KINETIC MODELING OF COMPLEX HETEROGENEOUSLY CATALYZED REACTIONS USING TEMPORAL ANALYSIS OF PRODUCT METHOD

Nurzhan Kushekbayev, Bachelor of Petroleum Engineering

Submitted in fulfillment of the requirements
for the degree of Masters of Science
in Chemical Engineering

School of Engineering
Department of Chemical Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Astana, Kazakhstan, 010000

Supervisor: Dr. Boris Golman,
Co-Supervisor: Dr. Stavros G. Poulopoulos

January 2018

Declaration

I hereby, declare that this manuscript, entitled "Kinetic modeling of complex heterogeneously catalyzed reactions using temporal analysis of product method", is the result of my own work except for quotations and citations, which I have duly acknowledged. I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Nurzhan Kushekbayev

Signature: Кушекбаев

Date: 24.01.2018

Abstract

The study of chemical interactions of reacting molecules on active sites of the catalyst plays an essential role in the heterogeneously catalyzed reactions. The complexity of heterogeneously catalyzed reactions makes it difficult to determine adsorbed intermediates formed during the reaction. The heterogeneously catalyzed reaction can undergo through more than one reaction mechanism with different rate laws. In this master thesis, the Temporal Analysis of Product (TAP) reactor model was used for numerical simulation of the multi-pathway reactions. To perform this, the mathematical model of TAP reactor was derived, and numerical simulation code was developed in the form of IPython notebook and verified with analytical solutions. Then numerical simulation algorithm was applied to simulate multipath CO oxidation in TAP reactor. The catalytic CO oxidation took place on ZnO catalyst via Langmuir-Hinshelwood mechanism, Eley-Rideal mechanism, and the combination of Langmuir-Hinshelwood and Eley-Rideal mechanism. The kinetic data for this reaction mechanism were taken from [1]. The simulation results for all cases indicate that production of CO_2 decreases as temperature increases, because of slow adsorption rate of O_2. Moreover, simultaneous Langmuir-Hinshelwood and Eley-Rideal mechanism was dominated by Langmuir-Hinshelwood reaction mechanism according to the simulation results.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Boris Golman for continuous support and guidance throughout my thesis research. Prof. Golman was very helpful in explaining and advising the way of solving issues that arose during research work.

I am also very thankful to Prof. Stavros Poulopoulos for his practical advises in improving my thesis, insightful comments, and prompt feedbacks.

Finally, I want to thank my family and friends who supported and motivated me in completing the MSc Program.

Table of Content

List of Figures

List of Tables

List of Abbreviations

ER	Eley-Rideal
LH	Langmuir-Hinshelwood
MOL	Method of Lines
ODE	Ordinary Differential Equations
PDE	Partial Differential Equations
TAP	Temporal Analysis of Product

Nomenclature

A_c cross-sectional area of reactor (cm^2)

C_A concentration of gaseous species A (mol/cm^3)

c_A dimensionless concentration of gas A

F_i molar flow of *i-th* species (mol/s)

$\bar{F}_A$ dimensionless flow of gas A at reactor outlet

D_k Knudsen diffusion coefficient (cm^2/s)

K_1 equilibrium constant

k^+ adsorption rate constants (cm^3/g cat s^{-1}),

k^- desorption rate constants (mol/g cat s^{-1})

$\bar{k}_1^+$ dimensionless adsorption constant

$\bar{k}_1^-$ dimensionless desorption constant

L length of reactor (cm)

M molecular weight (g/mol)

N_{pi} amount of substance of *i*-th species in pulse (mol)

n index for infinite series

P probability of finding inlet gas molecule at the exit of the reactor

R universal gas constant

r_i reaction rate of *i-th* species per weight of catalyst (mol/g cat s^{-1})

$\bar{r}$ average radius of pore (cm)

r_s radius of catalyst sphere (cm)

t time (s)

V reactor volume (cm^3)

W_{ix} flux of *i*-th species ($mol/cm^2\ s^{-1}$)

X conversion

x axial distance along the reactor length (cm)

Δx differential axial distance (cm)

Greek letters

δ_x delta function with respect to axial coordinate x

δ_z delta function with respect to dimensionless coordinate z

ε void fraction of catalyst bed

θ_v fractional coverage of empty sites

θ_A fractional coverage of adsorbed gas species A

ρ_b bulk density of catalyst bed (g cat/cm^3)

τ dimensionless time

$\Delta\tau_{valve}$ half of dimensionless opening time of pulse valve

u superficial velocity (cm/s)

Chapter 1 - Introduction

The catalytic reaction is a common phenomenon in our lives, which accompanies majority reactions in nature as well as in practical activity of humanity. The areas where catalytic reactions play an important role are petrochemical, chemical and biochemical processes, environmental mitigation of human activity, the creation of the new intermittent materials, development of a new source of energy and improvement of existing sources. Demonstrated field of application of heterogeneously catalyzed reactions requires more in-depth investigation of reaction mechanism and accompanying kinetics, and transport phenomena. As a result, the considerable attention is paid to surface intermediate interactions between molecules on active catalyst sites, where reactions are facilitated to break a molecular bond, thus reducing activation energy. The science of catalysis is the knowledge-intensive type of science, so this field of study can be combined with material science, surface chemistry, organic chemistry, non-organic chemistry, and chemical engineering [1].

In the modern world, the developed industrial countries are evaluated by the level of development of technology in the field of catalysis, because 90% of all current petrochemical and chemical processes are accomplished using a catalyst. Oil conversion ratio can reach about 90% in developed countries, whereas in Russia this value can reach about 70%. Besides, in the developed industrial countries catalytic

process creates about 30% of GDP. Since Kazakhstan's economy relies on oil and gas industry, Kazakhstan has high potential in the provision of petrochemical products. The most developed sector of Kazakhstan is the upstream industry, whereas the downstream (petrochemical) industry is still under development. The main drawbacks of oil refinery and petrochemical industry of Kazakhstan are low oil conversion ratio, low utilization capacity, depreciation of fixed capitals, low quality of oil products, and low efficiency of main catalytic processes [3]. The most promising catalytic reactions to be used in the chemical industry of Kazakhstan are deep hydrocracking, isomerization of n-alkanes, catalytic reforming, removal of aromatics, gas-to-liquid and Fisher-Tropsch processes. Therefore, it is essential to study catalytic reaction processes for the chemical industry of Kazakhstan to be competitive with the world standards as well as to develop proper innovative designs for industrial catalytic processes. Nevertheless, the development of a new catalyst requires expensive laboratory equipment, because the conventional kinetic modeling of the heterogeneous catalytic reactions are based on experimental measurements of rate laws for elementary reaction steps on catalysis surface. The recent development of information technology and computational methods allow simulating kinetic models using computers [4]. To do so firstly, it is necessary to choose appropriate reactors, where mass and heat transfer can be neglected, because these parameters can affect the reaction kinetics of heterogeneous catalytic reaction system.

Secondly, to understand deeply the chemistry, possible reaction mechanisms and detailed elementary reaction steps. Besides, the exact reaction mechanism in most case is unknown, and heterogeneous catalytic reaction can proceed via various path[5]. Summing up above points, temporal analysis of product method is the best technique to simulate kinetic models in computer. The design of the temporal analysis of product reactor is relatively simple and allow to eliminate the effect of heat and mass[6]. So objectives of this thesis is the simulation of the multipath heterogeneously catalyzed reaction using numerical model of TAP reactor. To simulate the multipath reaction, the oxidation of carbon monoxide is taken as an example, because it is simple and it has complex dynamic behavior [6-7]. As the reaction model, two common reaction mechanisms are chosen, namely Langmuir-Hinshelwood, Eley-Rideal, and combination of both mechanisms. In simulation, the single pulse mode of TAP reactor is used for various temperatures and pulse intensities. The simulation is carried out in 'one-zone-model', in other words reactor is filled with single catalyst bed.

The methodology used in the thesis is to derive a mathematical model for TAP reactor, simulate it using Python computational library, and qualitatively analyze the simulation results. Additionally, obtained numerical solution of the mathematical model of TAP reactor for different temperatures and pulse intensities can be useful to discriminate dominant reaction mechanism.

Chapter 2 - Literature review

2.1. Importance of transient experiments

In conventional reaction studies, the kinetic data are obtained in steady-state reactors, where the concentration of inlet and outlet streams are examined, and results of research can be correlated using rate laws and steady-state theories. The most of steady-state experiments for heterogeneous catalytic reactions are usually used to evaluate the performance of a catalyst. The complexity of the study of heterogeneous catalysis using steady-state technique is due to the variety of reaction steps and mechanisms on the active sites of the catalyst surface. The catalytic reaction can proceed through the following steps such as the external mass transfer from the bulk phase to the catalyst surface, diffusion of gaseous species in the porous catalyst space to the catalyst site, adsorption on the surface site, reaction on the surface, desorption, diffusion transfer of the product species to the catalyst, and external transfer into bulk fluid. Furthermore, the heat transfer effects and transport phenomena for different steps have different influence, whereas in transient experiments these effects are neglected [6]. Thus, the steady state reaction analysis does not depict the full picture of the reactant's interaction with catalyst surface. In contrast to steady-state experiments, the transient experiments will provide more

detailed information on reaction kinetics, intermediates, and the reaction mechanism for heterogeneous catalysis. Also, worth noting the heterogeneous catalyst reaction can undergo via various reaction path, which in turn each path has different elementary reaction steps. Azadi et al. in his research he suggested that Fischer-Tropsch reaction mechanism can comprise 128 elementary reactions [9]. In this reaction mechanism, every elementary reaction belongs to different reaction groups as: adsorption-desorption, monomer formation, chain growth, hydrogen abstraction, and water-gas shift. For this reason transient methods allow to discriminate each reaction path and reaction kinetics of each elementary steps[10].

Recently, non-steady state methods such as temporal analysis of products, temperature-programmed desorption, temperature-programmed reactions have become popular in heterogeneous catalyst study. These techniques can provide complete information on reaction kinetics for each elementary steps, reaction mechanism, and intermittent surface coverage desorption rate[11]. There are many transient experimental methods for determination of intrinsic reaction kinetics on heterogeneous catalyst, in thesis we will focus only on TAP method.

2.2. TAP experiments

The idea of TAP method was to support catalyst development and to interpret reaction steps. John Gleaves and his research team created the first prototype of TAP reaction system in 1980. Consequently, in 1983 the system was improved and two

reactors were built at the Monsanto company [12]. The main advantage of TAP reactor from other transient experiments is first, the short time resolution, which is enough for many simple reactions. Additionally, the time resolution of TAP reactor can be controlled by varying catalyst bed length and the width of the gas pulse. In other words, if we shorten the catalyst bed, the reactor residence time decreases, but time resolution increases. Varying the reactor and operational parameters enable us fully elucidate the mechanism of the catalytic reaction process, identify the sequence of reaction steps, define kinetic constants of elementary reactions, analyze active sites of catalyst etc.

Another significant advantage of TAP reactors is the operation under vacuum conditions, which allows us to neglect the external mass transfer limitations in reactor modeling. In conventional transient reactors, a carrier gas propagates the inlet pulse, while in TAP reactors the inlet pulse moves because of pressure gradient, and at higher intensities of a pulse, the Knudsen diffusion transports the gas molecules in the catalyst bed. In fact, the vital feature of Knudsen flow regime is that the diffusivities of individual components of the gas mixture are not affected by each other [13].

After introducing TAP response technique, the most of the publications focused on elucidating reaction mechanism, although the application of the TAP reactor is versatile for gas-solid interactions. In 1988 Gleaves and co-workers [11] published

the pioneering work, in which he developed the mathematical model based on PDEs describing chemical and transport processes in the reactor. These PDEs were solved analytically using the separation of variables method and moment analysis for defining the rate constants of adsorption and desorption. The analysis of the zeroth, first, or even second moments of product curves were applied to calculate the reaction conversion. Additionally, Gleaves described examples how to interpret TAP response curves with different kinetic constants. Zou *et al*. [14] studied the mathematical model proposed by Gleaves *et al*., verified the boundary conditions for PDE experimentally and validated the Knudsen diffusion in reactor.

After ten years Gleaves *et al.* [13] improved their experimental setup by changing the locations of catalyst in the reactor. In the given study, two deterministic TAP reactor models were analyzed for cases of diffusion, irreversible adsorption, and reversible adsorption. For the simple 'one-zone' model the reactor is filled with the sample of single catalyst, for complex three-zone-model, the catalyst is 'sandwiched' between the layers of non-reacting inert material. In this paper, Gleaves *et al*. introduced the dimensionless form of TAP reactor model and solved PDEs analytically using the separation of variables. The dimensionless flow at the exit of the reactor was derived as:

$$\bar{F}_A(z,\tau) = \pi \sum_{n=0}^{\infty} (-1)^n \left(2n+1\right) \exp(-(n+0.5)^2 \pi^2 \tau) \tag{2.1}$$

where $\bar{F}_A$ is the dimensionless flow of gas A at the reactor outlet, τ is the dimensionless time, and *n* is the index for infinite series.

Gleaves et al. also introduced the term "temporal probability density" and the statistical analysis was used to define the conversion of species. Thus, integration of 'distribution curve' in the interval (0,t) gives the probability of finding a molecule at the exit of the reactor:

$$P = \int_0^t F_A dt \tag{2.2}$$

where *P* is the probability of finding inlet gas molecule at the exit of the reactor. Integration of 'temporal distribution curve' in the interval from 0 to ∞ gives P=1. Therefore, the conversion X was defined as:

$$X = 1 - P \tag{2.3}$$

The number of studies was carried out to determine the adsorption rate constant. For instance, Olea *et al*. [15] studied CO oxidation on a newly developed Au/Ti(OH) catalyst to determine adsorption constants in the temperature range 298-473 K using TAP technique. In this study, transient responses resulted from the single-pulse experiments in the TAP reactor were analyzed using qualitative and quantitative approaches. To be precise, TAP experiments were analyzed by the statistical approach, and they adapted the analytical solution of PDEs to calculate the first-order adsorption/desorption rate constants. Consequently, the authors concluded that adsorption/desorption kinetics are identical for experimental and theoretical results.

Another methodology for estimation of transport and kinetics parameters was proposed by Tantake *et al.* [16]. In this study, authors aimed to compare regression analysis and moment analysis for different types of response curves, including exit flow rate curves and normalized responses. The percentage differences are used for the quantitative comparison of these parameters.

However, techniques mentioned above for TAP reactors are applicable only for the first order reaction rates, to normalize PDEs for the second order or higher order reaction rates the transformation of all variables to non-dimensionless form become cumbersome in the first place. Secondly, results are inaccurate, and finally, analytical solutions are not appropriate. Additionally, for more complicated reaction mechanisms as Langmuir-Hinshelwood or Eley-Redial, authors did not provide the explanation how these reaction mechanisms affect transient response curve.

2.3. Investigation of catalytic CO oxidation

Kobayashi *et al.* [17] started to apply the transient response methods to elucidate the mechanism of catalytic reactions. In this technique, mass and heat transfer effects, and the effects of back mixing and velocity distribution of gas flow in the catalyst bed are negligible. Initially, the feed concentration of gaseous reactant is held at constant, so the differential reactor was operated at steady state. Then, the feed concentration was changed stepwise and the transient responses of gaseous species were recorded at the reactor outlet to give information on reaction

mechanism. The authors studied the oxidation of CO and decomposition of N_2O on MnO_2, and showed the relationship between response curve and reaction mechanisms [17]. In subsequent research Kobayashi *et al.* [1] used the same technique to study CO oxidation on ZnO. In their research, the reaction mechanisms assumed to go through two different reaction paths which are progressing simultaneously on two different active sites. Authors divided reaction conditions into three regions with different pathways, namely Langmuir-Hinshelwood, Eley-Rideal, and combinations of Langmuir-Hinshelwood and Eley-Rideal. Results showed at low concentration of feed stream, the reaction undergo by Langmuir Hinshelwood route, whereas Eley-Rideal prevails at high concentration of inlet step. Golman [20] published an IPython notebook module to simulate transient step responses for multipath reactions.

The study was carried out by Redekop *et al.*[8], where the new strategy was developed to distinguish various reaction mechanisms using a decision tree. The proposed strategy was based on two concepts: the first one uses fractional coverage to express catalyst composition, the second one is to fit kinetic data to hypothetical reaction mechanisms. In this paper, the mathematical model of TAP reactor was solved forward in time by discretization on equally spaced grid point using Fortran code. As the model reaction Redekop *et.al* used multipath CO oxidation mechanism.

