vusse reaction.

3.4 Result and Discussion

3.4.1 Model verification

Figure 3.6 and Figure 3.7 show contour plots of the velocity magnitude and mole fraction of component A for the CFD model and the PINN model in a vertically sliced cross section of the reactor. The PINN model shows good agreement with the CFD model for all state variables, extracting the proper flow pattern. The L_2 relative error as presented in Eq. (3-25) of the PINN model is $1.98 * 10^{-1}$ for the velocity magnitude and $3.661 * 10^{-2}$ for the mole fraction of component A (Table 3.2). The first reason for this large L_2 relative error is that the CFD simulation does not provide an exact solution with a numerical error, which leads to errors in the PINN model. The second reason is that the system has many state variables to be learned and governed by highly nonlinear physical equations, which makes it difficult for the surrogate model to find the global optimum.

$$L_2 \text{ relative error} = \sqrt{\frac{\sum_{i=1}^{N}\left(y_{real,i} - y_{pred,i}\right)^2}{\sum_{i=1}^{N} y_{real,i}^2}} \tag{3-25}$$

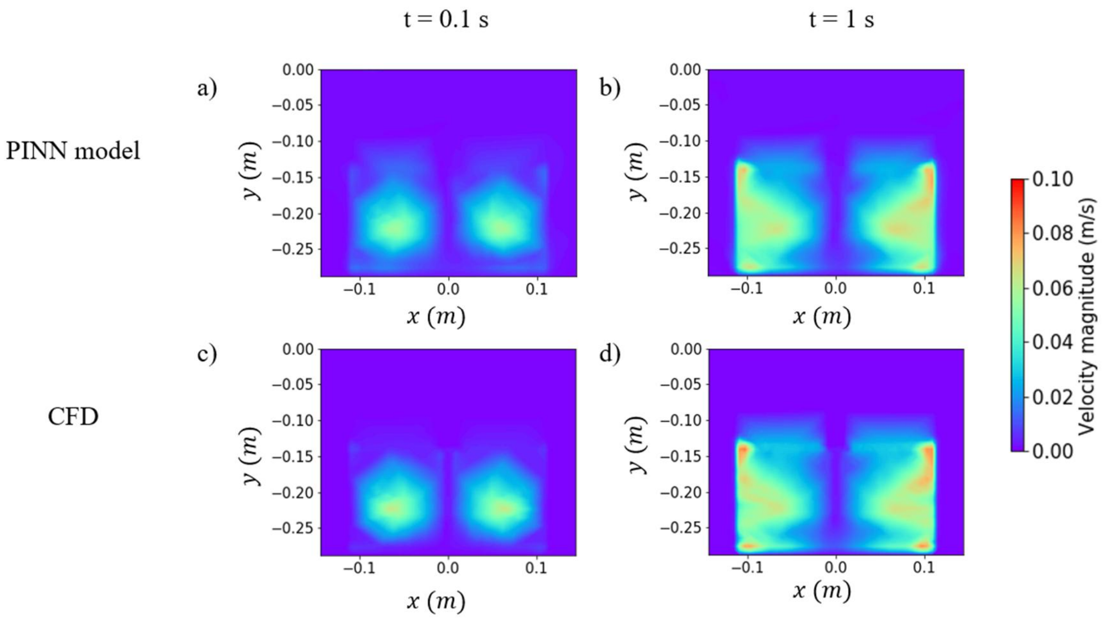


Figure 3.6 Comparison of PINN model and CFD for velocity magnitude: PINN model solutions of the velocity magnitude at 0.0005 s and 0.005 s (a, b), CFD solutions of the velocity magnitude at 0.0005 s and 0.005 s (c, d).

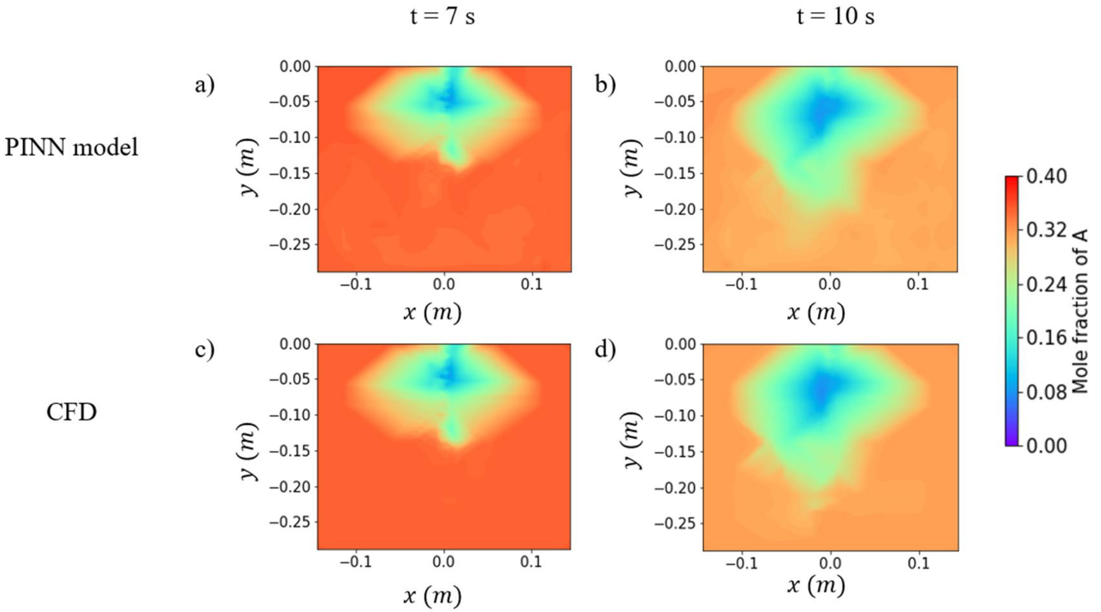


Figure 3.7 Comparison of PINN model and CFD for mole fraction of A: PINN model solutions of the mole fraction of component A at 0.035 s and 0.05 s (a, b), CFD solutions of the mole fraction of component A at 0.035 s and 0.05 s (c, d).

Table 3.2 L2 relative error of PINN model for states variables.

State variable	L_2 Relative error
Velocity magnitude	$1.98 * 10^{-1}$
Mole fraction of A	$3.66 * 10^{-2}$
Mole fraction of B	$2.21 * 10^{-2}$
Mole fraction of C	$1.25 * 10^{-2}$
Temperature	$1.22 * 10^{-1}$

Figure 3.8 shows a comparison between the performance of PINN and NN in developing the surrogate model. We use the same network structure for the NN as the PINN, and this network is trained with the loss function as follows:

$$MSE = w_{emp}(MSE_u + MSE_T + MSE_Y), \tag{3-26}$$

where the weight was assigned the same value as in the PINN model.

NN shows a higher accuracy in predicting output values, with an empirical loss lower than that of the PINN. However, the physics loss of the NN is much larger than that of the PINN because the NN does not learn physical laws. In Fig. 3.9, the images of h) and i) represent the z-component of the learned velocity vector of the PINN and NN's, respectively for the cross section in xz plane. While the PINN can extract the representative characteristic of the flow, NN shows noise that does not have any physical meaning because of overfitting. For a simple system, the empirical error of the NN is sometimes larger than that of the PINN; however, for the CSTR with an impeller, which has a complex geometry, PINN mostly has a larger empirical error than the NN. This is because the boundary condition, such as the non-slip condition on the surface of the impeller, that imposes non-linearity on the system makes it difficult for the PINN to obtain the optimum solution. Furthermore, the error of the governing equation, for example, the continuity equation, increases in regions where the velocity varies significantly, such as the boundary. Therefore, the loss term for the continuity equation implies that the model does not change significantly at the boundary condition, which yields a large empirical error. For this reason, the empirical error of PINN is larger than that of the NN, even though its physics error is much smaller than that of the NN.