Mergler *et al.* [18] presented the study where oxidation of CO over a $Pt/CoO_x/SiO_2$ and CoO_x/SiO_2 catalyst is investigated experimentally using TAP technique. In the study authors purposed to analyze the information on the reaction mechanism of carbon monoxide oxidation on $Pt/CoO_x/SiO_2$ catalyst. Three types of experiments were carried out such as single pulse experiment, pump-probe experiment, multi-pulse experiment. In single pulse experiment, CO and O_2 are pulsed at the same time. In the pump-probe experiment, one of the reactants is pulse first, then second reactant is pulsed after the measured time interval. The analysis of single pulse transient curve for CoO_x/SiO_2 catalyst confirmed that upon increasing the temperature, more CO and O_2 disappeared, whereas less CO_2 formed. The similar trend was observed when pulse intensity ratio CO/O_2 was taken as 2. In other words, the single pulse response on $Pt/CoO_x/SiO_2$ catalyst showed that an increase of temperature above 100 ^oC results in decreasing CO_2 formation. Again, authors of this paper did not provide the full information regarding experimental setup, diffusion coefficients and kinetic data. Therefore, experimental results of this paper cannot be compared quantitively with our simulation results.

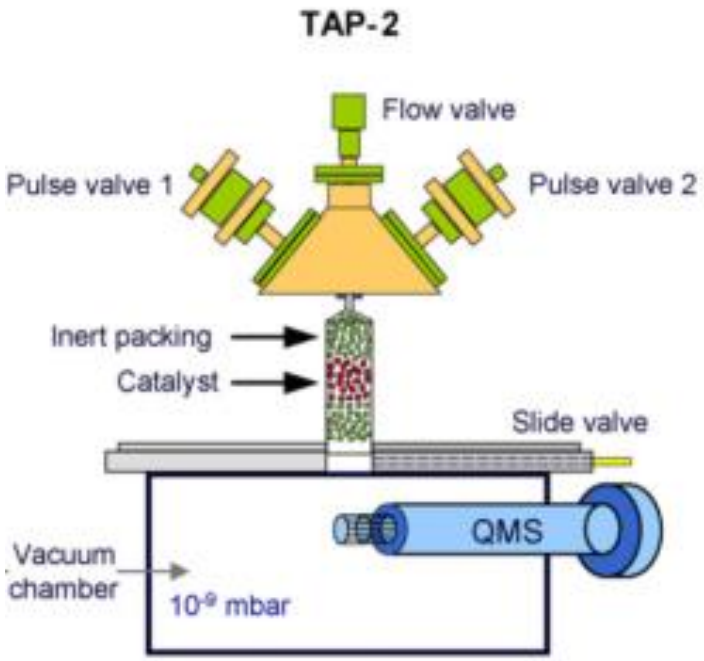
Chapter 3 - Methodology

3.1. TAP reactor description and operation

The TAP reactors have four principal components: high pulse valves, a catalytic microreactor, a high-vacuum system, and the quadrupole mass spectrometer (QMS). In this master thesis, the TAP-2 reactor is modeled. The distinguishing feature of TAP-2 reactor from the original TAP-1 reactor is that the high-speed pulse valves and microreactor are located outside the vacuum chamber as shown in Fig.1. This design allows excluding inaccuracy in measuring outlet pulses. The TAP microreactor itself represents a stainless steel cylinder of approximately 12.5 mm long and 6.4 mm in diameter. The microreactor is surrounded by a cooper sleeves, which is wounded with heater and cooling coil. The thermocouples are located in the cooper sleeves and they are connected to the computer. The temperature can be controlled in the interval from 100 ^{o}C to 600 ^{o}C. The microreactor is connected to the four high-speed valves mounted in the manifold. This manifold is electro-magnetically activated by wire coil with a short current pulse. During the short current pulse, an intense transient magnetic field is produced, which attracts magnetic disk attached to the valve stem. Thus, valve stem lifts out the valve so the small amount of gas pulse to flow out. Gas pulse enters the reactor, diffuses through

the catalyst in the Knudsen flow regime, reacts on the catalyst surface, and leaves the reactor. The mass spectrometry is used to analyze the outlet pulse.[12]

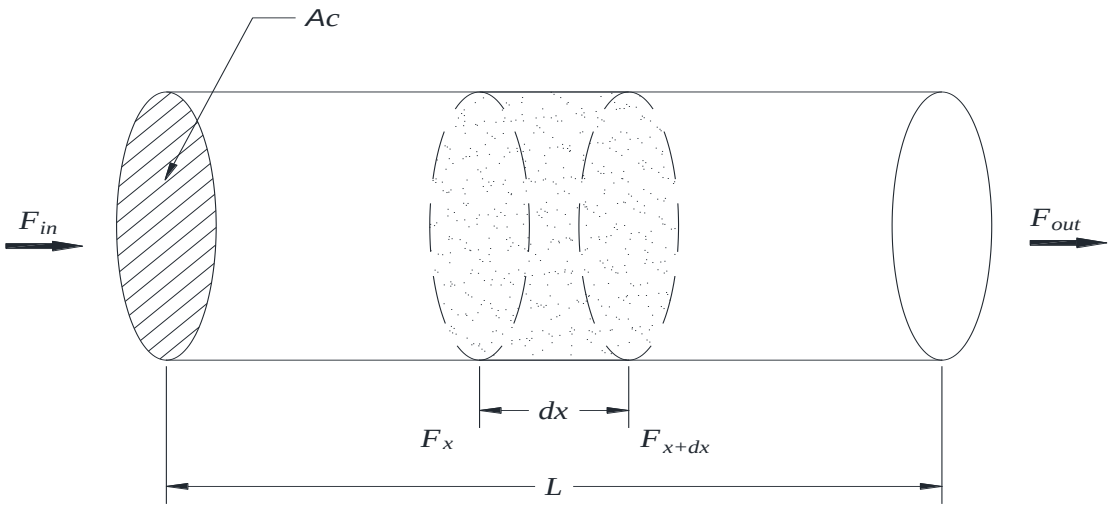
Figure 1 Simplified schematics of TAP-2 reactor[12]



3.2. Model development for TAP reactor

To have the better understanding of the reaction mechanism on the catalyst surface, it is necessary to derive the mathematical model for TAP reactors. For this purpose, we consider a fixed-bed reactor with cross-section area A_c, in which transient pulse is fed, as illustrated in Fig.2.

Figure 2 Illustration of fixed bed reactor



The general form of mass balance for a chemical reactor is given as:

In - Out + Generation - Consumption = Accumulation

The unsteady state mass balance for *i*-th gas species in the fixed-bed reactor is written as:

$$F_{i|x} - F_{i|x+\Delta x} + r_i \rho_b A_c \Delta x = \varepsilon_b \frac{dN_{pi}}{dt} \quad , \tag{2.4}$$

where $F_{i|x}$ is the molar flow of *i*-th species (mol/s), r_i is the reaction rate of *i*-th species per weight of catalyst (mol/g cat^{-1} s^{-1}), ρ_b is the bulk density of catalyst bed (g cat/cm^3) A_c is the cross-sectional area of reactor (cm^2), L is the length of reactor (cm), ε is the void fraction of catalyst bed, x is the axial distance along the reactor length (cm), Δx is the axial differential distance along the reactor length (cm), N_i is the number moles of *i*-th species (moles); and t is the time (s).

Dividing Eq. (2.4) by $A_c\Delta x$, which is equal to the reactor volume $V = A_c\Delta x$, we have the following equation:

$$-\frac{1}{A_c}\left(\frac{F_{i|x+\Delta x}-F_{i|x}}{\Delta x}\right)+r_i\rho_b=\frac{\varepsilon}{A_c\Delta x}\left(\frac{dN_{pi}}{dt}\right) \tag{2.5}$$

Taking the limit of Eq. (2.5) as Δx approaches 0, we obtain the following differential equation:

$$-\frac{1}{A_c}\left(\frac{\partial F_i}{\partial x}\right)+r_i\rho_b=\frac{\varepsilon}{A_c\Delta x}\left(\frac{\partial N_{pi}}{\partial t}\right) \tag{2.6}$$

For the right side of Eq.(2.6), we have the following expression: $\frac{1}{V}\frac{dN_{pi}}{dt}=\frac{dC_i}{dt}$, therefore Eq. (2.6) becomes:

$$-\frac{1}{A_c}\left(\frac{\partial F_i}{\partial x}\right)+r_i\rho_b=\varepsilon\frac{\partial C_i}{\partial t} \tag{2.7}$$

The molar flow rate of *i*-th gas phase species is given by:

$$F_i = A_c W_{i|x} \tag{2.8}$$

where W_{ix} is the flux of *i*-th species (mol/cm^2 s^{-1})

For constant total molar concentration, we have the following equation:

$$W_{i|x} = -D_i \frac{dC_i}{dx} + uC_i \tag{2.9}$$

where D_i is the diffusion coefficient, u is the superficial velocity.

The terms uC_i and $-D_i \frac{dC_i}{dx}$ in Eq.(2.9) represents the convective transport and diffusion, respectively. However, the convective transport can be neglected in the porous catalyst system where the pore radius is very small because the TAP reactors are operating under vacuum. Under these conditions, the Knudsen diffusion prevails, because the mean free path of the species is larger compared with the catalyst pore diameter. Whenever we observe the Knudsen diffusion, reacting components collide more often with pore walls than each other, and molecules of different species do not affect each other [19]. Considering above circumstances, we obtain the following equation:

$$D_k \frac{\partial^2 C_i}{\partial x^2} + r_i \rho_b = \varepsilon \frac{\partial C_i}{\partial t} \tag{2.10}$$

where D_k is the effective Knudsen diffusivity (cm^2/s)

In this mass balance equation, the adsorption/desorption and reaction kinetics are described by r_i term, depending on the reaction mechanism (Langmuir-Hinshelwood, Eley-Rideal etc.)

Eq.(2.10) in the case of non-reacting species (e.g. neon, argon and krypton) is given by [20]:

$$D_k \frac{\partial^2 C_i}{\partial x^2} = \varepsilon \frac{\partial C_i}{\partial t} \tag{2.11}$$

The input flux of gaseous species in TAP reactor is zero, because the pulse-valve is closed at t>0. As mentioned above, the typical TAP reactor is operated under vacuum conditions. Therefore, the concentration of all gaseous species at the reactor outlet is zero. These two statements determine the boundary conditions for inlet and outlet of the TAP reactor. Also, the inlet gas concentration is represented by the Delta Dirac function δ_x at *t=0*, because the inlet pulse width is much smaller than the time scale of the experiment. To sum up, above we have the following initial and boundary conditions [20]:

$$0 \le x \le L, t = 0, C_i = \delta_x \frac{N_{pi}}{\varepsilon A_c L} \tag{2.12}$$

$$x = 0, \frac{dC_i}{dx} = 0 \tag{2.13}$$

$$x = L, C_i = 0 \tag{2.14}$$

where N_{pi} is the number of pulsed molecules (mol).

Following equations describe the gas flow of *i*-th species at the reactor outlet:

$$F_i = -A_c D_k \frac{\partial C_i}{\partial x} \tag{2.15}$$

The fractional coverage of *i*-th species on catalyst site can be solved from a balance equation describing the mass balance of molecules on the surface of the catalyst bed:

$$\frac{\partial \theta_i}{\partial t} = k_1^+ C_i \theta_v - k_1^- \theta_i \tag{2.16}$$

where θ_i is the fractional coverage of adsorbed *i*-th species, k_1^+ is the rate at which *i*-th species attached to the surface, k_1^- is the rate at which *i*-th species detached from the surface, C_i is the concentration of *i*-th gas phase species. In other words, the first term on the right-hand side of the Eq.(2.16) describes the rate at which molecules are adsorbed onto the vacant site, and the second term describes the desorption rate from the occupied site [21].

As far as the irreversible adsorption on the surface of the catalyst are concerned the reaction mechanism undergoes through the following mechanism:

$$A + S \xrightarrow{k^+} AS$$

The surface coverage is negligible, due to small pulse intensity compared to the number of active sites of the catalyst bed, i.e. $\theta_v = 1$

The rate law of this equation is given by:

$$r_A = -k^+ C_A \tag{2.17}$$

where r_A is the rate of adsorption of gaseous reactant A on catalyst surface (mol/ g cat s^{-1}), k^+ is the adsorption rate constants (cm^3/g cat s^{-1}), and C_A is the concentration of gaseous species (mol/cm^3).

Substituting Eq. (2.17) in Eq.(2.10) results in:

$$\frac{\partial C_A}{\partial t} = D_k \frac{\partial^2 C_A}{\partial x^2} - \frac{\rho_b}{\varepsilon} k^+ C_A \tag{2.18}$$

with initial and boundary conditions:

$$0 \le x \le L, t = 0, C_A = \delta_x \frac{N_{pA}}{\varepsilon AL} \tag{2.19}$$

$$x = 0, \frac{dC_A}{dx} = 0 \tag{2.20}$$

$$x = L, C_A = 0 \tag{2.21}$$

Here we use D_k instead of $D_{k/}\varepsilon$ to be consistent with Fick's Law [11].

If <u>reversible adsorption/desorption</u> occurs on the catalyst surface, the reaction mechanism is followed by this path:

$$A + S \underset{k_1^-}{\overset{k_1^+}{\rightleftarrows}} AS$$

The mass balance of A in the gas phase is described by these two equations:

$$\frac{\partial C_A}{\partial t} = D_k \frac{\partial^2 C_A}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_1^+ C_A + k_1^- \theta_A) \tag{2.22}$$

$$\frac{\partial \theta_A}{\partial t} = k_1^+ C_A - k_1^- \theta_A \tag{2.23}$$

where θ_A is the fractional coverage of occupied sites with gas adsorbed species A. The initial and boundary conditions of these equations are given by Eq. (2.12)-(2.14).

For more complicated cases, when reversible adsorption-desorption and irreversible reactive species A in the gas phase is pulsed into the TAP reactor, thus obtaining B as the product, Eq. (2.22) and (2.23) need to be modified to include the reaction term into mass balance equations. The reaction mechanism is shown in the following scheme:

$$\begin{aligned} A + S &\underset{k_1^-}{\overset{k_1^+}{\rightleftarrows}} AS \\ AS &\xrightarrow{k_2} BS \\ BS &\underset{k_3^-}{\overset{k_3^+}{\rightleftarrows}} B + S \end{aligned} \tag{2.24}$$

The mass balances of this reaction mechanism are given by:

$$\frac{\partial C_A}{\partial t} = D_k \frac{\partial^2 C_A}{\partial x^2} + \frac{\rho_b}{\varepsilon}\left(-k_1^+ C_A + k_1^- \theta_A\right) \tag{2.25}$$

$$\frac{\partial \theta_A}{\partial t} = k_1^+ C_A - k_1^- \theta_A - k_2 \theta_A \tag{2.26}$$

$$\frac{\partial C_B}{\partial t} = D_k \frac{\partial^2 C_B}{\partial x^2} + \frac{\rho_b}{\varepsilon}\left(-k_3^+ \theta_B + k_3^- C_B\right) \tag{2.27}$$

$$\frac{\partial \theta_B}{\partial t} = k_3^+ \theta_B - k_3^- C_B + k_2 \theta_A \tag{2.28}$$

with initial and boundary conditions:

$$0 \le x \le L, t = 0, C_A = \delta_x \frac{N_{pA}}{\varepsilon AL} \tag{2.29}$$

$$x = 0; \frac{\partial C_A}{\partial x} = 0; \frac{\partial C_B}{\partial x} = 0 \tag{2.30}$$

$$x = L; C_A = 0; C_B = 0 \tag{2.31}$$

3.3. Numerical solution of PDEs

To solve TAP equations model described in the previous chapter, the system of PDEs defined for every component *i* is needed to be solved simultaneously. So far, the number of methods have been proposed for the numerical solution of PDEs, where transient models are described through two independent variables, namely time and space. For this particular model, we apply the technique known as the method of lines (MOL). In this numerical technique, simple transformations are applied to PDEs, by which each PDE is converted into the system of ordinary differential equations (ODEs). Usually, the partial derivative with respect to the space variable is chosen for approximation, whereas the derivative with respect to time remains. In our case, the axial distance of the reactor is divided into a set of grid points, where the derivatives are approximated by finite differences. The left-hand side of equations specifies the change of the concentration in time at every grid point. The general form a system of ODEs is given by:

$$\frac{dC_i^j}{dt} = f_i(\mathrm{C}_i^1, ..., \mathrm{C}_i^N), \mathrm{j} = 1, ..., \mathrm{N} \tag{2.32}$$

where C_i^j is the concentration of *i*-th component at grid point *j*, *j* is the grid point number. And N is the total number of the grid points [22].

3.4. Finite difference approximations

If we consider a grid equally spaced in *x*, where value of the *C(x)* is known at every point, then the first derivative of *C(x)* at grid point x_j is approximated through Taylor series. For $C(x_{j+1})$ Taylor series can be given by:

$$C\left(x_{j+1}\right)=C\left(x_j\right)+\frac{dC\left(x_j\right)}{dx}\Delta x+\frac{1}{2!}\frac{d^2C\left(x_j\right)}{dx^2}\Delta x^2\ldots \tag{2.33}$$

where $\Delta x=x_{j+1}-x_j$ is the grid point spacing. Likewise, the Taylor series for $C\left(x_{j-1}\right)$ can be given by:

$$C\left(x_{j-1}\right)=C\left(x_j\right)+\frac{dC\left(x_j\right)}{dx}\left(-\Delta x\right)+\frac{1}{2!}\frac{d^2C\left(x_j\right)}{dz^2}\left(-\Delta x\right)^2+\ldots \tag{2.34}$$

where $-\Delta x=x_{j-1}-x_{i.}$ Considering that the first derivative of $C\left(x_j\right)$ with respect to *x* is present in both equations, by expressing that derivative and subtracting Eq. (2.33) from (2.34) the second order central difference approximation is obtained:

$$\frac{dC\left(x_j\right)}{dx}=\frac{C\left(x_{j+1}\right)-C\left(x_{j-1}\right)}{2\Delta x} \tag{2.35}$$

In this formula, the first derivative of $C\left(x_j\right)$ respect to x_j is calculated by the dependent variables at grid point x_{j-1} and x_{j+1}, that are located at equal distances on either side of the point x_j. However, the main drawback of this method is that for

boundary conditions at points x_1 and x_n the imaginary points at x_0 and x_{n+1} need to be known. For this reason, the approximation formulas have to be derived for boundary conditions. This can be done in a similar way as above by writing Taylor series for $C(x_2)$ and $C(x_3)$. Thus, the second order forward approximation to the first derivative of $C(x_1)$ respect to x is given by:

at $j=1$:

$$\frac{dC(x_1)}{dx} = \frac{-3C(x_1)+4C(x_2)-C(x_3)}{2\Delta x} \tag{2.36}$$

The derivative of $C(x_N)$ with respect to x is acquired by writing Taylor series for $C(x_{N-1})$ and $C(x_{N-2})$.

at $j=N$:

$$\frac{dC(x_N)}{dx} = \frac{3C(x_N)-4C(x_{N-1})+C(x_{N-2})}{2\Delta x} \tag{2.37}$$

To summarize, the set of Eqs. (2.35)-(2.37) allows us to calculate the first derivative at all grid points [22].

The second order derivatives are derived as above by summation of Taylor series, Eqs. (2.33) and (2.34). Thus, the finite difference approximations for the second derivative of $C(x_i)$ with respect to x in inner points are obtained by [23]:

$$C\left(x_{j+1}\right)+C\left(x_{j-1}\right)=2C\left(x_j\right)+\frac{d^2C\left(x_j\right)}{dx^2}\Delta x^2+\ldots \tag{2.38}$$

$$\frac{d^2C\left(x_j\right)}{dx^2}=\frac{C\left(x_{j+1}\right)-2C\left(x_j\right)+C\left(x_{j-1}\right)}{\left(\Delta x\right)^2} \tag{2.39}$$

3.5. Numerical transformation of PDEs

In order to develop a general case, which can be applied for further complicated reaction mechanisms, we have chosen the reaction shown in scheme Eq.(2.24), which undergoes through TAP reactor described in Eq.(2.25)-(2.28) To make the first step by transforming these equations numerically, it is necessary to introduce a grid of uniformly spaced *N* points along the length of the reactor as:

$$0=x_0<x_1<x_2<x_3<\ldots<x_N=L$$

Thus, we have *N+1* grid points and N grid intervals. According to the next step, the second order spatial derivatives in the PDEs (2.25)-(2.28) are replaced by finite difference approximation formulas as Eq.(2.35). Thus, for each PDE we have the system of ODE equations for internal points:

$$\frac{dC_A\left(x_j\right)}{dt}=D_k\left(\frac{C_A\left(x_{j+1}\right)-2C_A\left(x_j\right)+C_A\left(x_{j-1}\right)}{\left(\Delta x\right)^2}\right)+\frac{\rho_b}{\varepsilon}\left(-k_1^+C_A\left(x_j\right)+k_1^-\theta_A\left(x_j\right)\right);\ \mathrm{j}=1,\ldots,\mathrm{N}-1$$

(2.40)

$$\frac{d\theta_A}{dt}(x_j)=k_1^+C_A\left(x_j\right)-k_1^-\theta_A\left(x_j\right)-k_2\theta_A\left(x_j\right);\ \mathrm{j}=0,\ldots,\mathrm{N} \tag{2.41}$$

$$\frac{dC_B\left(x_j\right)}{dt}=D_k\left(\frac{C_B\left(x_{j+1}\right)-2C_B\left(x_j\right)+C_B\left(x_{j-1}\right)}{\left(\Delta x\right)^2}\right)+\frac{\rho_b}{\varepsilon}\left(-k_3^+\theta_B\left(x_j\right)+k_3^-C_B\left(x_j\right)\right);\ \mathrm{j}=1,...,\mathrm{N}-1$$

(2.42)

$$\frac{d\theta_B}{dt}(x_j)=k_3^+\theta_B\left(x_j\right)-k_3^-C_B\left(x_j\right)+k_2\theta_B\left(x_j\right);\ \mathrm{j}=0,...,\mathrm{N} \tag{2.43}$$

The initial and boundary conditions described in the previous chapter completely determine the solution of PDE. The boundary conditions of TAP reactor are given by Eqs.(2.29)-(2.31). With the help of boundary conditions, we specify the first and the last points of spatial derivatives, in our particular case these are inlet and outlet of the reactor. The boundary condition at the inlet of the reactor for product A is defined as [24]:

$$-A_c\varepsilon D_k\frac{\partial C_A}{\partial x}|_{x=0}=N_{pi}\delta_x\left(t\right) \tag{2.44}$$

The finite difference approximation for point x_0 is defined by:

$$\frac{dC_A\left(x_0\right)}{dx}=\frac{-3C_A\left(x_0\right)+4C_A\left(x_1\right)-C_A\left(x_2\right)}{2\Delta x} \tag{2.45}$$

By substituting $\frac{\partial C_A}{\partial x}$ in Eq. (2.45) into Eq. (2.44) results in:

$$C_A\left(x_0\right)=\frac{4C_A\left(x_1\right)}{3}-\frac{C_A\left(x_2\right)}{3}+\frac{N_{pi}\delta_x\left(t\right)2\Delta x}{3A_c\varepsilon D_k} \tag{2.46}$$

This equation is used to evaluate the concentration of A at the inlet of reactor. However, the product B, which is not present in the initial pulse, is evaluated at the inlet of the reactor by:

$$C_B(x_0) = 0 \tag{2.47}$$

The boundary conditions for A and B components at the reactor outlet, for the last grid point x_N, are defined by following two conditions:

$$\begin{aligned} C_A(x_N) &= 0 \\ C_B(x_N) &= 0 \end{aligned} \tag{2.48}$$

Exit flow rate of component *i* can be defined by Eq. (2.15):

$$F_i = -A_c D_k \frac{\partial C_i}{\partial x}\Big|_{x=L} \tag{2.49}$$

By inserting above derived boundary conditions for $\frac{\partial C_i}{\partial x}$ at point L Eq.(2.37), we have following equation for exit flow rate:

$$F_i = -A_c D_k \frac{\partial C_i}{\partial x}\bigg|_{x=L} = -A_c D_k \left(\frac{3C_i(x_N) - 4C_i(x_{N-1}) + C_i(x_{N-2})}{2\Delta x} \right) \tag{2.50}$$

As we know that at last point concentration of $C_i(x_N)$ is equal to zero:

$$F_i = -A_c D_k \frac{\partial C_i}{\partial x}\bigg|_{x=L} = -A_c D_k \left(\frac{-4C_i(x_{N-1}) + C_i(x_{N-2})}{2\Delta x} \right) \tag{2.51}$$

In our model algorithm, we divided this value to N_{pi} to normalize this simulation results.

3.6. Derivation of model reaction

The oxidation of CO was chosen to simulate as a model reaction, due to its diverse kinetic behavior as well as its importance in numerous applications. For instance, CO oxidation is applied in cold start exhaust emission control for

automobiles[25]. Derivation of rate equations for oxidation of CO are given in (Golman 2014), where two different active sites identified, S_1 and S_2. In first path reaction mechanism assumed to proceed via Langmuir-Hinshelwood (L-H), the second path is progressed according to Eley Rideal (E-R) reaction mechanism. The overall reaction is:

$$2CO + O_2 \rightarrow 2CO_2 \tag{3.1}$$

Reaction Path I (L-H reaction mechanism)

$$\begin{aligned} &CO(g) + S_1 \underset{k_1^-}{\overset{k_1^+}{\rightleftarrows}} CO*S_1 \\ &O_2(g) + 2S_1 \underset{k_2^-}{\overset{k_2^+}{\rightleftarrows}} 2O*S_1 \\ &CO*S_1 + 2O*S_1 \xrightarrow{k_3} CO_2(g) + 2S_1 \end{aligned} \tag{3.2}$$

For our model problem, the rate law for each elementary reaction needs to be considered, which may comprise several mechanisms. The rate expression for adsorption of CO on catalyst surface can be defined as:

$$r_1 = -k_1^+ C_{CO} + k_1^- \theta_{CO*S_1} \tag{3.3}$$

The absence of fraction of vacant site in Eq. (3.3) is due to low pulse intensity at the reactor inlet, which means that the concentration of pulsed molecules in short period of time is not enough to occupy whole catalyst active sites, so the fraction of vacant sites in TAP reactors can be taken as 1. In other words, the number of adsorption site

is much higher than the number of adsorbed molecules. The following equations describes the rate expression for adsorption of O_2:

$$r_2 = -k_2^+ C_{O_2} + k_2^- \theta_{O^*S_1}^2 \tag{3.4}$$

The principal difference between these two rate expressions, Eqs. (3.3) and (3.4), is that adsorption of CO is carried through associative adsorption, whereas O_2 is adsorbed through dissociative adsorption [7]. The rate expression for the formation of CO_2 is given by:

$$r_3 = -k_3 \theta_{CO^*S_1} \theta_{O^*S_1} \tag{3.5}$$

The overall mass balances of gas components for Langmuir-Hinshelwood mechanism pathway can be expressed in following equations:

$$\frac{\partial C_{CO}}{\partial t} = D_k \frac{\partial^2 C_{CO}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_1^+ C_{CO} + k_1^- \theta_{CO^*S_1}) \tag{3.6}$$

$$\frac{\partial C_{O_2}}{\partial t} = D_k \frac{\partial^2 C_{O_2}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_2^+ C_{O_2} + k_2^- \theta_{CO^*S_1}^2) \tag{3.7}$$

$$\frac{\partial C_{CO_2}}{\partial t} = D_k \frac{\partial^2 C_{CO_2}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_3 \theta_{CO^*S_1} \theta_{O^*S_1}) \tag{3.8}$$

The following equations define the mass balances for surface intermediates regarding fractional coverage:

$$\frac{\partial \theta_{CO^*S_1}}{\partial t} = k_1^+ C_{CO} - k_1^- \theta_{CO^*S_1} - k_3 \theta_{CO^*S_1} \theta_{O^*S_1} \tag{3.9}$$

$$\frac{\partial \theta_{O^*S_1}}{\partial t} = 2k_1^+ C_{O_2} - 2k_1^- \theta_{O^*S_1}^2 - k_3 \theta_{CO^*S_1} \theta_{O^*S_1} \tag{3.10}$$

Reaction Path I (E-R reaction mechanism):

$$\begin{aligned} &O_2(g) + 2S_1 + 2e^- \underset{k_4^-}{\overset{k_4^+}{\rightleftarrows}} 2O^- * S_2 \\ &CO(g) + O^- * S_2 \xrightarrow{k_5} CO_2(g) + S_2 + e^- \end{aligned} \tag{3.11}$$

We use the similar method to express the rate law for elementary steps as:

$$r_4 = -k_4^+ C_{O_2} + k_4^- \theta_{O^-*S_2}^2 \tag{3.12}$$

$$r_5 = -k_5 C_{CO} \theta_{O^-*S_2} \tag{3.13}$$

The mass balance of gas components for Eley-Rideal reaction mechanism can be expressed:

$$\frac{\partial C_{O_2}}{\partial t} = D_k \frac{\partial^2 C_{O_2}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_4^+ C_{O_2} + k_4^- \theta_{O^-*S_2}^2) \tag{3.14}$$

$$\frac{\partial C_{CO_2}}{\partial t} = D_k \frac{\partial^2 C_{CO_2}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_5 C_{CO} \theta_{O^-*S_2}) \tag{3.15}$$

$$\frac{\partial C_{CO}}{\partial t} = D_k \frac{\partial^2 C_{CO}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_5 C_{CO} \theta_{O^-*S_2}) \tag{3.16}$$

For Eley-Rideal reaction mechanism only one surface intermediate is present, namely oxygen anion, so the mass balance of adsorbed species is:

$$\frac{\partial \theta_{O^-*S_2}}{\partial t} = 2k_4^+ C_{O_2} - 2k_4^- \theta_{O^*S_2}^2 - k_5 C_{CO} \theta_{O^*S_2} \tag{3.17}$$

If the oxidation of CO on catalyst surface is progressing simultaneously through both reaction mechanism, but on different active sites, then the overall reaction model is defined by following equations:

$$\frac{\partial C_{CO}}{\partial t} = D_k \frac{\partial^2 C_{CO}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_1^+ C_{CO} + k_1^- \theta_{CO*S_1} - k_5 C_{CO} \theta_{O^-*S_2})) \tag{3.18}$$

$$\frac{\partial C_{O_2}}{\partial t} = D_k \frac{\partial^2 C_{O_2}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_2^+ C_{O_2} + k_2^- \theta_{CO*S_1}^2 - k_4^+ C_{O_2} + k_4^- \theta_{O^-*S_2}^2) \tag{3.19}$$

$$\frac{\partial C_{CO_2}}{\partial t} = D_k \frac{\partial^2 C_{CO_2}}{\partial x^2} + \frac{\rho_b}{\varepsilon}(-k_3 \theta_{CO*S_1} \theta_{O*S_1} - k_5 C_{CO} \theta_{O^-*S_2}) \tag{3.20}$$

The kinetic constants obtained from [1] will be used for CO oxidation on the catalyst surface. The reaction path for both reaction mechanisms are given by Eq.(3.2) and Eq.(3.11). Kobayashi *et.al* [22] provided kinetic parameters of CO oxidation on ZnO catalyst (Kadox 25 New Jersey Zinc Co). The rate expression for this catalyst was derived using steady-state analysis via Hougen-Watson procedure as:

$$r_{L-H} = \frac{k_3 K_1 K_2^{0.5} P_{CO} P_{O_2}^{0.5}}{(1 + K_1 P_{CO} + K_2^{0.5} P_{O_2}^{0.5})} \tag{3.21}$$

$$r_{E-R} = \frac{k_5 K_4^{0.5} P_{CO} P_{o_2}^{0.5}}{1 + K_4^{0.5} P_{o_2}^{0.5}} \tag{3.22}$$

$$r_{total} = r_{E-R} + r_{L-H} \tag{3.23}$$

The kinetic constants of ZnO catalyst given in Table 1 [22]. The rate constant in Table 1 is given in terms of partial pressure of species. The reversible desorption rate constant K_1 , K_2 and K_2 are given in terms of equilibrium constants, and it can be calculated by simple formula. In our case the TAP reactor model is derived in terms of concentration on reacting species. Hence, kinetic constants given in Table 1 should be multiplied by R*T*60. Results of this operation is given in Table 2.