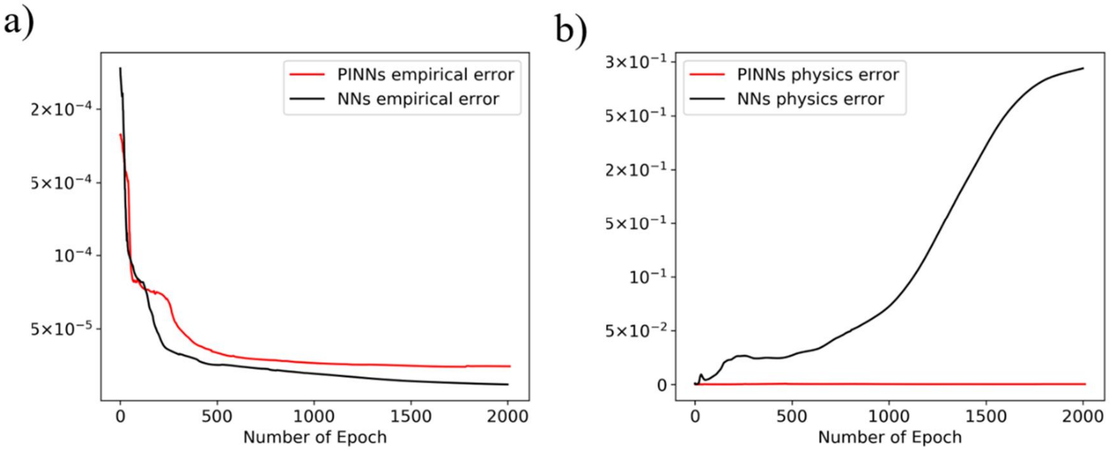


Figure 3.8 Loss graph of PINN and NN: a) empirical loss of PINN and NN against number of epochs and b) physics loss of PINN and NN against number of epochs.

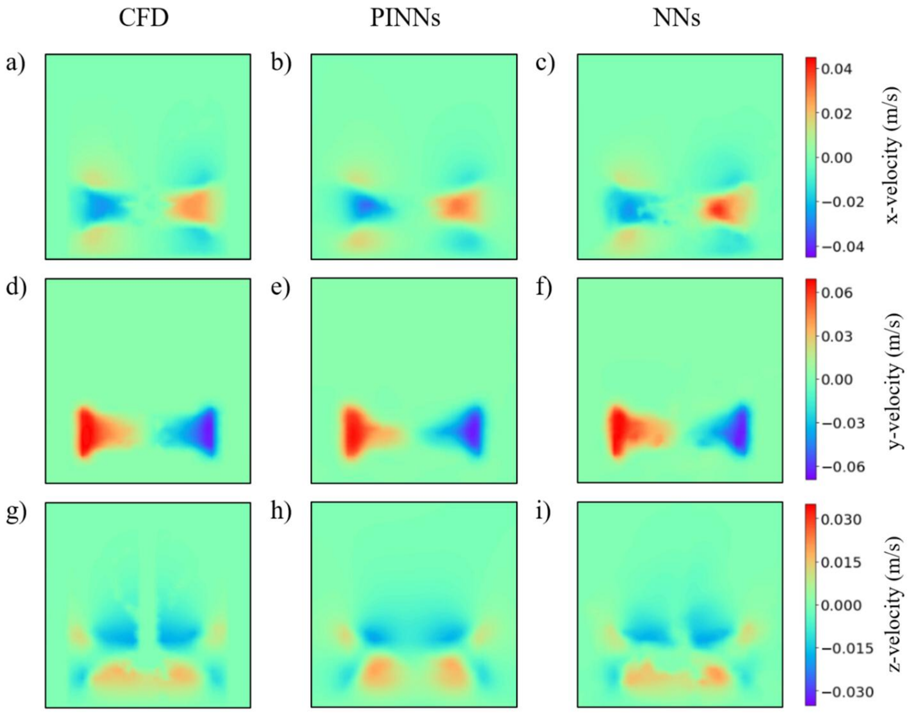


Figure 3.9 Comparison of PINN model, NN model, and CFD for velocity components: x-direction of velocity solutions obtained using CFD, PINN model, and NN model (a, b, c), y-direction of velocity solutions obtained using CFD, PINN model, and NN model (d, e, f), z-direction of ve-locity solutions obtained using CFD, PINN model, and NN model (g, h, i).

3.4.2 Improvement of model performance

3.4.2.1 Mini-batch training

We encountered a memory allocation error when more than approximately 5000 data were trained at once using a 16G GPU. This is because TensorFlow pre-allocates the entire memory of the GPU [59]. Mini-batch training can resolve the memory error issue by dividing the entire dataset into several batches, making it possible to require memory only for the batch size, and not the total data size. The parameter is updated for every mini-batch of n training data points. When the batch size is 1, the so-called stochastic gradient descent, the model encounters the issue of high fluctuation of parameters during training [60]. A proper batch size should be specified to avoid model performance degradation and reduce the variance of the updated parameters for more stable convergence. Batch training not only resolves the memory allocation error but also improves the learning speed of the model. As shown in Figure 3.10, using smaller batch sizes can lead to faster convergence. While a large batch size is known to lead to poor generalization [60], the trained model with smaller batches showed a smaller error.

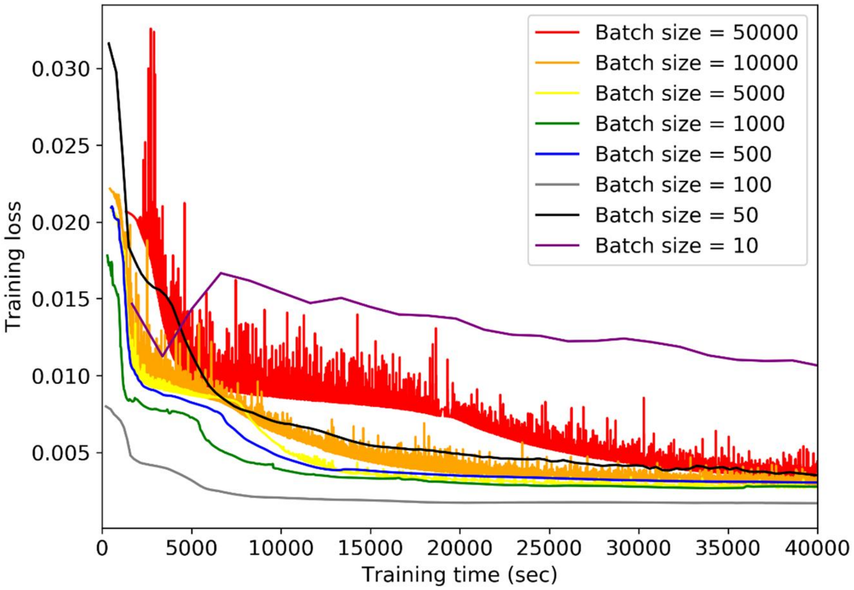


Figure 3.10 Total loss of PINN model for different batch sizes against Training time (sec).

3.4.2.2 Weighted loss function

The loss function consists of two parts: empirical and physics terms. The value of each term should be maintained at a similar level for proper learning. To maintain the size of each term, we introduce a weight for every loss term. The weight of each physics loss term is presented in Table 3.3. Because the loss values of the momentum and energy conservation equations are very high, the weight of each term was set to a low value. The terms constituting the energy conservation equation are, for example, $k\nabla T$, inherently larger than the terms of the momentum conservation equation because the scale of the temperature variable is much larger than that of the velocity variables. After setting the initial weights, we adjusted the weights further. We found that each weight affects the performance of the model. As shown in Figure 3.11, when the weight of the empirical term is set to a high value, the contour plots show more obvious and detailed patterns. This is because the model behaves similarly to a NN as the weight of the empirical error increases. Accordingly, the empirical loss decreases and the physics loss increases as the weight of the empirical error is set to a high value, as shown in Figure 3.12.

Table 3.3 Weight of each loss term.

w_{phy_1}	1
w_{phy_2}	$1 * 10^{-6}$
w_{phy_3}	$1 * 10^{-18}$
w_{phy_4}	1

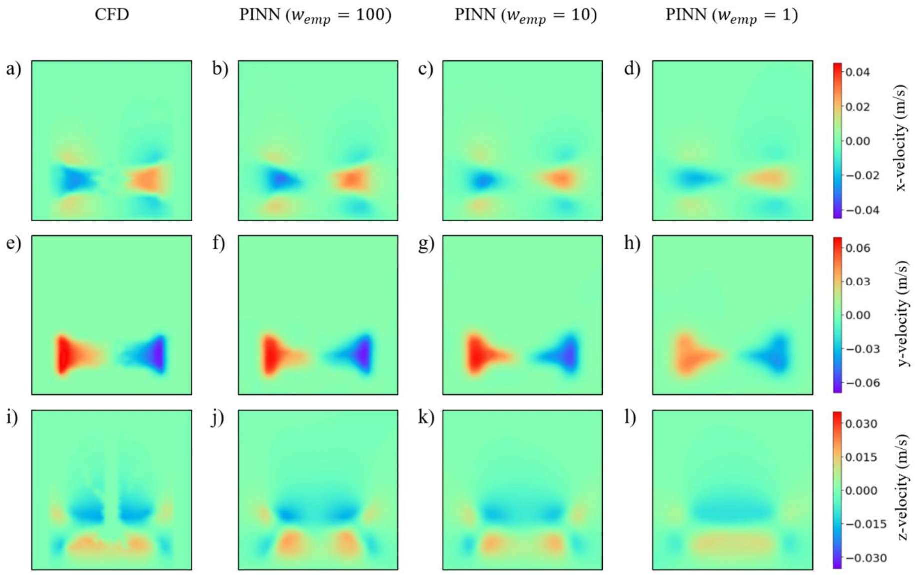


Figure 3.11 Comparison of PINN models for several empirical term weights: x-direction of veloc-ity solutions obtained using CFD and PINN model for empirical term weights of 100, 10, and 1 (a, b, c, d), y-direction of velocity solutions obtained using CFD and PINN model for empirical term weights of 100, 10, and 1 (e, f, g, h), z-direction of velocity solutions obtained using CFD and PINN model for empirical term weights of 100, 10, and 1 (i, j, k, l).

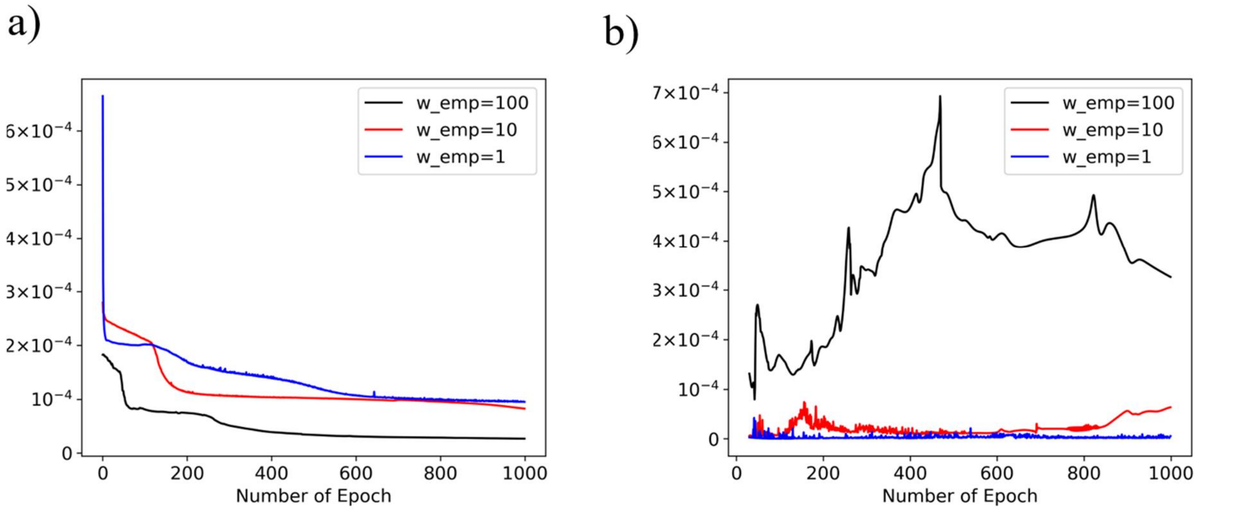


Figure 3.12 Loss graph for various empirical term weights: a) empirical loss of PINN model for several empirical term weights against number of epochs, b) physics loss of PINN model for several empirical term weights against number of epochs.

3.4.2.3 Similarity-based sampling strategy

When the data are selected randomly based on time and position, we face the problem of poor model accuracy when the state changes rapidly. Because our system starts from a stationary state, it changes rapidly at the beginning until it reaches a stationary state. This leads to degradation in the model performance, as shown in Figure 3.13. To address this issue, we introduce a similarity-based sampling strategy. This sampling strategy is based on weighted sampling proposed by Kalal et al. [61], where the data are sampled based on given weights. In this study, we defined such weights with data similarities.

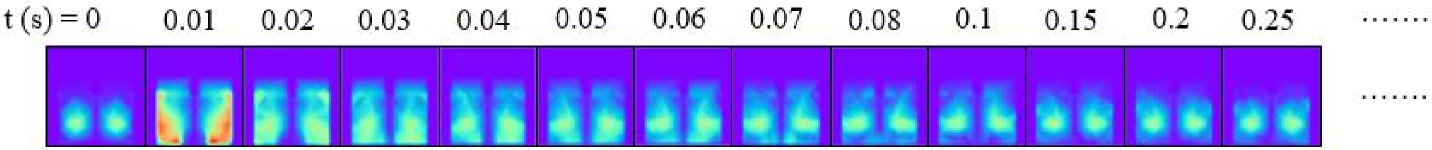


Figure 3.13 Velocity magnitude with time.

In a similarity-based sampling strategy, the number of sampled data in a certain domain is determined based on their similarity. We sample more significantly changing data, and we reduce the number of samples where the data points at different time points show a similar pattern. The similarities between the states are measured by the mean absolute error between adjacent times $\mathrm{t_i}$ and $\mathrm{t_{i+1}}$ for every state variable that is located at the same position as shown in Eq. (3-26) n is the number of data points for which similarity is calculated. The number of sampling data $\mathrm{N_i}$ for time index i is calculated using Eq. (3-27) and the distribution of the number of sampling data over time for the system is presented in Figure 3.14. As expected, most of the sampling data were populated during the early stage in time, and the number of samples decreased as the system reached a steady state.

$$S_{i,j} = \frac{1}{\frac{1}{n}\sum\left|y_{i+1}^{j} - y_{i}^{j}\right|} \tag{3-27}$$

$$j = u, v, w, f_a, f_b, f_c, k, \qquad i = time\ \ index$$

$$N_i = N * \frac{1}{\sum_{j=1}^{N_{var}} S_{i,j}} \tag{3-28}$$

As shown in Figure 3.15, the model with random sampling showed poor performance between 0 and 0.02 s. This is because the number of sampled data points was insufficient compared to the high variety in the data pattern. Therefore, the model with the weighted sampling showed a considerably higher accuracy during the time interval of 0–0.02 s compared with the random sampling model. Numerically, as shown in Figure 3.16, the model with similarity-based sampling achieved an $\mathrm{L_2}$ relative error of up to five times lower than that of the random sampling model at $5 * 10^{-4}$ s.

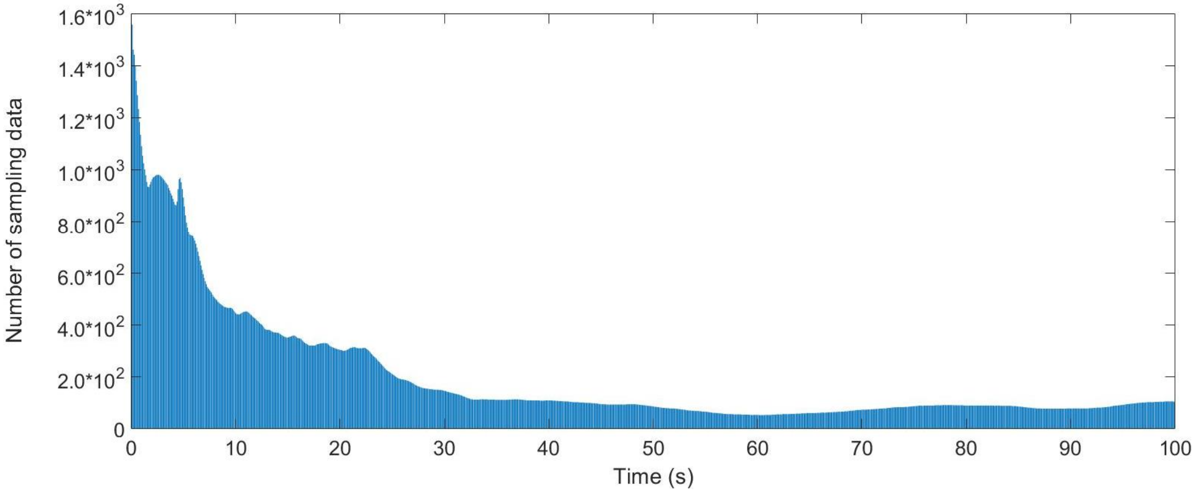


Figure 3.14 Number of sampling data over time ($\boldsymbol{total\ \ number\ \ of\ \ data\ \ sets = 2 * 10^5}$).