Table 1. Rate constants for reaction 1, 2 and 4 [1]

T [^{o}C]	k_1 , k_2 [mol g^{-1} min^{-1} atm^{-1}]	K_1 [atm^{-1}]	K_2 [atm^{-1}]	k_3 [mol g^{-1} min^{-1} atm^{-1}]	k_4 [mol g^{-1} min^{-1} atm^{-1}]	K_4 [atm^{-1}]	k_5 [mol g^{-1} min^{-1} atm^{-1}]
130	$3.88*10^{-5}$	249	382	$7.76*10^{-8}$	$9.25*10^{-6}$	105	$1.85*10^{-8}$
140	$8.10*10^{-5}$	198	208	$1.62*10^{-7}$	$2.63*10^{-5}$	30.4	$5.26*10^{-8}$
150	$1.73*10^{-4}$	145	113	$3.46*10^{-7}$	$6.85*10^{-5}$	10.1	$1.37*10^{-7}$
160	$4.01*10^{-4}$	122	65.1	$8.02*10^{-7}$	$1.63*10^{-4}$	3.1	$3.25*10^{-7}$
170	$7.5*10^{-4}$	103	34.8	$1.5*10^{-6}$	$4.25*10^{-5}$	1.14	$8.5*10^{-7}$

Table 2. Converted rate constants

T [°C]	k_1^+, [cm^3/g cat s^{-1}]	$k_{1\ 1}^-$ [mol/g cat s^{-1}]	k_2^+ [cm^3/g cat s^{-1}]	k_2^- [mol/g cat s^{-1}]	k_3 [cm^3/g cat s^{-1}]	k_4^+ [cm^3/g cat s^{-1}]	k_4^- [mol/g cat s^{-1}]	k_5 [cm^3/g cat s^{-1}]
130	77	0,309	77	0,202	0,154	18,4	0,175	0,0367
140	165	0,832	165	0,792	0,329	53,5	1,76	0,107
150	360	2,48	360	3,19	0,721	143	14,1	0,285
160	855	7,01	855	13,1	1,71	347	112	0,693
170	1640	15,9	1640	47	3,27	927	813	1,85

3.7. Simulation environment

For simulation of the reaction process in the TAP reactor, at first we need to specify constant values, which are close to real cases. Therefore, we are going to use approximate values from resembling experimental techniques. The effective Knudsen diffusion can be calculated by:

$$D_k = \varepsilon^2 \frac{2\bar{r}}{3}\sqrt{\frac{8RT}{\pi M}} \tag{3.24}$$

where $\bar{r}$ is the average radius of pore, M is the molecular weight, R is the universal gas constant, and T is the temperature. The average pore radius is given by:

$$\bar{r} = \frac{2\varepsilon}{3(1-\varepsilon)} r_s \tag{3.25}$$

where r_s is the radius of catalyst sphere.

Gleaves *et al.* [11] calculated the value of effective Knudsen diffusion as D_k=33 cm^2/s for carbon dioxide at 400 ^{o} C, 0.3 mm diameter particles, and void fraction of 0.38. This value is close to measured Knudsen diffusion for carbon monoxide pulse over 25-40 mm catalyst particle. We used in our simulations the microractor with diameter of 2 cm and length of 2.54 cm. TAP reactor will be based on this information. IPython code solver does not work for pulse intensity over 10^7. Therefore the single pulse simulations are carried out for pulse intensities of 10^4, 10^5, and 10^6

Chapter 4 - Results and discussion

4.1. Model verification

In this section, to verify the numerical solution algorithm of TAP reactor model the comparison of numerical results with analytical solution will be carried out. To pursue this, the analytical solution by Gleaves *et al*. [20] is applied to verify the correctness of the model. The cases of non-adsorbing gases (diffusion), irreversible adsorption, and reversible adsorption will be investigated.

Case 1. Non-reacting gas

The mass balance for non-reacting gas A in TAP reactor is given by Eq.(2.11) as:

$$D_k \frac{\partial^2 C_A}{\partial x^2} = \varepsilon \frac{\partial C_A}{\partial t} \tag{4.1}$$

An initial and boundary conditions are defined by:

$$0 \le x \le L, t = 0, C_i = \delta_x \frac{N_{pi}}{\varepsilon A_c L} \tag{4.2}$$

$$x = 0, \frac{\partial C_i}{\partial x} = 0 \tag{4.3}$$

$$x = L, C_i = 0 \tag{4.4}$$

The measured flow rate, F_A , at the reactor exit is described by:

$$F_A = -A_c D_k \frac{\partial C_A}{\partial x}\Big|_{x=L} \tag{4.5}$$

Let's transform Eq.(2.11) into dimensionless form by introducing the dimensionless axial coordinate, $z = \frac{x}{L}$, concentration $c_A = \frac{C_A}{N_{pi} / \varepsilon A_c L}$, and time $\tau = \frac{tD_k}{\varepsilon L^2}$, Eq.(4.1) can be written in dimensionless form as:

$$\frac{\partial c_A}{\partial \tau} = \frac{\partial^2 c_A}{\partial z^2} \tag{4.6}$$

with the dimensionless initial conditions:

$0 \le z \le 1, \tau = 0 : c_A = \delta_z$,

and the boundary conditions:

$z = 0 : \frac{\partial c_A}{\partial z} = 0$

$z = 1 : c_A = 0$

The analytical solution of this equation for c_A can be evaluated by method of separation of variables, which is expressed in the following form[2]:

$$c_A(z,\tau) = 2\sum_{n=0}^{\infty} \cos\left(\left(n+\frac{1}{2}\right)\pi z\right) \exp\left(-\left(n+\frac{1}{2}\right)^2 \pi^2 \tau\right) \tag{4.7}$$

The dimensionless flow rate is expressed as:

$$\bar{F}_A(z,\tau) = -\frac{\partial c_A(z,\tau)}{\partial z} = \pi \sum_{n=0}^{\infty} (-1)^n (2n+1) \sin\left((n+0.5)\pi z\right) exp\left(-(n+0,5)^2 \pi^2 \tau\right) \tag{4.8}$$

The dimensionless flow rate at the exit of the reactor, when z=1:

$$\bar{F}_A(z,\tau) = \pi \sum_{n=0}^{\infty} (-1)^n (2n+1) exp\left(-(n+0,5)^2 \pi^2 \tau\right) \tag{4.9}$$

To solve Eq.(4.6) numerically by the MOL, the boundary conditions need to be specified at first and last grid points. For the concentration at the inlet of the reactor we have the following boundary conditions[24]:

$$-A_c \varepsilon D_k \frac{\partial C_A}{\partial x}|_{x=0} = N_{pi} \delta_x(t) \tag{4.10}$$

By introducing dimensionless transformations as in above equations, the boundary conditions for inlet of the reactor become:

$$\frac{\partial c_A(z=0)}{\partial z} = -\delta_z(\tau) \tag{4.11}$$

where dimensionless $\delta_z(\tau)$ can be defined for the inlet pulse as:

$$\delta_z(\tau) = \frac{1}{2\Delta\tau_{valve}}(1+\sin(\frac{\pi}{\Delta\tau_{valve}}(\tau - \frac{\Delta\tau_{valve}}{2}))), \quad 0 \le \tau \le 2\Delta\tau_{valve} \tag{4.12}$$

or by the finite Gaussian pulse:

$$\delta_z(\tau) = \frac{1.3027}{\Delta\tau_{valve}} \exp\left(-\frac{4(\tau - \Delta\tau_{valve})^2}{\tau(2\Delta\tau_{valve} - \tau)}\right), \quad 0 \le \tau \le 2\Delta\tau_{valve} \tag{4.13}$$

here $\Delta\tau_{valve}$ is the half of dimensionless opening time of pulse valve[24].

For the dimensionless concentration at the outlet of the reactor:

$$c_A(z=1) = 0 \tag{4.14}$$

These two boundary conditions for the MOL are specified as conditions for the first and last points of the grid. So after introducing a grid with uniformly spaced N points to the dimensionless form of PDE, as $0 = z_0 < z_1 < z_2 < ... < z_N = 1$, and using the central difference formula by Eq. (2.39) to approximate the spatial derivative, the system of ODEs for internal grid points is defined by:

$$\frac{dc_A(z_j)}{d\tau} = \frac{c_A(z_{j-1}) - 2c_A(z_j) + c_A(z_{j+1})}{2\Delta z} \tag{4.15}$$

The boundary condition for the first grid point Eq.(4.11) can be evaluated using the second-order forward finite difference approximation formula by Eq. (2.36):

$$\frac{dc_A(z_0)}{dz} = \frac{-3c_A(z_0) + 4c_A(z_1) - c_A(z_2)}{2\Delta z} = -\delta_z(\tau) \tag{4.16}$$

$$c_A(z_0) = \frac{4}{3}c_A(z_1) - \frac{1}{3}c_A(z_2) + 2\Delta z\delta_z(\tau) \tag{4.17}$$

The concentration at the last point of grid is defined by:

$$c_A(z_n = 1) = 0 \tag{4.18}$$

Exit flow rate of c_A can be defined by following equation:

$$F_A = -A_c D_k \frac{\partial C_A}{\partial x}\Big|_{x=L} \tag{4.19}$$

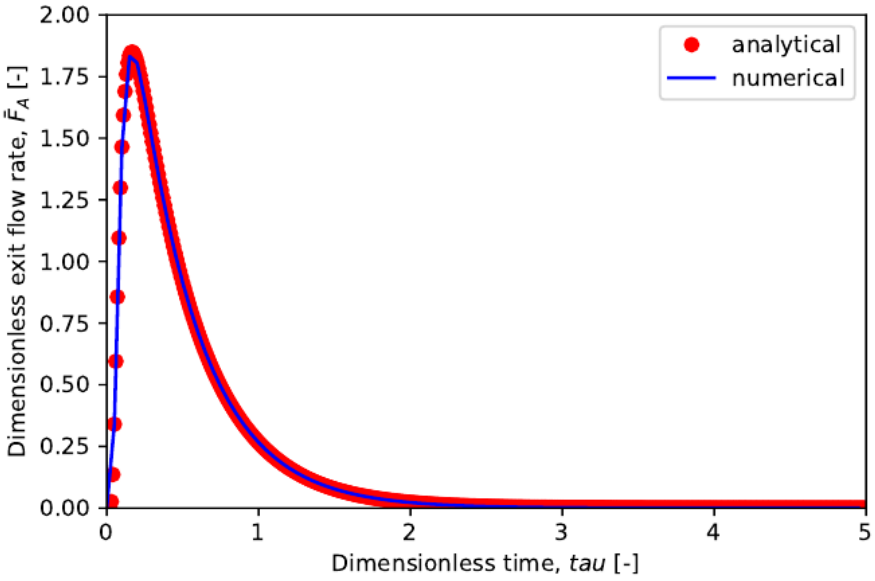
After introducing substitutions to dimensionless form, the dimensionless exit flow rate become:

$$\bar{F}_A = -\left(\frac{-4C_i\left(x_{N-1}\right) + C_i\left(x_{N-2}\right)}{2\Delta x} \right) \tag{4.20}$$

The numerical solution and the analytical solution algorithm are given in Appendix A. The IPython ODE solver uses build in Real-valued Variable-coefficient Ordinary Differential Equation solver, which applies backward differentiation formula method for stiff problems. In our simulation we set absolute tolerance and relative tolerance as 10^{-6} , which aligns the ODE solver to specified variable discretization[26].

Figure 3 illustrates the resulting transient response curves obtained by analytical and numerical solutions of PDEs for non-reacting species, when the rates of adsorption and desorption are equal to zero.

Figure 3 Transient response curve for non-reacting species



As we can observe, the dimensionless exit flow peaked at 1.85, so this value could be taken as a reference for estimation diffusivity, as no reaction occurs on catalyst surfaces. Gleaves *et al*.[20] used this methodology for estimation the Knudsen diffusivity Secondly, we may notice that the diffusion curves for both analytical and numerical solutions coincide with each other.

Case 2. Irreversible adsorption.

The mass balance of the first order irreversible adsorption on catalyst surface is given by Eq. (2.18). This PDE is also transformed into dimensionless form by introducing following substitutions as: $z = \frac{x}{L}$, $c_A = \frac{C_A}{N_{pA} / \varepsilon A_c L}$, $\tau = \frac{tD_k}{\varepsilon L^2}$, and $\bar{k}^+ = \frac{\rho_b L^2}{D_k} k^+$. Dimensionless form of Eq.(2.18) is defined as:

$$\frac{\partial c_A}{\partial \tau} = \frac{\partial^2 c_A}{\partial z^2} - \bar{k}^+ c_A \tag{4.21}$$

The MOL is applied to Eq.(4.21), with the boundary conditions by Eqs.(4.17) and (4.18). Thus, ODEs at internal grid points are given by:

$$\frac{dc_A(z_j)}{d\tau} = \frac{c_A(z_{j-1}) - 2c_A(z_j) + c_A(z_{j+1})}{2\Delta z} - \bar{k}^+ c_A(z_j) \tag{4.22}$$

The analytical solution for dimensionless concentration c_A is given by[13]:

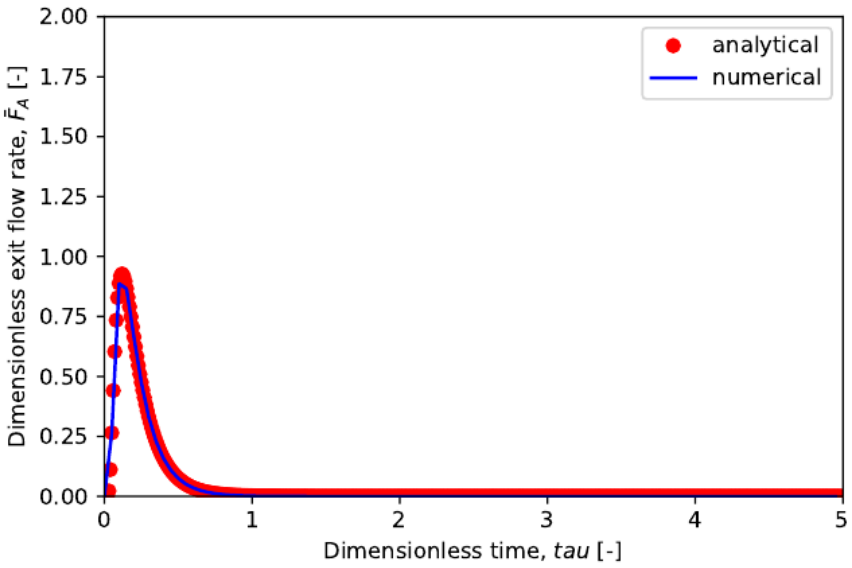
$$c_A(z,\tau) = 2\exp(-\bar{k}^+\tau)\sum_{n=0}^{\infty}\cos\left((n+\frac{1}{2})\pi z\right)\exp\left(-(n+\frac{1}{2})^2\pi^2\tau\right) \tag{4.23}$$

The dimensionless flow rate is expressed as:

$$\bar{F} = \pi \exp(-\bar{k}^{+}\tau)\sum_{n=0}^{\infty}(-1)^{n}(2n+1)\exp(-(n+0.5)^{2}\pi^{2}\tau) \tag{4.24}$$

Dimensionless adsorption constant is taken as $\bar{k}^{+} = 5$.

Figure 4 Transient response curve for irreversible adsorption, $\bar{k}^{+} = 5$



The computer codes utilizing the numerical solution algorithm and the analytical solution algorithm are given in Appendix B. Figure 4 demonstrates analytical and numerical solutions of PDE for irreversible adsorption with constant adsorption kinetics k_a=5. In the second case, the transient response curve peaked at 0.82, which is significantly lower than the peak of the transient curve for gas species, which did not react on the surface (Fig. 3). Thus, the part of pulsed molecules is adsorbed on the catalyst surface, while other part is detected at the exit of the reactor. Likewise,

previous results the curves for both analytical and numerical solution of PDE coincide with each other.

Case 3 Reversible adsorption.

In the case of reversible adsorption taking place on the catalyst surface, the mass balance consists of two PDEs, first is for gas species and second for surface intermediates. The TAP reactor model for reversible adsorption is given by Eq.(2.22) - (2.23). The same substitutions are applied to transfer equations in dimensionless form: $z = \frac{x}{L}$, $c_A = \frac{C_A}{N_{pA} / \varepsilon AL}$, $\tau = \frac{tD_k}{\varepsilon L^2}$, $\bar{k}_1^+ = \frac{\rho_b L^2}{D_k} k_1^+$. The dimensionless desorption rate constant and fractional coverage are defined as: $\bar{k}_1^- = k_1^- \frac{\varepsilon L^2}{D_k}$, and $\bar{\theta}_A = \theta_A \frac{\rho_b A \varepsilon}{N_{pA}}$

Finally, the dimensionless form of PDEs for reversible adsorption are given by:

$$\frac{\partial c_A}{\partial \tau} = \frac{\partial^2 c_A}{\partial z^2} - \bar{k}_1^+ c_A + \bar{k}_1^- \bar{\theta}_A \tag{4.25}$$

$$\frac{\partial \bar{\theta}_A}{\partial \tau} = \bar{k}_1^+ c_A - \bar{k}_1^- \bar{\theta}_A \tag{4.26}$$

After application of MOL as in previous cases, the set of ODEs in internal points is given by:

$$\frac{\partial c_A(z_j)}{\partial \tau} = \frac{c_A(z_{j-1}) - 2c_A(z_j) + c_A(z_{j+1})}{2\Delta z} - \bar{k}_1^+ c_A(z_j) + k_1^- \bar{\theta}_A(z_j) \tag{4.27}$$

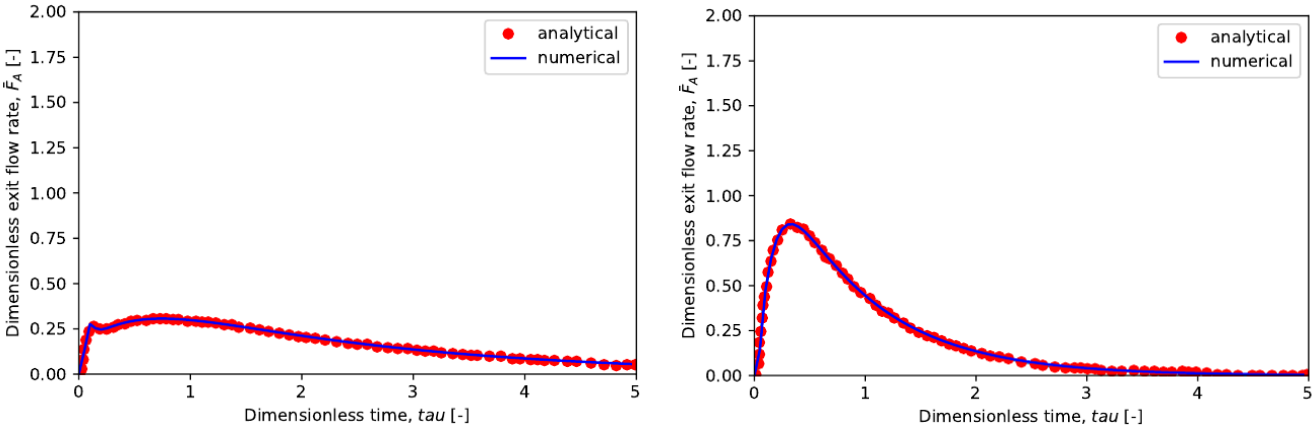
$$\frac{d\bar{\theta}_A(z_j)}{d\tau} = \bar{k}_1^+ c_A(z_j) - \bar{k}_1^- \bar{\theta}_A(z_j) \tag{4.28}$$

The computer codes used for calculation of numerical and analytical solutions are given in Appendix C.

The complete analytical solution for the dimensionless exit flow rate and concentration can be found in the literature [13]. Figure 5 (a) shows the transient pulse response when $\bar{k}_1^+ = 20$ and $\bar{k}_1^- = 5$. We can notice that the response curve has a sharp rise then gradually decreases, which indicates slow desorption of reactant. At the same time, when both adsorption and desorption kinetic constants are equal to 20 (Fig.5b), the response curve peaked at 0.8 and then dropped rapidly as the result of high desorption rate. Again, both numerical and analytical solutions have coincided, as shown in Fig. 5.

In the first case when non-reacting gas species are pulsed into TAP reactor the resulting curve can be used as a 'reference curve' to estimate transport parameters in the reactor. The second case and third case are used to evaluate the adsorption and desorption rate. These cases are considered to compare the numerical and analytical solutions to verify the accuracy of the method. An excellent agreement between analytical and numerical results validates the model and numerical algorithm as well as confirms the computer code. Therefore, we will apply algorithm and computer code for modeling of transient responses of complex multipath reactions.

Figure 5. Transient response curves for reversible adsorption, a) $\bar{k}_1^+ = 20$ ***and*** $\bar{k}_1^- = 5$***, b)*** $\bar{k}_1^+ = 20$ ***and*** $\bar{k}_1^- = 20$***,***



A B

4.2. Multipath reaction analysis

In previous cases, reaction model equations were presented in dimensionless form for simple non-reaction diffusion and adsorption-desorption of the gas species. However, for complicated reaction mechanism, we are going to use the non-normalized form of TAP reactor equations.

Case 4. Langmuir-Hinshelwood reaction path

In this case, we will consider the reaction that proceeds between adsorbed species according to Langmuir-Hinshelwood reaction mechanism. The complete Langmuir-Hinshelwood reaction route is given in Eq. (3.2) and the TAP reactor model are described in Equations (3.6-3.10). For simplification of the TAP reactor model, we take the number of molecules for each reactant species at the inlet reactor as 10^4, 10^5 and 10^6, respectively. Results of the simulation for different temperatures are given in Fig. 7, when the pulse intensity is equal to 10^4 molecules. As we can observe, the flow rates of CO_2 at the reactor outlet drops with temperature. Moreover, the carbon dioxide production rate decreases as temperature increased (Figure 6). At the same time, if we look at the flow rate of O_2 (Figure 7), it is evident that O_2 flow rate increases with temperature. This pattern is observed because the adsorption rate of O_2 on catalyst surface is slow. At higher temperatures, the adsorption rate of oxygen decreases resulting in higher peak on transient response of O_2. Therefore, the

adsorption rate of O_2 is not enough to achieve full consumption of O_2. The exit flow rate of CO_2 decreases because of low amount of adsorbed oxygen species available for reaction. Furthermore, the outlet flow rate of CO at all temperatures is zero, because O_2 is an excess species, and CO is fully reacted to form CO_2.

Similar simulations were carried for higher pulse intensities with inlet pulses containing 10^5 and 10^6 molecules, and Figs. 8 and 9 illustrate results of simulations. For 10^5 and 10^6 pulse intensities we can observe that CO_2 exit flow rate is slightly higher for 130 ^oC and 140 ^oC, and there is no significant change at higher temperatures. Also as in the previous case, the performance of catalyst for CO_2 formation decreases at the higher temperatures and a more pronounced differences are observed for O_2 curves. If we compare transient response curves for similar temperatures but different pulse intensities, we can notice that O_2 exit flow rate increases with pulse intensities, as more O_2 is supplied to the reactor. The maximum flow rate of oxygen reaches 12.

Based on simulation results discussed above we can make following conclusion:

A) In case of Langmuir-Hinshelwood reaction mechanism, the rate of CO_2 formation decreases when the reaction temperature increases.

B) The exit flow rate of O_2 increases considerably with temperature and pulse intensity.

C) The adsorption rate of O_2 is slower than the consumption rate of O_2 in reaction to form carbon dioxide.

D) Carbon monoxide supplied to the reactor is consumed completely in the reaction, resulting increasing exit flow rate of O_2 and decreasing flow rate of CO.

Figure 6 Transient response of CO_2 for Langmuir-Hinshelwood reaction path

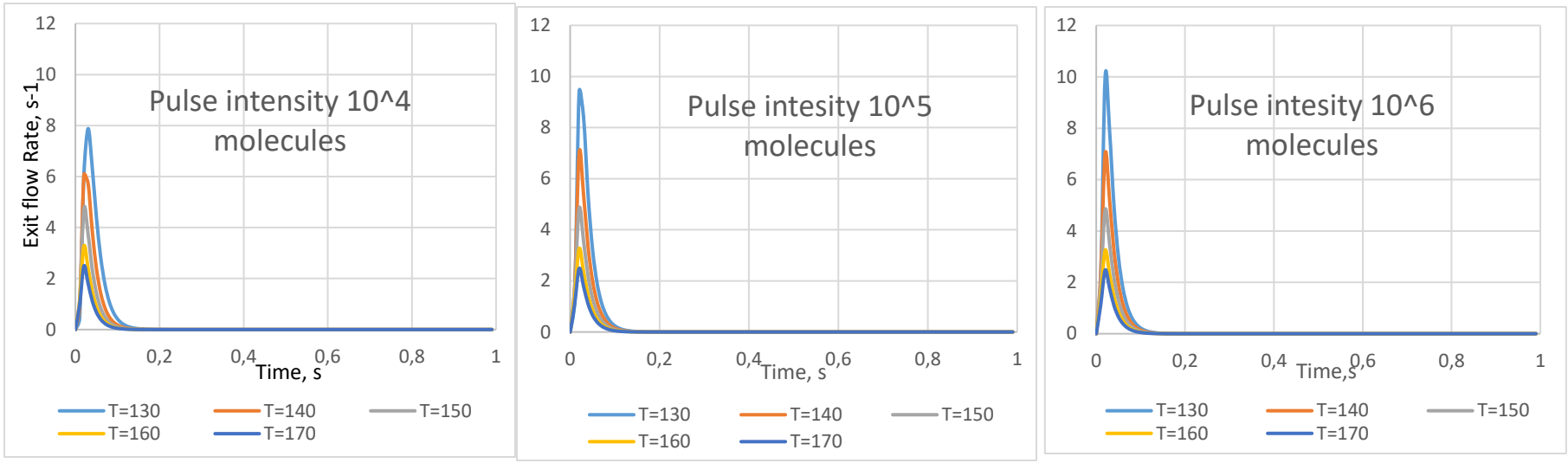


Figure 7 Transient responses of LH for different temperatures and pulse containing 10^4 molecules

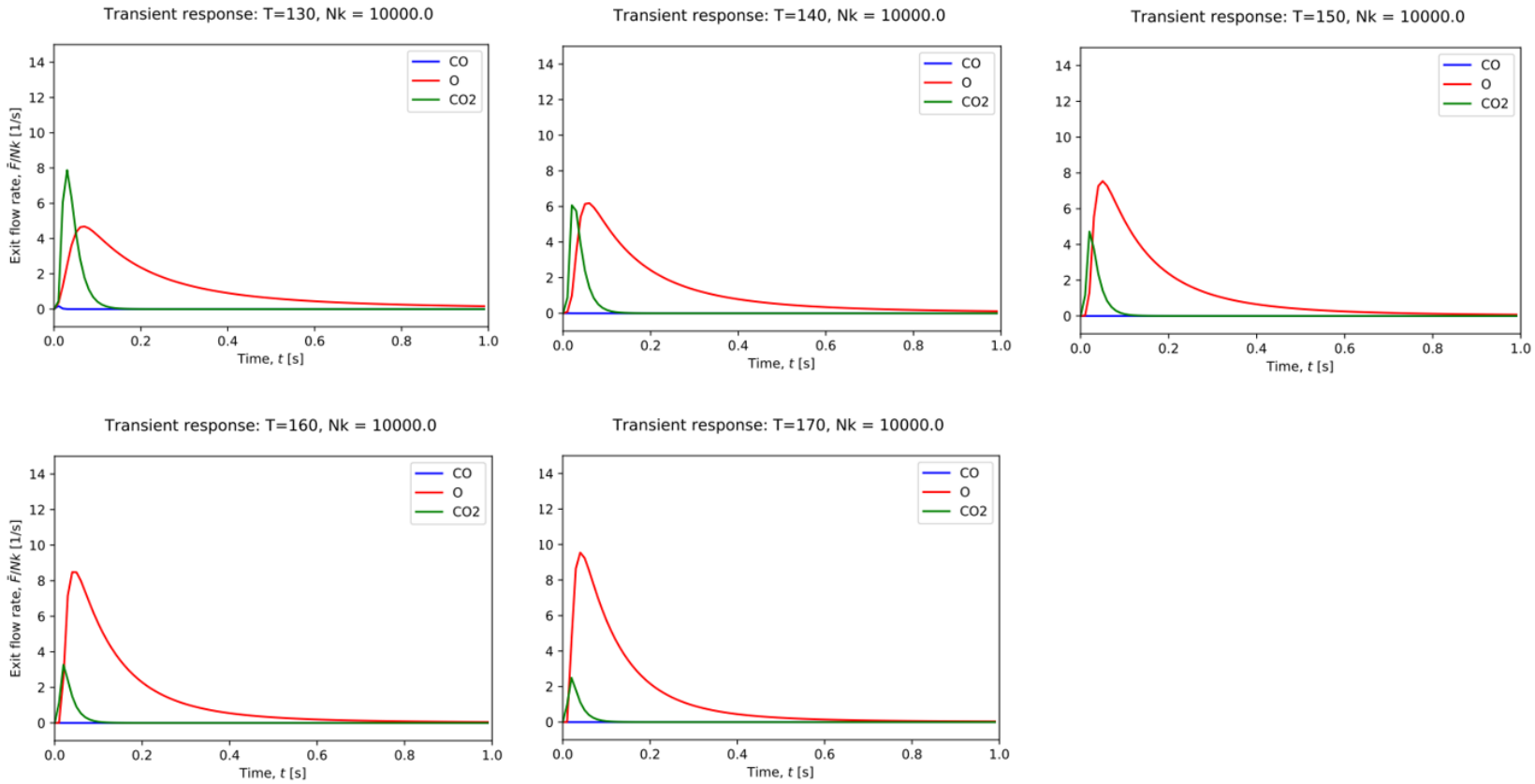


Figure 8 Transient responses of LH for different temperatures and pulse containing 10^5 molecules