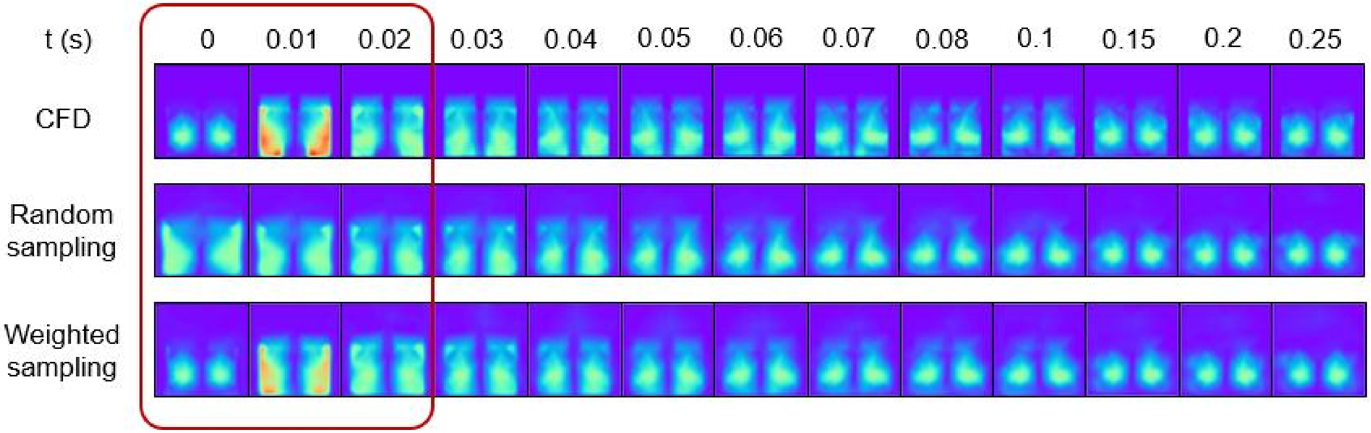


Figure 3.15 Comparison of random sampling and weighted sampling.

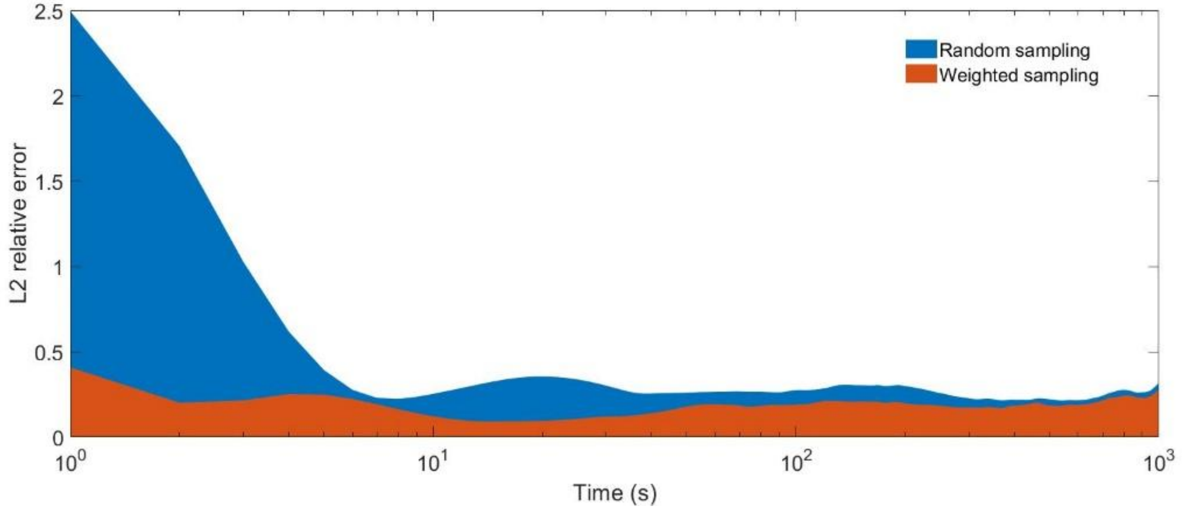


Figure 3.16 L2 relative error of random sampling and weighted sampling over time for velocity magnitude.

3.4.3 Comparison of PINN model with 1-D ODE model

In the previous section, we investigated the performance of PINN, which finds a solution comparable to CFD with training data and governing equations. In particular, in the reactor model, the inhomogeneous state due to the internal flow plays an important role in determining product quality. The CFD model can simulate the fluid flow, but it has a limitation in industrial applications because of its long calculation time. Although a simple one-dimensional (1-D) ODE model can be simulated in real time, it cannot consider the fluid flow inside the reactor. PINN resolves both the problems of the CFD and the ODE model.

In this section, we investigate the difference in the results of the 1-D ODE, CFD, and PINN models, and how each model has different implementation times. As shown in Figure 3.17, the molar concentration of component A calculated by the 1-D ODE model is smaller than that determined by the CFD and PINN models. This is because the components B, C, and D produced from component A are accumulated in the upper part of the reactor that cannot be stirred, hindering the reaction of component A. In the 1-D ODE model, states that are different from the actual phenomena are predicted because of the assumption of the homogenous phase. From the results in Figure 3.16, it is obvious that the fluid flow should not be ignored because it has a large effect on the reactor system state, such as temperature and product specification. However, as presented in Table 3.4, the CFD model takes $3.72 * 10^5$ s using 16 CPU cores of Intel® Xeon® CPU E5-2697 v2 @2.70GHz to implement the model that prevents its use in real-time applications. Even though it takes about $4 * 10^4$ s to train $2 * 10^5$ data sets, it takes very short time to implement the model after training. Therefore, the PINN model can be used for real-time optimization or control of a system that includes fluid flow. The PINN is 1300 times faster to implement than CFD, making it suitable for use in real-time problems; the accuracy of the PINN is also similar to that of CFD, as shown in Figure 3.17.

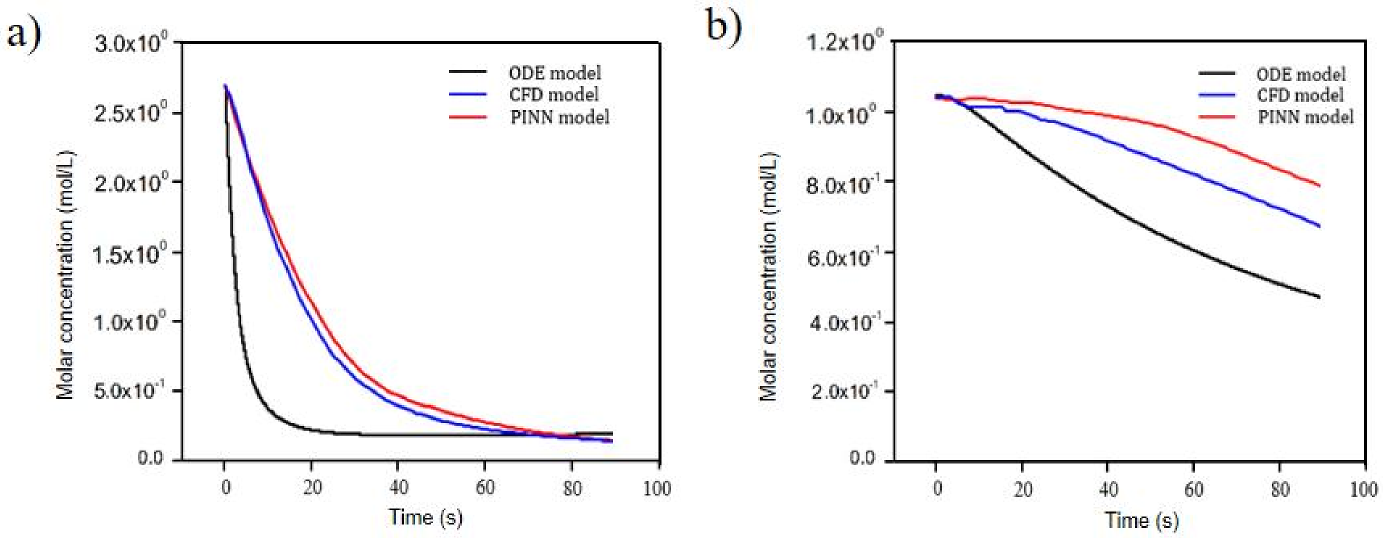


Figure 3.17 Comparison of ODE, CFD, and PINN model: a) average mole concentration of component A and b) average mole concentration of component B.

Table 3.4 Calculation times of ODE, PINN, and CFD. (Calculation time: total elapsed time during simulation).

Model	ODE	PINN	CFD
Calculation time (s)	$7.21 * 10^{-2}$	$2.86 * 10^{2}$	$3.72 * 10^{5}$

3.5 Conclusion

In this study, we investigated the possibility of using PINN to develop a surrogate model of CSTRs with the Van de Vusse reactions. We could find a solution that satisfies all the governing equations of mass, momentum, energy conservation, and species transport.

Because the system was governed by various types of physical equations and contained a multi-reference frame, we could not find a proper solution with the default setting of the PINN, which has been presented by Raissi [49]. Thus, a modified model framework that can implement a multi-reference frame system was introduced. In addition, we suggested three ways to enhance the performance of the PINN. First, we introduced mini-batch training. A full batch can only be used when the number of data is small be-cause it requires a large amount of memory. In addition, it is known that a large batch size leads to poor generalization. A small batch can provide a model with a stochastic effect that allows the model to explore a wider range. For our system, we confirmed that PINN also has the advantages of a mini-batch that contributes to faster convergence and better generalization. Second, we used a weighted loss function. This was necessary because the scale of the physics loss was considerably larger than the empirical loss term, which led to poor learning performance. After we set each term to a similar size, a more detailed adjustment of the weight was performed. We tested the effects of weights on the training results. The model showed a more detailed pattern when the weight of the empirical term was larger, while the model likely learnt a representative pattern when the weight of the empirical term was smaller. This could be adjusted according to the purpose of the model. However, it has a problem in that priori knowledge is required for determining the value of the weight because the scale of the physics loss terms is not known before the simulation. In future work, we will develop an adaptive learning method, which can adaptively determine the weight of each term during training. Third, we introduced a similarity-based sampling strategy. The concept of "similarity" was proposed and the number of sampling data was allocated in inverse proportion to the similarity. By using similarity-based sampling, the model accuracy was considerably improved in cases when a sudden change in the system state occurred.

The solution from PINN has the same fidelity as that of CFD with similar accuracy. CFD is the only way to simulate a system that involves fluid flow. However, CFD has a critical problem of a high computational cost, which makes it difficult to apply to various problems such as system design and optimization. We confirmed that the PINN can dramatically reduce the calculation time while maintaining the fidelity of the CFD. However, it is difficult to obtain as high accuracy as CFD with PINN, especially for the more complex the system. Therefore, the application of PINN is inevitably limited in application that requires as much accuracy as CFD.

In this work, we trained and developed the surrogate model for a single CFD case. We expect the model can be extended to a general surrogate model that can simulate various operating conditions with training on various cases. Like any other machine learning technique, PINN is hard to predict the completely outside of training domain. However, because the PINN learns physics equations, it will have a higher accuracy and would be more reliable in untrained regions than a black box model. This work shows that a system with complex flows can be simulated in real time and suggests the possibility that such a model can be used for system optimization and control.

3.6 Appendix

$$f_{cont} = u_x + v_y + w_z \tag{3-A1}$$

$$f_{u_{in}} = \rho\bigl(u_t + u_x u + u_y v + u_z w - \omega(u_x y - u_y x - v)\bigr) + p_x - \mu(u_{xx} + u_{yy} + u_{zz}) \tag{3-A2}$$

$$f_{u_{out}} = \rho\bigl(u_t + u_x u + u_y v + u_z w\bigr) + p_x - \mu(u_{xx} + u_{yy} + u_{zz}) \tag{3-A3}$$

$$f_{v_{in}} = \rho\bigl(v_t + v_x u + v_y v + v_z w - \omega(v_x y - v_y x + v)\bigr) + p_y - \mu(v_{xx} + v_{yy} + v_{zz}) \tag{3-A4}$$

$$f_{v_{out}} = \rho\bigl(v_t + v_x u + v_y v + v_z w\bigr) + p_y - \mu(v_{xx} + v_{yy} + v_{zz}) \tag{3-A5}$$

$$f_w = \rho\bigl(w_t + w_x u + w_y v + w_z w\bigr) + p_z - \mu(w_{xx} + w_{yy} + w_{zz}) \tag{3-A6}$$

$$\begin{aligned}
&f_{k_{in}} \\
&= \rho C_p \Bigl(T_t - \frac{p_t}{\rho} + (uu_t - \omega u_t y + vv_t + \omega v_t x + ww_t) + uT_x + vT_y + wT_z\Bigr) \\
&+ \rho(u(uu_x - \omega u_x y + vv_x + \omega v + ww_x) \\
&+ v\bigl(uu_y - \omega u_y y - \omega u + vv_y + \omega v_y x + ww_y\bigr) \\
&+ w(uu_z - \omega u_z y + vv_z + \omega v_z x + ww_z) - K(T_{xx} + T_{yy} + T_{zz})) - \mu((u - \omega y)(u_{xx} \\
&+ u_{yy} + u_{zz}) + (v + \omega x)(v_{xx} + v_{yy} + v_{zz}) + w(u_{xx} + u_{yy} + u_{zz}) + (u_y + v_x)^2 \\
&+ (u_z + w_x)^2 + (v_z + w_y)^2 + 2({u_x}^2 + {v_y}^2 + {w_z}^2))
\end{aligned} \tag{3-A7}$$

$$\begin{aligned}
&f_{k_{out}} \\
&= \rho C_p\bigl(k_t + uT_x + vT_y + wT_z\bigr) + up_x + vp_y + wp_z - K\bigl(T_{xx} + T_{yy} + T_{zz}\bigr) \\
&- \mu(u(u_{xx} + u_{yy} + u_{zz}) + v(v_{xx} + v_{yy} + v_{zz}) + w(u_{xx} + u_{yy} + u_{zz}) + (u_y + v_x)^2 \\
&+ (u_z + w_x)^2 + (v_z + w_y)^2 + 2({u_x}^2 + {v_y}^2 + {w_z}^2))
\end{aligned} \tag{3-A8}$$

$$\begin{aligned}
&f_{spec_{a_{in}}} \\
&= \Bigl({f_a}_t + (u - \omega y){f_a}_x + (v + \omega x){f_a}_y + w{f_a}_z\Bigr) - D\Bigl({f_a}_{xx} + {f_a}_{yy} + {f_a}_{zz}\Bigr) + f_a r_1 \\
&+ \frac{r_3 \rho}{Mw_a f_a^2 (2 - f_a - f_b - f_c)}
\end{aligned} \tag{3-A9}$$

$$\begin{aligned}
&f_{spec_{b_{in}}} \\
&= \Bigl({f_b}_t + (u - \omega y){f_b}_x + (v + \omega x){f_b}_y + w{f_b}_z\Bigr) - D\Bigl({f_b}_{xx} + {f_b}_{yy} + {f_b}_{zz}\Bigr) - f_a r_1 \\
&+ f_b r_2
\end{aligned} \tag{3-A10}$$

$$\begin{aligned}
&f_{spec_{c_{in}}} \\
&= \Bigl({f_c}_t + (u - \omega y){f_c}_x + (v + \omega x){f_c}_y + w{f_c}_z\Bigr) - D\Bigl({f_c}_{xx} + {f_c}_{yy} + {f_c}_{zz}\Bigr) - f_b r_2
\end{aligned} \tag{3-A11}$$

$$f_{spec_{a_{out}}} = \left(f_{a_t} + uf_{a_x} + vf_{a_y} + wf_{a_z}\right) - D\left(f_{a_{xx}} + f_{a_{yy}} + f_{a_{zz}}\right) + f_a r_1 + \frac{r_3 \rho}{Mw_a f_a^2 (2 - f_a - f_b - f_c)} \tag{3-A12}$$

$$f_{spec_{b_{out}}} = \left(f_{b_t} + uf_{b_x} + vf_{b_y} + wf_{b_z}\right) - D\left(f_{b_{xx}} + f_{b_{yy}} + f_{b_{zz}}\right) - f_a r_1 + f_b r_2 \tag{3-A13}$$

$$f_{spec_{c_{out}}} = \left(f_{c_t} + uf_{c_x} + vf_{c_y} + wf_{c_z}\right) - D\left(f_{c_{xx}} + f_{c_{yy}} + f_{c_{zz}}\right) - f_b r_2 \tag{3-A14}$$

$$y_x = \frac{\partial y}{\partial x} \tag{3-A15}$$

$$y_{xz} = \frac{\partial^2 y}{\partial x \partial z} \tag{3-A16}$$

$$r_1 = k_1 C_A \tag{3-A17}$$

$$r_2 = k_2 C_B \tag{3-A18}$$

$$r_3 = k_3 C_A^2 \tag{3-A19}$$

3.7 Notations

A	Pre-exponential factor
b^λ	Bias parameter at layer λ
C_p	Specific heat
C_{a_0}	Initial concentration of component A
C_{b_0}	Initial concentration of component B
C_{c_0}	Initial concentration of component C
C_{d_0}	Initial concentration of component D
C_{a_f}	Concentration of A at input flow
D	Diffusivity
E	Internal energy