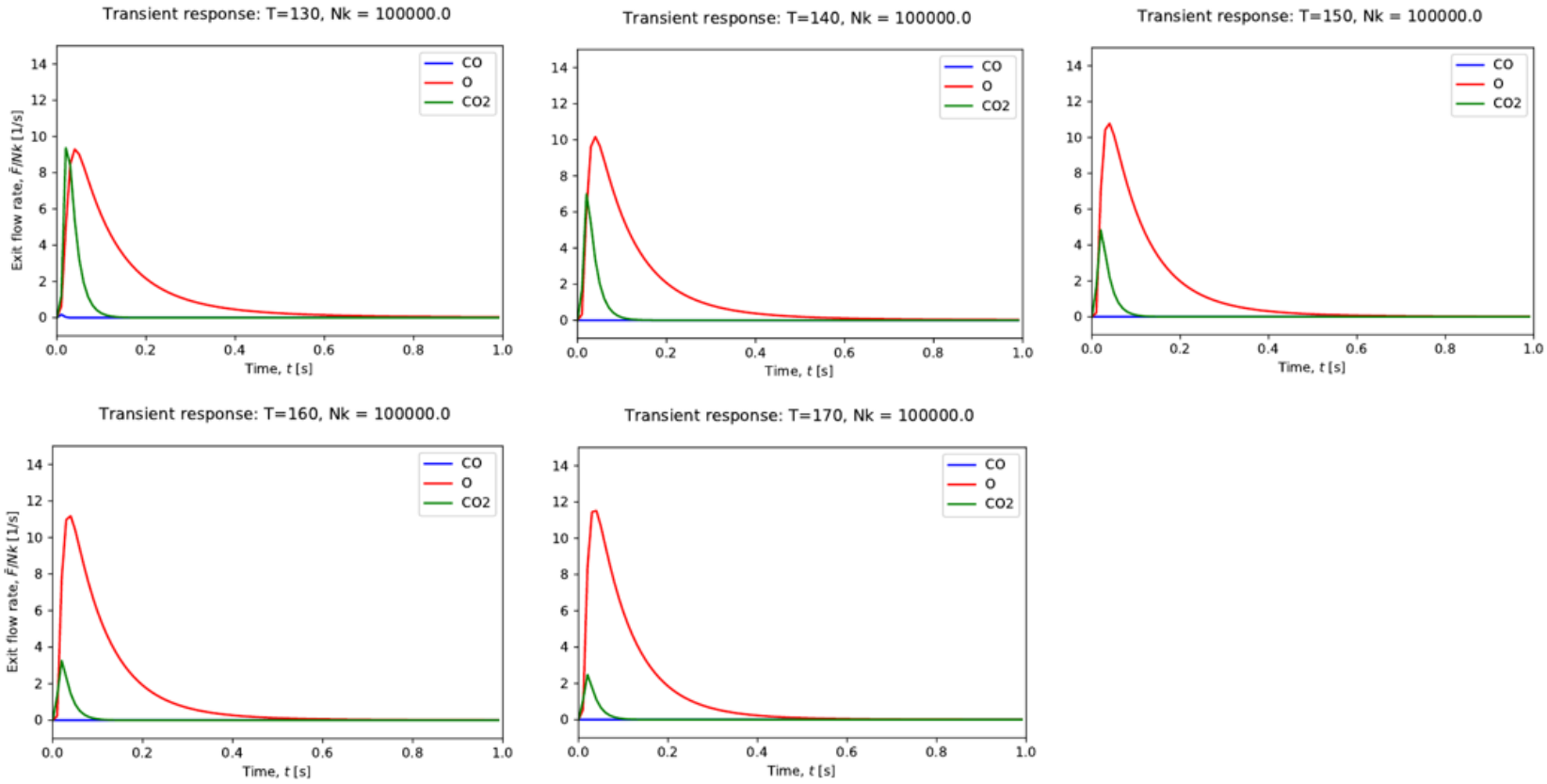
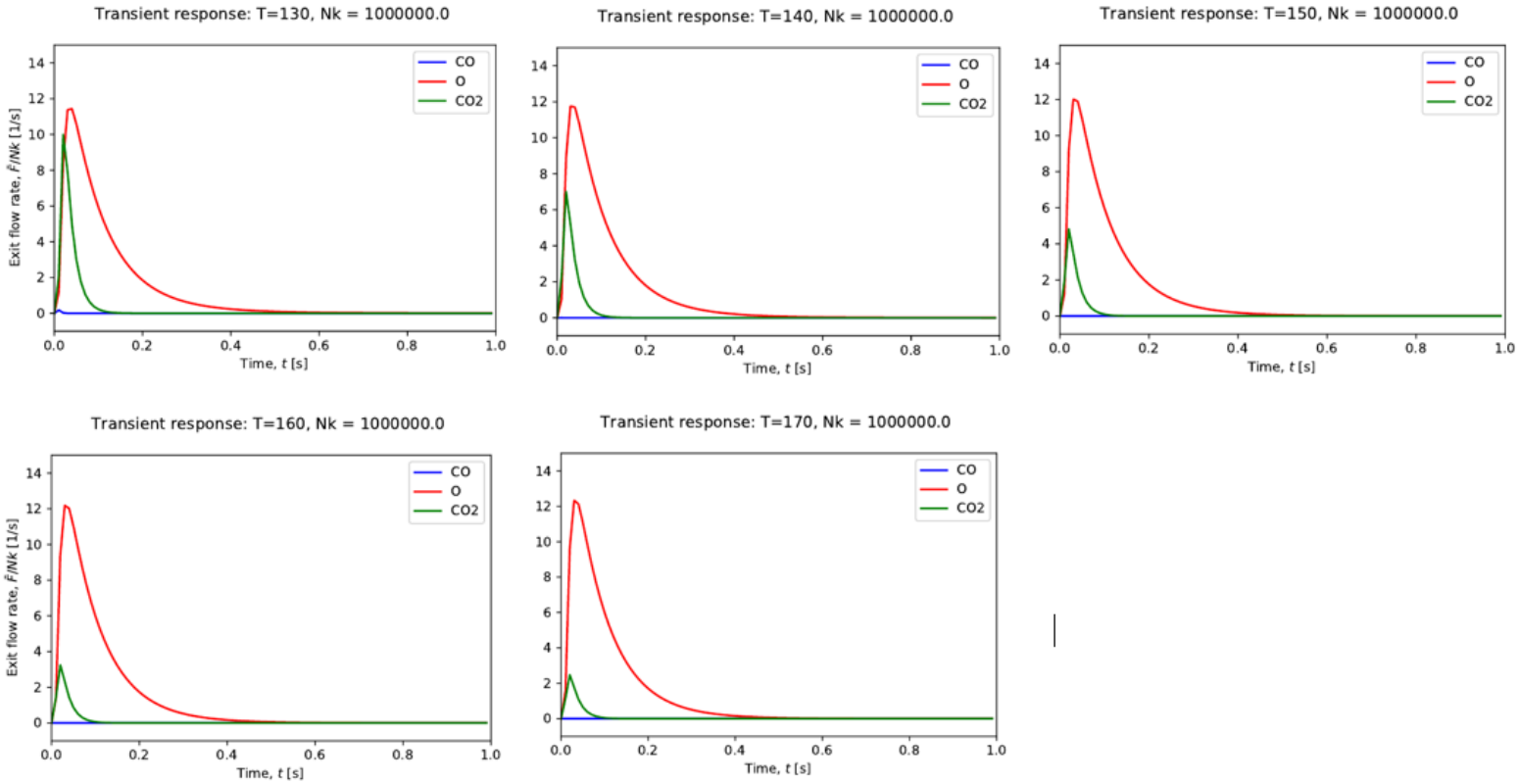


Figure 9 Transient responses of LH for different temperatures and pulse containing 10^6 molecules



Case 5. Eley-Rideal reaction path

In this case, the discussion will be focused on the reaction that proceeds under Eley-Rideal reaction route. The model reaction is given in equations (3.12) and (3.13), and TAP reactor model is given in equations (3.14-3.17). Figure 11 shows results of simulation for 10^4 pulse intensity. We can notice that at lower temperatures the flow rate of CO_2 is considerably higher even though reaction rate of CO_2 is lower than in case of Langmuir-Hinshelwood reaction route. Moreover, we can observe the emergence of CO at the outlet of the reactor, and a closer look at the chart indicates that CO is depleted faster than O2. This pattern can be explained by the fact that in ER mechanism O_2 is adsorbed on catalyst surface through dissociative adsorption mechanism, while carbon monoxide is reacted with oxygen anion on the catalyst surface from gas phase, this can be seen in reaction model in Eq.(3.11). Again we notice that as temperature increases the flow rate for O_2 increases, but not as pronounced as in case of Langmuir-Hinshelwood. On the contrary, the flow rates of CO decreases with higher temperature, because CO is not adsorbed, but reacted from gas phase. The emergence of carbon monoxide in some simulations indicates that carbon monoxide is excess reactant.

The simulations were carried out for higher pulse intensities as in case of Langmuir-Hinshelwood mechanism. The flow rate of O_2 remains unchanged for all cases, but CO and CO_2 exit flow rates significantly fluctuate with pulse intensities.

For 10^4, 10^5 , 10^6 pulse intensities we can observe the similar drop of the CO_2 flow rate at higher temperatures, as shown in Fig. 10. Besides, it should be emphasized that exit flow rate of carbon dioxide in the Eley-Rideal route is significantly higher than in Langmuir-Hinshelwood reaction route.

According to the simulation results and discussion above for Eley-Rideal reaction mechanism, we have reached following results:

A) The rate of carbon dioxide formation decreases when temperature increases similarly as in case of Langmuir-Hinshelwood reaction mechanism.

B) The exit flow rate of carbon monoxide decreases as temperature increases, indicating that carbon monoxide is limiting reactant in lower temperature.

C) The exit flow rate of oxygen remains relatively stable when temperature increases, because consumption rate of carbon monoxide increases with temperature.

D) For higher pulse intensities, carbon dioxide decreasing trend is pronounced considerably.

Figure 10 Transient responses of CO_2 for Eley-Rideal reaction route

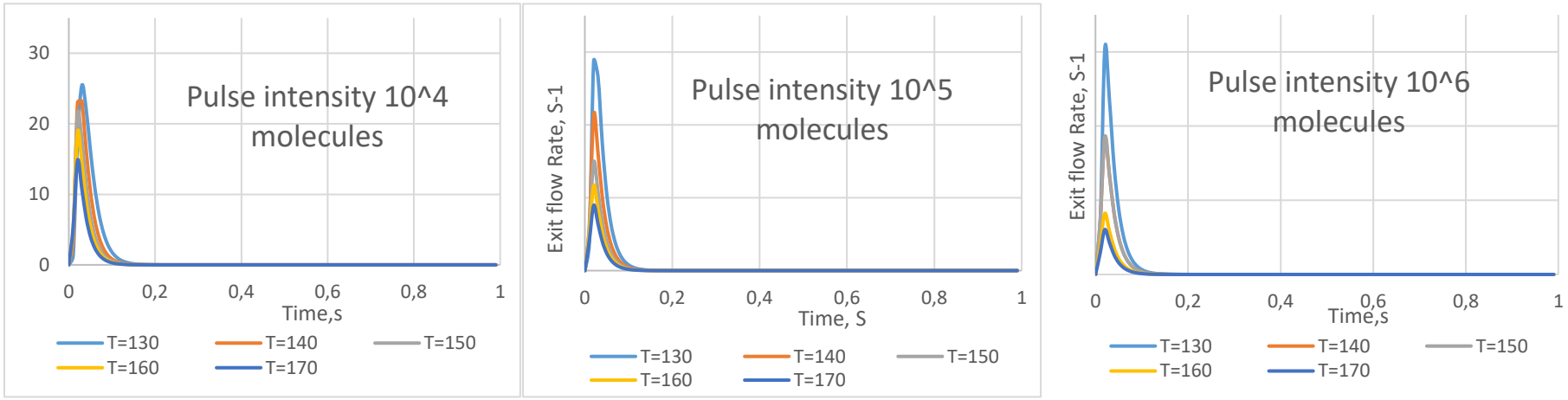


Figure 11 Transient responses of ER for different temperatures and pulse containing 10^4 molecules

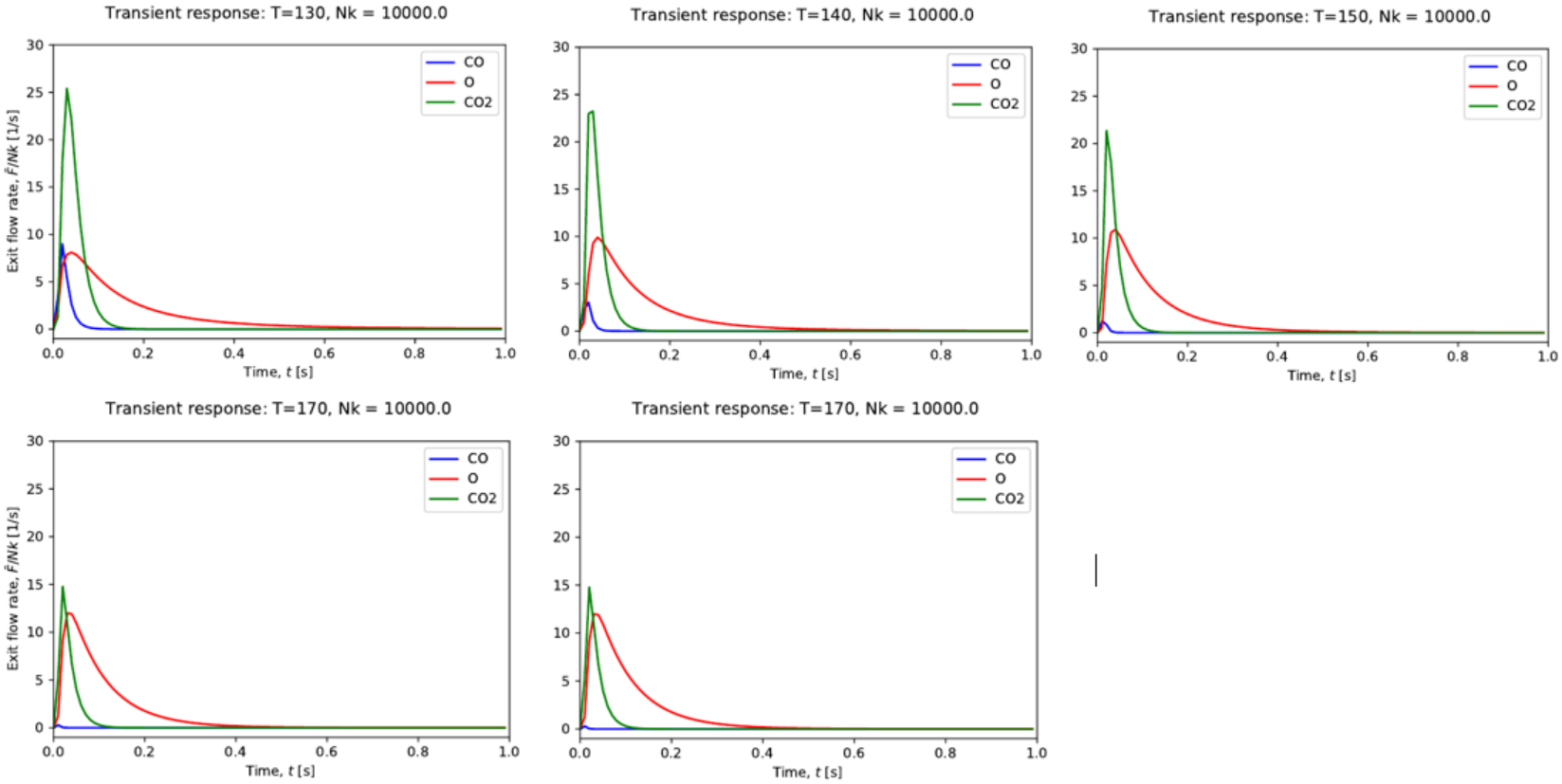


Figure 12 Transient responses of ER for different temperatures and pulse containing 10^5 molecules