E_r	Relative internal energy
E_a	Activation energy
f_a	Mole fraction of component A
f_b	Mole fraction of component B
f_c	Mole fraction of component C
f_d	Mole fraction of component D
f_i	Governing equations $\{\mathrm{f}_1\colon f_{cont}, f_2\colon f_{u_{in}}, f_3\colon f_{u_{out}}, f_4\colon f_{v_{in}}, f_5\colon f_{v_{out}}, f_6\colon f_w, f_7\colon f_{k_{in}}, f_8\colon f_{k_{out}}, f$ $f_{11}\colon f_{spec_{b_{in}}}, f_{12}\colon f_{spec_{c_{in}}}, f_{13}\colon f_{spec_{a_{out}}}, f_{14}\colon f_{spec_{b_{out}}}, f_{15}\colon f_{spec_{c_{o}}}$ $\}$
f_{cont}	Continuity equation
$f_{u_{in}}$	Momentum equation for x-direction for inner domain
$f_{u_{out}}$	Momentum equation for x-direction for outer domain
$f_{v_{in}}$	Momentum equation for y-direction for inner domain
$f_{v_{out}}$	Momentum equation for y-direction for outer domain
f_w	Momentum equation for z-direction
$f_{k_{in}}$	Energy conservation equation for inner domain
$f_{k_{out}}$	Energy conservation equation for outer domain
$f_{spec_{a_{in}}}$	Species tranport equation for component A for inner domain
$f_{spec_{b_{in}}}$	Species tranport equation for component B for inner domain
$f_{spec_{c_{in}}}$	Species tranport equation for component C for inner domain
$f_{spec_{a_{ou}}}$	Species tranport equation for component A for outer domain
$f_{spec_{b_{ou}}}$	Species tranport equation for component B for outer domain

$f_{spec_{C_{out}}}$	Species tranport equation for component C for outer domain
H_r	Relative total enthalpy
K	Thermal conductivity
MSE	Total loss
MSE_f	Total Physics loss
$MSE_{f_{con}}$	Continuity equation error
MSE_{f_u}	Momentum conservation equation error
MSE_{f_k}	Energy conservation equation error
$MSE_{f_{spe}}$	Species transport equation error
MSE_T	Output variable of temperature error
MSE_u	Output variable of velocity error
MSE_Y	Output variable of components mole fraction error
MSE_y	Total empirical error
Mw_A	Molecular weight
N_u	Number of data point
N_f	Number of colllocation point for f
N_{var}	Number of output variables
N	Number of total training data
p	Pressure
Q_f	Input mass flowrate
R_i	Net rate of the production of species i
$\vec{r}$	Position vector from the origin of the moving system
$S_{i,j}$	Similarity index at time i and component j

T	Temperature
T_0	Initial temperature of reactor
T_f	Input flow temperature
T_k	Wall temperature
u	x-direction velocity component
u_j	Ouput variables $\{u_1: u, u_2: v, u_3: w, u_4: k, u_5: f_a, u_6: f_b, u_7: f_c\}$
v	y-direction velocity component
$\vec{v}$	Absolute velocity vector
$\overrightarrow{v_r}$	Relative velocity vector
w	z-direction velocity component
$\vec{w}$	Rotating angular velocity
$\boldsymbol{W}^{\lambda}$	Weight parameter at layer λ
w_{emp}	Weight for empirical loss term
w_{phy}	Weight for physics loss term
X_{in}	Inner domain
X_{out}	Outer domain
x	x-direction location
Y_i	Mass fraction of i^{th} species
y	y-direction location
$\boldsymbol{y}(\boldsymbol{x}^i)$	Predicted output from the input x^i
$\boldsymbol{y}^i$	Exact solution of output variables at x^i
z	z-direction location
ρ	Density
μ	Viscosity

$\bar{\bar{\tau}}$	Stress tensor
$\bar{\bar{\tau}}_r$	Relative stress tensor
$\mathcal{N}$	Feed forward multi-layer NN
θ	Network parameter
σ^{λ}	Activation function at layer λ

Chapter 4
Concluding Remarks

4.1 Summary of contributions

In this thesis, computationally efficient strategies about two systems are introduced which can maintain model fidelity but effectively reduce the calculation time. The major contribution of each Chapter is listed below.

- Chapter 2 suggests new algorithm called Block KMC which is ~10 ti 32 times faster than traditional KMC. This algorithm is not limited to LDPE, and can be applied to any system that satisfies the following conditions. 1) Steady state system, 2) One reaction is much faster than the other reactions (e.g. propagation reaction). This work provides general strategy that can effectively simulate polymer kinetics and it makes it possible to perform the real time analysis of polymer reactor system which was very difficult before.

- Chapter 3 shows surrogate model development strategy of CFD-based reactor model using PINN. This is more robust and reliable model because physical laws are trained together with the data. This work provides guideline on how to develop a surrogate model for a reactor system through PINN. For example, how the model framework is designed for multi-reference frame system, how the network structure is define, how to define the loss function and training methodology for successful learning. With those contents it will be easier for other researchers to develop the reactor surrogate model using PINN.

4.2 Future work

As the process advances, the demand for high fidelity models to improve process efficiency is increasing. However, since the model accuracy and calculation time are always traded off, the more accurate models are made, the longer the calculation time takes. In this thesis, computationally efficient strategies about two reactor system is introduced which can maintain the fidelity but effectively reduce simulation time. From the first part of this work, a new methodology that can effectively simulate polymer kinetics was proposed. This solves the problem that it is almost impossible to consider the flow and kinetics together because of the high computational load due to the complex polymer reaction. Therefore, based on the research, it became easier to simulate a polymer reactor in which both flow and kinetic were considered, and the effect of flow on the polymer properties can be analyzed for the future work. From the second part of this work, more robust and reliable reactor surrogate model is developed through PINN. This model can be implemented in real time, so it is suitable for performing process optimization. However, for this work, surrogate model is developed for a single CFD case. The PINN model which was trained on a single CFD case cannot be implement properly for different operating conditions without any training for those conditions. It is expected the model can be extended to a general surrogate model that can simulate various operating conditions with training on various cases. It is also expected that the PINN can predict other conditions effectively for the conditions close to the learned operating conditions with higher accuracy than black-box model. The Ultimate goal is to develop a model that can simulate the industrial scale process in real time with high fidelity and to perform process optimization, control and finding optimal design using this model.

Reference

[1] F. C. Schilling, F. L. Mccrackin, H. L. Wagner, and F. A. Bovey, "Short-Chain and Long-Chain Branching in Low-Density Polyethylene," *Macromolecules*, vol. 535, no. 6, pp. 76–80, 1975.

[2] J. K. Beasley, "The Molecular Structure of Polyethylene. IV. Kinetic Calculations of the Effect of Branching on Molecular Weight Distribution," *J. Am. Chem. Soc.*, vol. 75, no. 24, pp. 6123–6127, 1953.

[3] W. H. Ray, "On the Mathematical Modeling of Polymerization Reactors," *J. Macromol. Sci. Part C*, vol. 8, no. 1, pp. 1–56, 1972, doi: 10.1080/15321797208068168.

[4] T. J. Crowley and K. Y. Choi, "Calculation of Molecular Weight Distribution from Molecular Weight Moments in Free Radical Polymerization," *Ind. Eng. Chem. Res.*, vol. 36, no. 5, pp. 1419–1423, 1997, doi: 10.1021/ie960623e.

[5] P. D. Iedema, M. Wulkow, and H. C. J. Hoefsloot, "Modeling molecular weight and degree of branching distribution of low-density polyethylene," *Macromolecules*, vol. 33, no. 19, pp. 7173–7184, 2000, doi: 10.1021/ma991711o.

[6] S. H. Son, J. J. Han, and J. M. Lee, "Modeling of the polymerization of linear monomers in the presence of multifunctional units," *Polymer (Guildf).*, vol. 126, pp. 74–86, 2017, doi: 10.1016/j.polymer.2017.08.011.

[7] B. M. Louie, G. M. Carratt, and D. S. Soong, "Modeling the free radical solution and bulk polymerization of methyl methacrylate," *J. Appl. Polym. Sci.*, vol. 30, no. 10, pp. 3985–4012, 1985, doi: 10.1002/app.1985.070301004.

[8] A. G. Mikos, C. G. Takoudis, and N. A. Peppas, "Kinetic Modeling of Copolymerization / Cross-Linking Reactions Mw," *Macromolecules*, vol. 19, no. 8, pp. 2174–2182, 1986.

[9] A. E. H. S. Zhu, "Chain-Length-Dependent Termination for Free Radical Polymerization," *Macromolecules*, vol. 22, no. 7, pp. 3093–3098, 1989.

[10] S. Zhu and A. E. Hamielec, "Gel formation in free radical polymerization via chain transfer and terminal branching," *J. Polym.*

Sci. Part B Polym. Phys., vol. 32, no. 5, pp. 929–943, 1994, doi: 10.1002/polb.1994.090320516.

[11] E. Mastan and S. Zhu, “Method of moments: A versatile tool for deterministic modeling of polymerization kinetics,” *Eur. Polym. J.*, vol. 68, pp. 139–160, 2015, doi: 10.1016/j.eurpolymj.2015.04.018.

[12] G. J. Wells and W. Harmon Ray, “Prediction of polymer properties in LDPE reactors,” *Macromol. Mater. Eng.*, vol. 290, no. 4, pp. 319–346, 2005, doi: 10.1002/mame.200400128.

[13] P. Pladis and C. Kiparissides, “A comprehensive model for the calculation of molecular weight-long-chain branching distribution in free-radical polymerizations,” *Chem. Eng. Sci.*, vol. 53, no. 18, pp. 3315–3333, 1998, doi: 10.1016/S0009-2509(98)00133-X.

[14] R. A. Hutchinson, “Modelling of chain length and long-chain branching distributions in free-radical polymerization,” *Macromol. Theory Simulations*, vol. 10, no. 3, pp. 144–157, 2001, doi: 10.1002/1521-3919(20010301)10:3<144::AID-MATS144>3.0.CO;2-A.

[15] S. X. Zhang and W. H. Ray, “Modeling of Imperfect Mixing and Its Effects on Polymer Properties,” *AIChE J.*, vol. 43, no. 5, pp. 1265–1277, 1997, doi: 10.1002/aic.690430515.

[16] H. Tobita, “Simultaneous long-chain branching and random scission: I. Monte Carlo simulation,” *J. Polym. Sci. Part B Polym. Phys.*, vol. 39, no. 4, pp. 391–403, 2001, doi: 10.1002/1099-0488(20010115)39:4<391::AID-POLB1011>3.0.CO;2-3.

[17] N. Yaghini and P. D. Iedema, “Molecular weight/branching distribution modeling of low-density-polyethylene accounting for topological scission and combination termination in continuous stirred tank reactor,” *Chem. Eng. Sci.*, vol. 116, pp. 144–160, 2014, doi: 10.1016/j.ces.2014.04.039.

[18] D. Meimaroglou, P. Pladis, A. Baltsas, and C. Kiparissides, “Prediction of the molecular and polymer solution properties of LDPE in a high-pressure tubular reactor using a novel Monte Carlo approach,” *Chem. Eng. Sci.*, vol. 66, no. 8, pp. 1685–1696, 2011, doi: 10.1016/j.ces.2011.01.003.

[19] D. M. Kim and P. D. Iedema, “Molecular weight distribution in low-density polyethylene polymerization; impact of scission mechanisms

in the case of a tubular reactor," *Chem. Eng. Sci.*, vol. 59, no. 10, pp. 2039–2052, 2004, doi: 10.1016/j.ces.2004.02.002.

[20] C. Kiparissides, A. Krallis, D. Meimaroglou, P. Pladis, and A. Baltsas, "From molecular to plant-scale modeling of polymerization processes: A digital high-pressure low-density polyethylene production paradigm," *Chem. Eng. Technol.*, vol. 33, no. 11, pp. 1754–1766, 2010, doi: 10.1002/ceat.201000325.

[21] P. D. Iedema and H. C. J. Hoefsloot, "Synthesis of branched polymer architectures from molecular weight and branching distributions for radical polymerisation with long-chain branching, accounting for topology-controlled random scission," *Macromol. Theory Simulations*, vol. 10, no. 9, pp. 855–869, 2001, doi: 10.1002/1521-3919(20011101)10:9<855::AID-MATS855>3.0.CO;2-A.

[22] D. M. Kim, M. Busch, H. C. J. Hoefsloot, and P. D. Iedema, "Molecular weight distribution modeling in low-density polyethylene polymerization; impact of scission mechanisms in the case of CSTR," *Chem. Eng. Sci.*, vol. 59, no. 3, pp. 699–718, 2004, doi: 10.1016/j.ces.2003.11.008.

[23] D. M. Kim and P. D. Iedema, "Modeling of branching density and branching distribution in low-density polyethylene polymerization," *Chem. Eng. Sci.*, vol. 63, no. 8, pp. 2035–2046, 2008, doi: 10.1016/j.ces.2007.12.033.

[24] N. Yaghini and P. D. Iedema, "Predicting molecular weight distribution by deterministic modeling and Monte Carlo simulations of radical polymerization with branching and scission allowing for multiradicals and gelation in various reactor configurations," *Chem. Eng. Sci.*, vol. 130, pp. 310–318, 2015, doi: 10.1016/j.ces.2015.03.003.

[25] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions with delays," *J. Chem. Phys.*, vol. 81, no. 25, pp. 2340–2361, 1977, doi: 10.1063/1.2710253.

[26] H. Tobita, "Molecular weight distribution in random crosslinking of polymer chains," *J. Polym. Sci. Part B Polym. Phys.*, vol. 33, no. 8, pp. 1191–1202, 1995, doi: 10.1002/polb.1995.090330805.

[27] H. Tobita, "Free-radical polymerization with long-chain branching and scission in a continuous stirred-tank reactor," *Macromol. React. Eng.*, vol. 7, no. 5, pp. 181–192, 2013, doi: 10.1002/mren.201200070.

[28] H. Tobita, “Continuous free-radical polymerization with long-chain branching and scission in a tanks-in-series model,” *Macromol. Theory Simulations*, vol. 23, no. 3, pp. 182–197, 2014, doi: 10.1002/mats.201300148.

[29] M. Rogošić, H. J. Mencer, and Z. Gomzi, “Polydispersity index and molecular weight distributions of polymers,” *Eur. Polym. J.*, vol. 32, no. 11, pp. 1337–1344, 1996, doi: 10.1016/S0014-3057(96)00091-2.

[30] E. Neuhaus *et al.*, “Modeling the polymeric microstructure of LDPE in tubular and autoclave reactors with a coupled deterministic and stochastic simulation approach,” *Macromol. Theory Simulations*, vol. 23, no. 7, pp. 415–428, 2014, doi: 10.1002/mats.201400014.

[31] C. Schütte and M. Wulkow, “A Hybrid galerkin-monte-carlo approach to higher-dimensional population balances in polymerization kinetics,” *Macromol. React. Eng.*, vol. 4, no. 9–10, pp. 562–577, 2010, doi: 10.1002/mren.200900073.

[32] D. Eckes and M. Busch, “Coupled Deterministic and Stochastic Modeling of an Industrial Multi-Zone LDPE Autoclave Reactor,” *Macromol. Symp.*, vol. 360, no. 1, pp. 23–31, 2016, doi: 10.1002/masy.201500115.

[33] D. Meimaroglou and C. Kiparissides, “A novel stochastic approach for the prediction of the exact topological characteristics and rheological properties of highly-branched polymer chains,” *Macromolecules*, vol. 43, no. 13, pp. 5820–5832, 2010, doi: 10.1021/ma1005233.

[34] P. Feucht, B. Tilger, and G. LUFT, “Prediction of molar mass distribution, number and weight average degree of polymerization and branching of low density polyethylene,” *Chem. Eng. Sci.*, vol. 40, no. 6, pp. 1935–1942, 1985.

[35] H. Tobita, “Model-based reactor design in free-radical polymerization with simultaneous long-chain branching and scission,” *Processes*, vol. 3, no. 4, pp. 731–748, 2015, doi: 10.3390/pr3040731.

[36] I. L. Chien, T. W. Kan, and B. S. Chen, “Dynamic simulation and operation of a high pressure ethylene-vinyl acetate (EVA) copolymerization autoclave reactor,” *Comput. Chem. Eng.*, vol. 31, no. 3, pp. 233–245, 2007, doi: 10.1016/j.compchemeng.2006.06.007.

[37] D. Meimaroglou, A. Krallis, V. Saliakas, and C. Kiparissides, “Prediction of the bivariate molecular weight - Long chain branching distribution in highly branched polymerization systems using Monte

Carlo and sectional grid methods," *Macromolecules*, vol. 40, no. 6, pp. 2224–2234, 2007, doi: 10.1021/ma0623439.

[38] N. H. Kolhapure and R. O. Fox, "CFD analysis of micromixing effects on polymerization in tubular low-density polyethylene reactors," *Chem. Eng. Sci.*, vol. 54, no. 15–16, pp. 3233–3242, 1999, doi: 10.1016/S0009-2509(98)00370-4.

[39] L. Marini and C. Georgakis, "The effect of imperfect mixing on polymer quality in low density polyethylene vessel reactors," *Chem. Eng. Commun.*, vol. 30, no. 6, pp. 361–375, 1984, doi: 10.1080/00986448408911139.

[40] B. Sanderse, S. P. van der Pijl, and B. Koren, "Review of computational fluid dynamics for windturbine wake aerodynamics," *Wind Energy*, vol. 14, no. 7, pp. 799–819, 2011, doi: 10.1002/we.