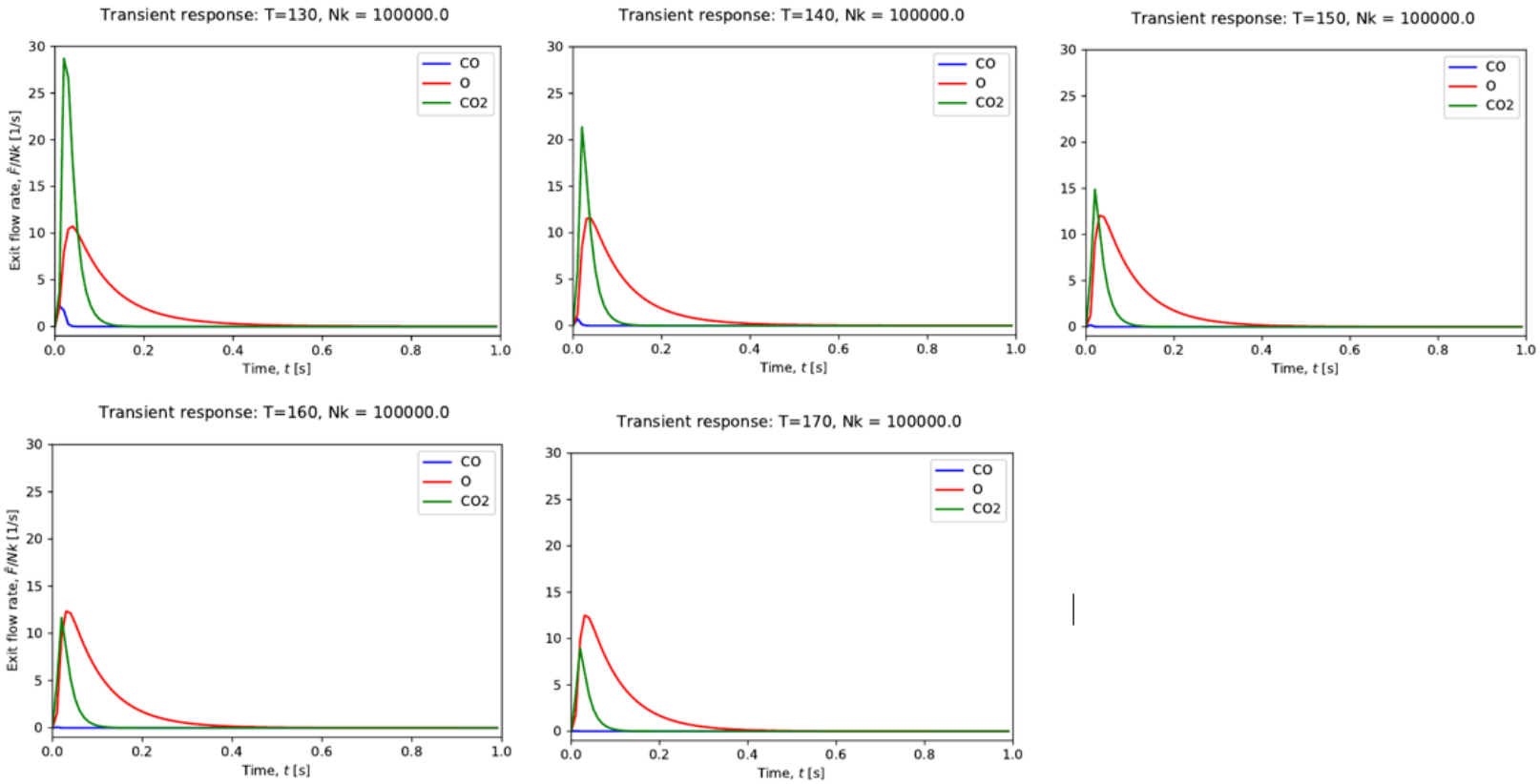
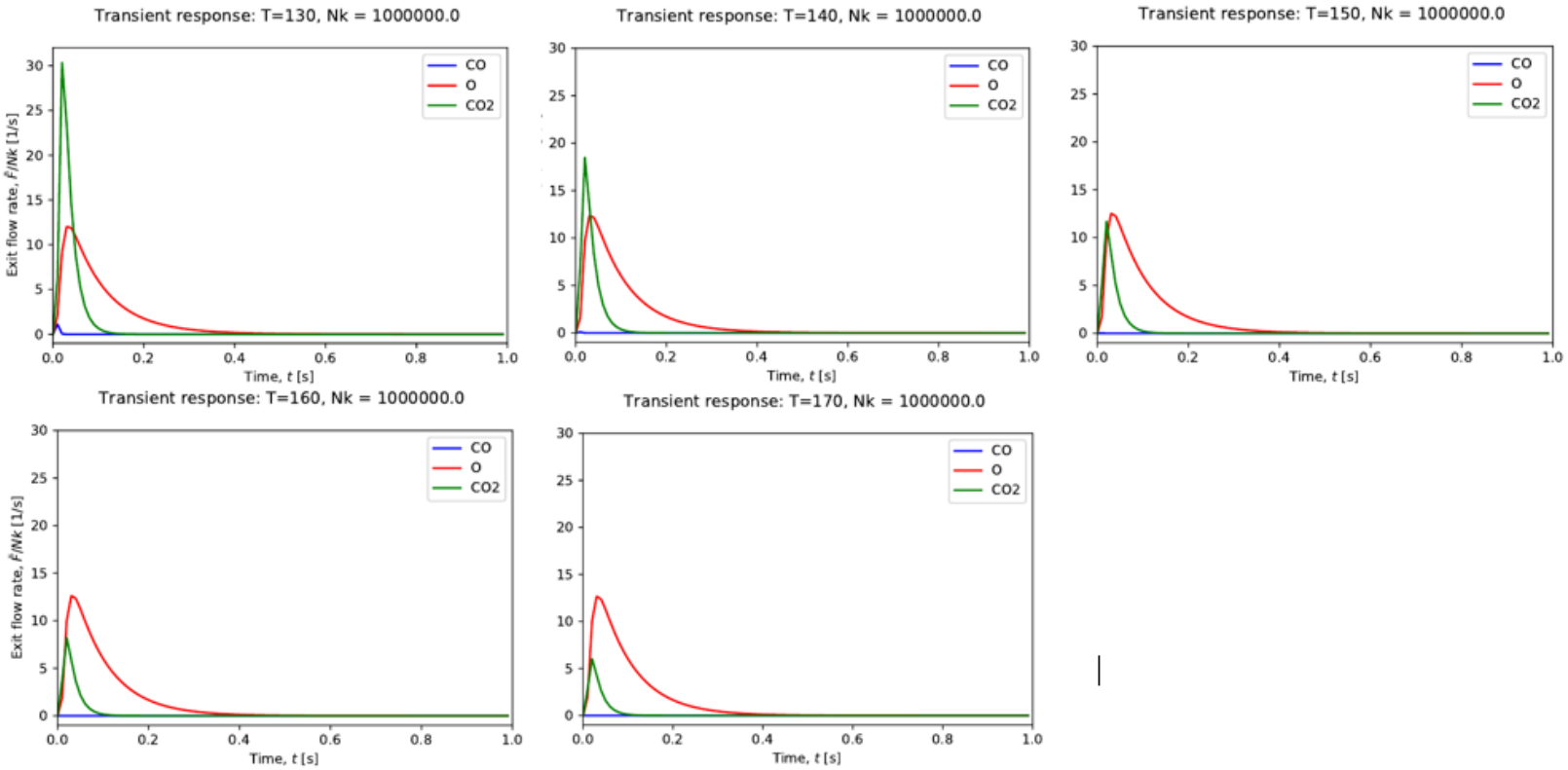


Figure 13 Transient responses of ER for different temperatures and pulse containing 10^6 molecules



Case 6. Combination of Langmuir-Hinshelwood and Eley-Rideal reaction mechanism

In previous cases, carbon monoxide oxidation mechanism was analyzed when two different reaction mechanisms were progressing separately from each other. As we can notice the exit curves of previous two cases are completely different from each other as well as reaction mechanisms. In the current case, Langmuir-Hinshelwood and Eley-Rideal reaction mechanisms are proceeding simultaneously on two different active sites (S_1 and S_2). The model reaction is described in the previous chapter by Eqs.(3.3), (3.4), (3.5) and (3.12), (3.13) . Figure 15 shows transient responses when the inlet pulse intensity consists of 10^4 molecules. As we can observe, the maximum exit flow rate of carbon dioxide is peaked at 7, which is lower than in similar separate reaction mechanisms. As in the previous reaction mechanisms, the production of CO_2 decreases with increasing temperature, as shown in Fig. 14. It should be noted that the transient curve of oxygen is smoother indicating that the adsorption /desorption rates are different from those in previous reaction mechanisms. However, this pattern is observed only at low pulse intensities.

Results of the simulation of multipath Langmuir-Hinshelwood and Eley-Redial mechanism for higher pulse intensities indicate similar behavior as in separate reaction mechanisms. These behaviors include the drop of exit flow rate of CO_2 as temperature rises, and an increase of exit flow rate of oxygen with temperature. The

maximum value of exit flow rate for O_2 reaches 12, which is identical to the maximum values in previous cases.

Figure 14 Transient responses of CO_2 for Langmuir-Hinshelwood and Eley-Rideal reaction route

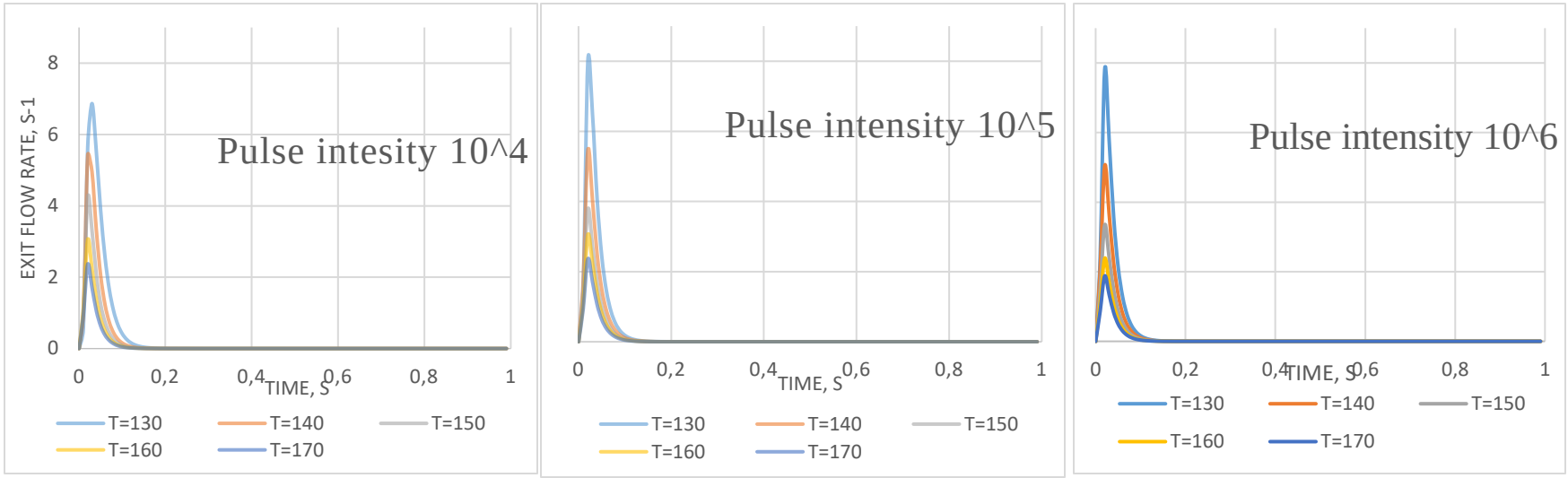


Figure 15 Transient responses of combination LH-ER for different temperatures and pulse containing 10^4 molecules

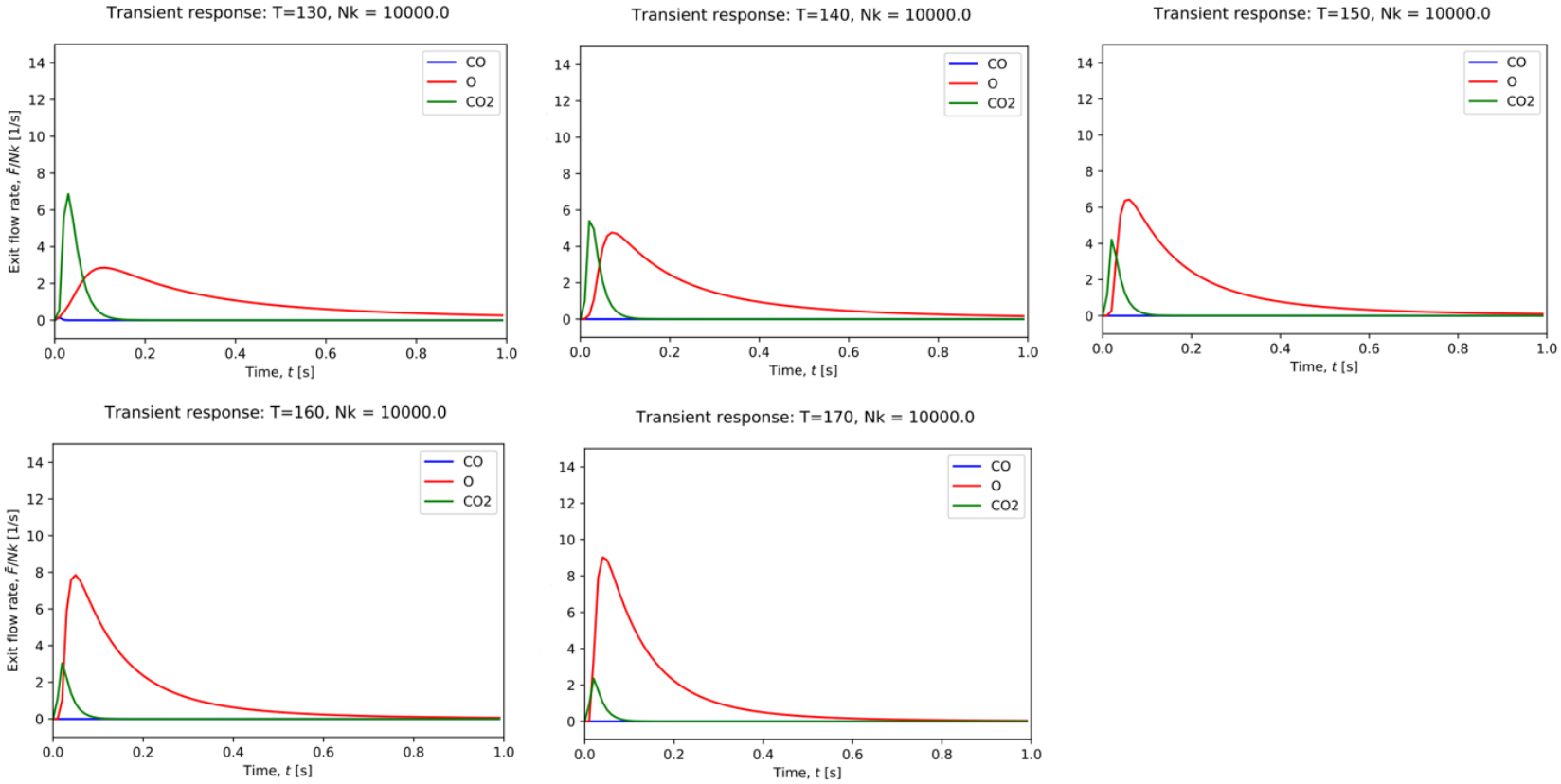


Figure 16 Transient responses of combination LH-ER for different temperatures and pulse containing 10^5 molecules

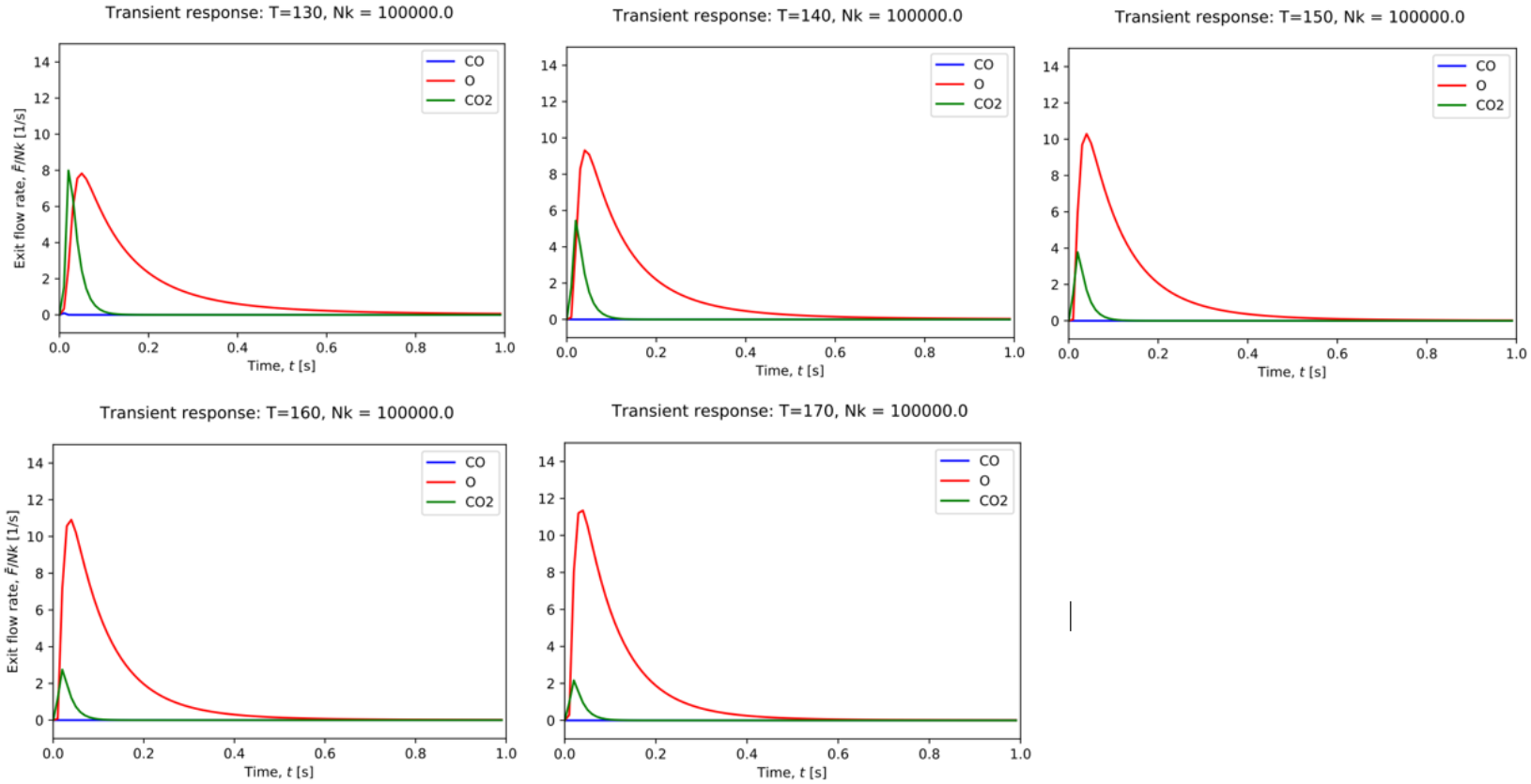
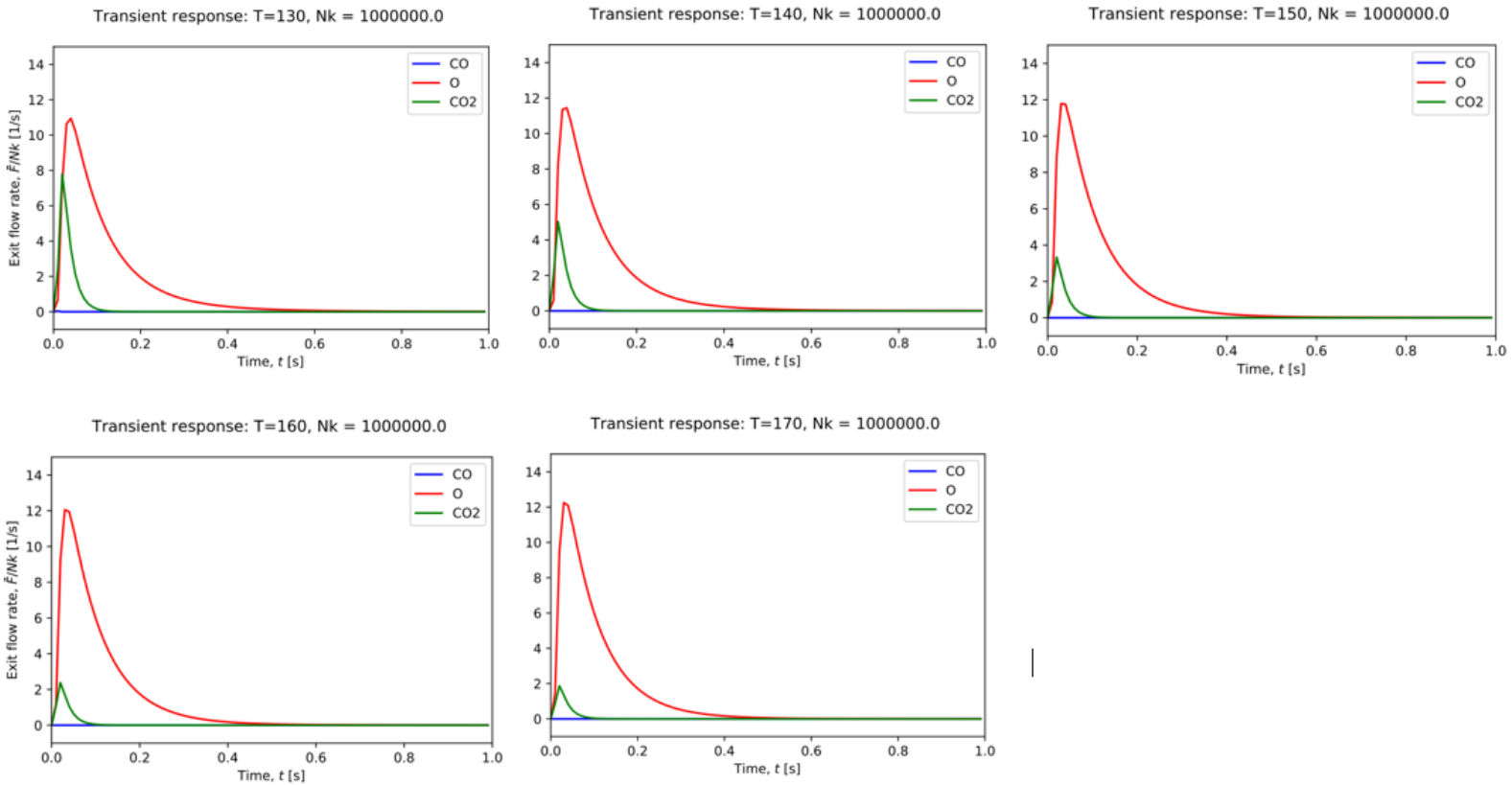