[41] F. J. Zajaczkowski, S. E. Haupt, and K. J. Schmehl, "A preliminary study of assimilating numerical weather prediction data into computational fluid dynamics models for wind prediction," *J. Wind Eng. Ind. Aerodyn.*, vol. 99, no. 4, pp. 320–329, 2011, doi: 10.1016/j.jweia.2011.01.023.

[42] C. K. Harris *et al.*, "Computational fluid dynamics for chemical reactor engineering," *Chem. Eng. Sci.*, vol. 51, no. 10, pp. 1569–1594, 1996, doi: 10.1016/0009-2509(96)00021-8.

[43] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Prog. Aerosp. Sci.*, vol. 45, no. 1–3, pp. 50–79, 2009, doi: 10.1016/j.paerosci.2008.11.001.

[44] S. Park, J. Na, M. Kim, and J. M. Lee, "Multi-objective Bayesian optimization of chemical reactor design using computational fluid dynamics," *Comput. Chem. Eng.*, vol. 119, pp. 25–37, 2018, doi: 10.1016/j.compchemeng.2018.08.005.

[45] Y. D. Lang, A. Malacina, L. T. Biegler, S. Munteanu, J. I. Madsen, and S. E. Zitney, "Reduced order model based on principal component analysis for process simulation and optimization," *Energy and Fuels*, vol. 23, no. 3, pp. 1695–1706, 2009, doi: 10.1021/ef800984v.

[46] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *J. Fluid Mech.*, vol. 807, pp. 155–166, 2016, doi: 10.1017/jfm.2016.615.

[47] J. Na, K. Jeon, and W. B. Lee, “Toxic gas release modeling for real-time analysis using variational autoencoder with convolutional neural networks,” *Chem. Eng. Sci.*, vol. 181, pp. 68–78, 2018, doi: 10.1016/j.ces.2018.02.008.

[48] G. Carleo *et al.*, “Machine learning and the physical sciences,” *Rev. Mod. Phys.*, vol. 91, no. 4, p. 45002, 2019, doi: 10.1103/RevModPhys.91.045002.

[49] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.

[50] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, “Physics-informed neural networks for high-speed flows,” *Comput. Methods Appl. Mech. Eng.*, vol. 360, p. 112789, 2020, doi: 10.1016/j.cma.2019.112789.

[51] X. I. A. Yang, S. Zafar, J. X. Wang, and H. Xiao, “Predictive large-eddy-simulation wall modeling via physics-informed neural networks,” *Phys. Rev. Fluids*, vol. 4, no. 3, p. 34602, 2019, doi: 10.1103/PhysRevFluids.4.034602.

[52] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, “NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations,” *J. Comput. Phys.*, vol. 426, p. 109951, 2021, doi: 10.1016/j.jcp.2020.109951.

[53] S. Kuntanapreeda and P. M. Marusak, “Nonlinear extended output feedback control for CSTRs with van de Vusse reaction,” *Comput. Chem. Eng.*, vol. 41, pp. 10–23, 2012, doi: 10.1016/j.compchemeng.2012.02.010.

[54] C. T. Chen and S. T. Peng, “Design of a sliding mode control system for chemical processes,” *J. Process Control*, vol. 15, no. 5, pp. 515–530, 2005, doi: 10.1016/j.jprocont.2004.11.001.

[55] F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson, “Nonlinear model-based control using second-order Volterra models,” *Automatica*, vol. 31, no. 5, pp. 697–714, 1995, doi: 10.1016/0005-1098(94)00150-H.

[56] K. Graichen, V. Hagenmeyer, and M. Zeitz, “Design of adaptive feedforward control under input constraints for a benchmark CSTR based on a BVP solver,” *Comput. Chem. Eng.*, vol. 33, no. 2, pp. 473–483, 2009, doi: 10.1016/j.compchemeng.2008.11.002.

[57] B. M. Åkesson and H. T. Toivonen, "A neural network model predictive controller," *J. Process Control*, vol. 16, no. 9, pp. 937–946, 2006, doi: 10.1016/j.jprocont.2006.06.001.

[58] G. A. Ridlehoover and R. C. Seagrave, "Optimization of Van de Vusse Reaction Kinetics Using Semibatch Reactor Operation," *Ind. Eng. Chem. Fundam.*, vol. 12, no. 4, pp. 444–447, 1973, doi: 10.1021/i160048a008.

[59] C. Meng, M. Sun, J. Yang, M. Qiu, and Y. Gu, "Training Deeper Models by GPU Memory Optimization on TensorFlow," *Proc. ML Syst. Work. NIPS*, no. Nips, pp. 1–8, 2017, [Online]. Available: http://learningsys.org/nips17/assets/papers/paper_18.pdf.

[60] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–16, 2016.

[61] Z. Kalal, J. Matas, and K. Mikolajczyk, "Weighted sampling for large-scale boosting," *BMVC 2008 - Proc. Br. Mach. Vis. Conf. 2008*, pp. 1–10, 2008, doi: 10.5244/C.22.42.

Abstract in Korean (국문초록)

화학 공학 분야의 많은 연구자들은 분석 및 계산 모델을 이용하여 시스템의 거동을 해석하고 최적화, 설계 및 제어를 수행하고 있다. 하지만 모델의 정확도와 계산시간은 거래되는 관계에 있어 계산시간이 오래걸리는 문제 때문에 모델의 정확도를 타협할 수 밖에 없는 실상이다. 이 학위논문에서는 두 시스템에 대해 모델의 충실도를 유지하면서도 계산시간을 효과적으로 줄일 수 있는 계산 효율적인 전략을 소개한다.

첫 번째 시스템은 고분자 반응기로 고분자의 반응에 중점을 두고 있다. 연속 교반 탱크 반응기에 대한 저밀도 폴리에틸렌의 분자량 분포를 예측하기 위해 모멘트 방법과 몬테 카를로 시뮬레이션 기법을 결합한 하이브리드 접근 방식이 제안되었다. 계산 효율적인 시뮬레이션을 위해 반복되는 반응들을 집합인 '블락'이라는 개념이 새로이 도입되었다. '블락 키네틱 몬테 카를로'라고 불리는 이 모델은 Neuhaus 가 제안한 모델보다 약 10~32 배 빠르다. 이 모델은 모든 정상 상태시스템에 적용할 수 있으며, 특정 반응이 다른 반응들보다 훨씬 빠른 경우에 계산 시간 감소효과를 누릴 수 있다. 또한 시스템의 운전 온도 및 개시제의 농도가 분자량 분포에 미치는 영향에 대해 사례 연구를 수행하였다. 180 개의 사례 연구 시뮬레이션을 바탕으로 분자량 분포가 숄더를 보이는 조건에 대한 정량적 지침을 제안하였다.

두번째 시스템은 전산유체역학 기반의 반응기 시스템이다. 전산유체역학은 유체의 흐름을 해석함에 있어 필수적인 기법이다. 특히 화학공학반응기에서 유체의 흐름은 내부의 온도나 농도에 큰 영향을 미친다. 그러나 전산유체역학은 계산시간이 오래걸린다는 단점으로 인해 실시간 최적화 및 공정 제어와 같은 응용에 사용이 제한된다. 이 학위논문에서는 시스템의 지배 방정식을 훈련할 수 있는 물리정보신경망(PINN)을 사용하여 Van de Vusse 반응이 포함된 연속 교반 탱크 반응기의 대리 모델을 개발한다. 화학 반응기 시스템이 따르는 모든 종류의 지배 방정식을 훈련할 수 있으며 다중 참조 프레임 시스템을 훈련 할 수 있는 물리정보신경망 모델 구조를 제안한다. 물리정보신경망(PINN)은 기존에 신경망 모델이 가지는 과적합 문제나 데이터가 많이 필요하다는 점 그리고 물리적 의미를 반영할 수 없다는 문제들을 해결할 수 있다. 또한 메모리 오류 및 모델의 발산 문제를 해결하기 위하여 Raissi 가 제안한 기존의 물리정보신경망(PINN)을 두 가지 방법으로 수정하였다. 1) 미니 배치 훈련; 2) 가중 손실 함수. 그리고 학습 데이터를 추출함에 있어 무작위 추출에 비해 정확도를 최대 5 배까지 향상시킬 수 있는 유사성 기반 추출 전략을 제안한다. 이 연구가 화학 공정의 고성능 대리 모델 개발을 위한 지침이 되기를 희망한다.

주요어: 높은 충실도 모델, 멀티 스케일 모델링, 분자량 분포, 저밀도 폴리에틸렌, 복합 모델, 모멘트 법, 키네틱 몬테 카를로 물리정보신경망, 대리 모델, 전산유체역학, 화학 반응기

학 번 : 2015-22822