Figure 17 Transient responses of combination LH-ER for different temperatures and pulse containing 10^6 molecules



The conclusion for simulation on combined Langmuir-Hinshelwood and Eley-Rideal are:

A) Formation of carbon dioxide goes down as temperature increases for all pulse intensities

B) The exit flow rate of oxygen increases when temperature increases, however in 10^6 molecules pulse the oxygen flow rate shows no change for temperature grow.

C) As in case of Langmuir-Hinshelwood CO is completely consumed in reaction.

In the next analysis, we compare our simulation results of carbon monoxide oxidation with published experimental results. In our analysis, we use data which obtained from 10^5 pulse intensity simulations, because they show typical transient response curves. If we consider the affinity of the curves for different cases, we notice that the shape of the transient pulses for carbon dioxide in combined routes is similar to those for Langmuir-Hinshelwood reaction mechanism as shown in Figs.19 and 20. Indeed, according to these results, the overall reaction route is prevailed by Langmuir-Hinshelwood reaction mechanism. On the other hand, this conclusion coincides with one mentioned by Kobayashi *et al.* [1], using the results of steady-state analysis he confirmed that the formation of CO_2 undergoes through Langmuir-Hinshelwood reaction route in case of low concentration of CO. The results of [1]

work are demonstrated in Fig.18, where curves of LH-ER and LH coincide till the point B. Additionally, results for all cases showed that production of CO_2 decreases as temperature rises, which is consistent with results obtained by Mergler *et al.*[18], for CO oxidation on $Pt/CoO_x/SiO_2$ catalyst.

Figure 18. Illustration of the contribution of each reaction path to the steady-state rate.[1]

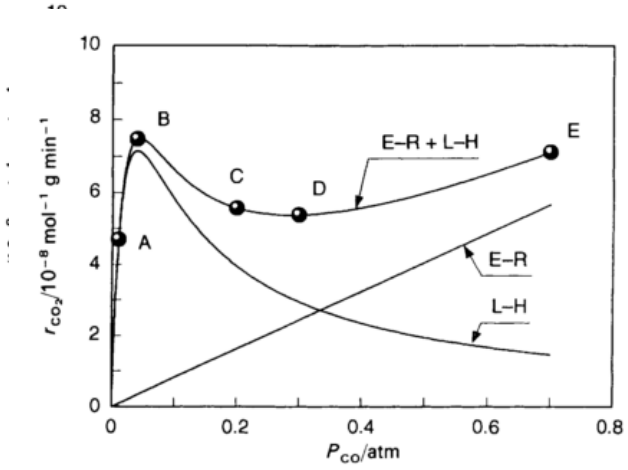
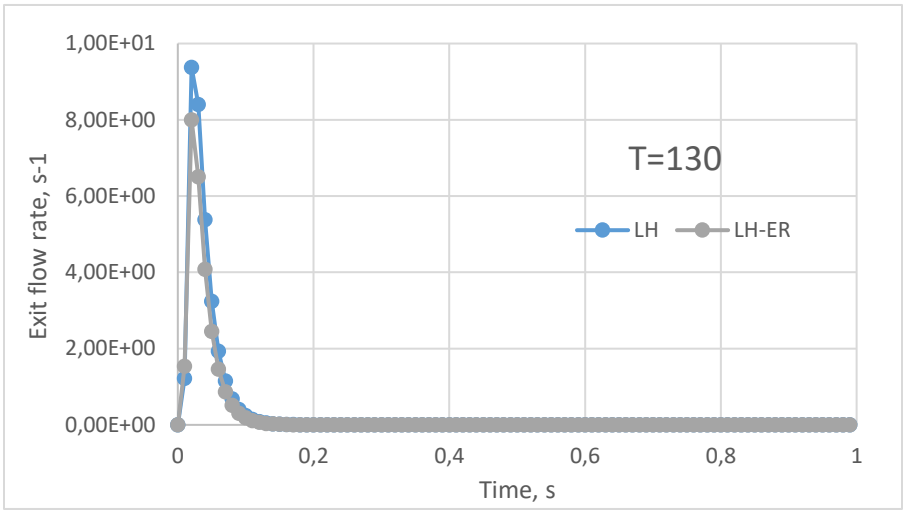
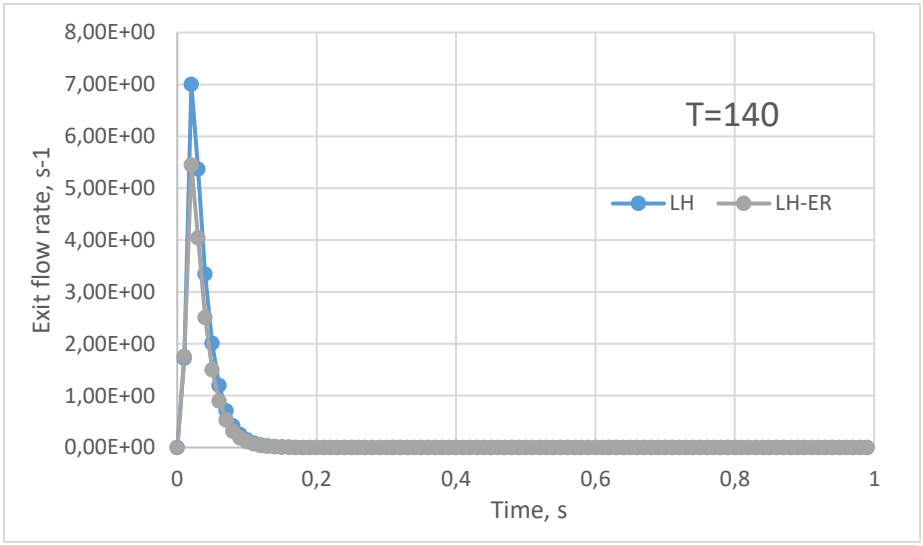
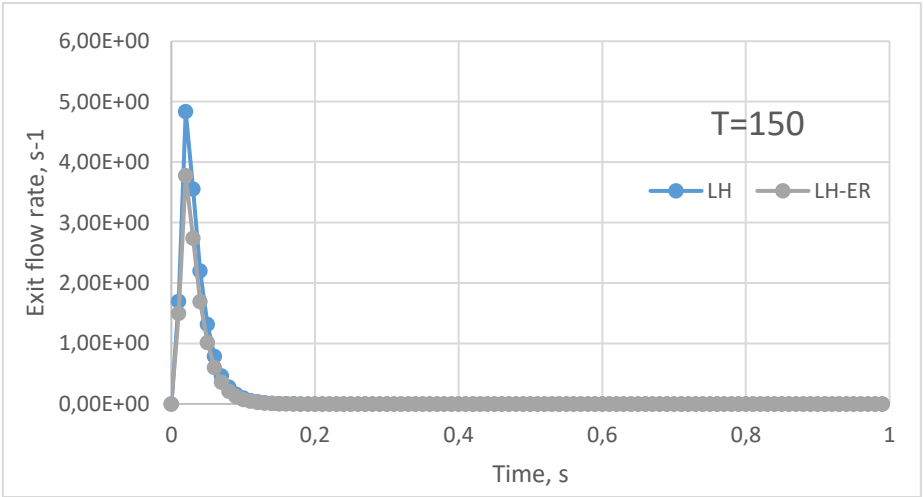
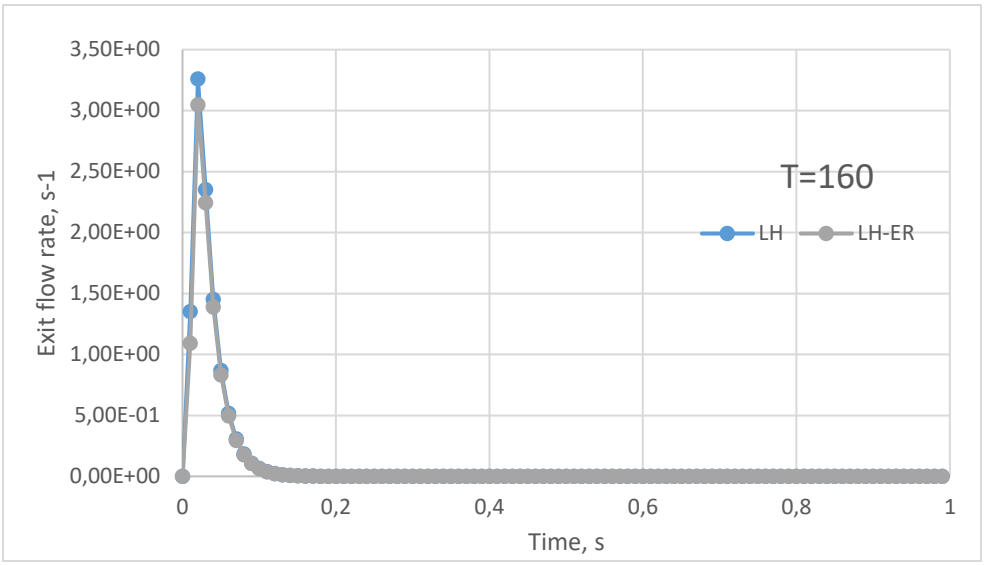


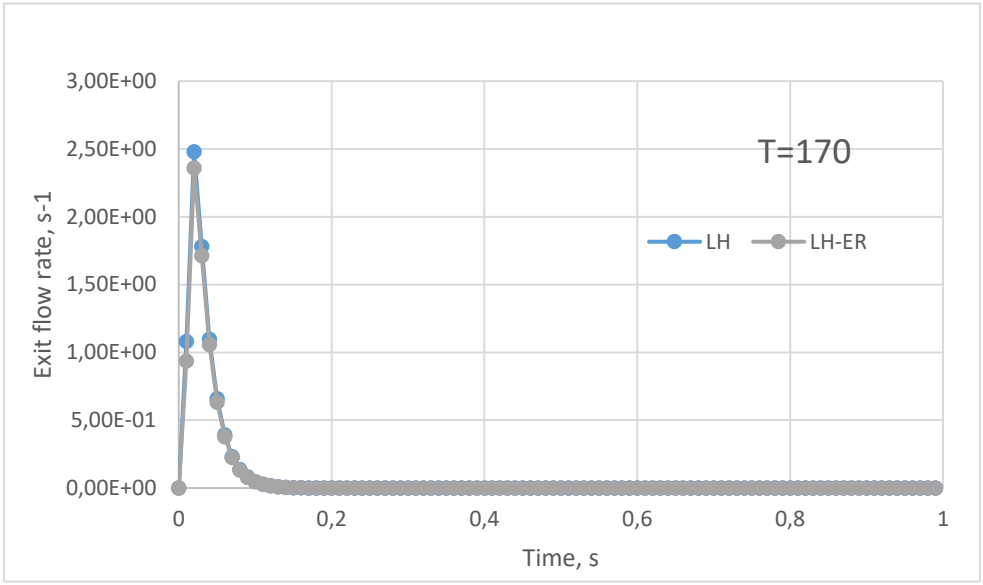
Figure 19. Transient response curves of CO_2 for 10^5pulse intensity








3,50E+00
3,00E+00
2,50E+00
2,00E+00
1,50E+00
1,00E+00
5,00E-01
0,00E+00
Exit flow rate, s-1
0
0,2
0,4
0,6
0,8
1
Time, s
T=160
LH
LH-ER


3,00E+00
2,50E+00
2,00E+00
1,50E+00
1,00E+00
5,00E-01
0,00E+00
Exit flow rate, s-1
0
0,2
0,4
0,6
0,8
1
Time, s
T=170
LH
LH-ER

Chapter 5 - Conclusion

The main objective of this thesis was to study the application of TAP technique for analysis of complex multi-path reactions. For this purpose, the mathematical model of TAP reactor was derived, the numerical algorithm and computer code were developed and verified on models having analytical solutions. The carbon monoxide oxidation on ZnO (K25-ZnO) catalyst is taken as an example of multi-path reaction. Results of computer simulation of Langmuir Hinshelwood mechanism confirm that the exit flow rate of O_2 increases as temperature rises, due to slow adsorption rate of O_2. For Eley-Rideal mechanism, we observed the emergence of CO at reactor outlet at low pulse intensities, the higher peak of CO_2 than in the case of Langmuir-Hinshelwood mechanism. The computer simulation of combined Langmuir-Hinshelwood and Eley-Ridel mechanism showed the similar transient response curve as in case of separate Langmuir-Hinshelwood reaction mechanism, due to low pulse intensities.

In the future work, at first, the model could be enhanced to simulate the multi-pulse experiments in order to evaluate the performance of the catalyst by inserting complimentary functions in the computer code. Secondly, the computer code could be further developed to evaluate the reaction kinetic parameters for different catalysts by fitting calculated transient curve to experimental results. Thirdly,

Method of lines applied in this thesis can also be used to solve heat equations. Finally, the computational model could be extended to simulate a three-zone reactor, where the reactive catalyst is installed between two inert particle layers.

References

[1] M. Kobayashi, B. Golman, and T. Kanno, “Diversity of the kinetics of carbon monoxide oxidation on differently prepared zinc oxides,” *J. Chem. Soc. Faraday Trans.*, vol. 91, no. 9, p. 1391, 1995.

[2] G. S. Yablonsky, M. Olea, and G. B. Marin, “Temporal analysis of products: Basic principles, applications, and theory,” *J. Catal.*, vol. 216, no. 1–2, pp. 120–134, 2003.

[3] A. M. N.Nadirov, “Catalysis and Petrochemistry,” *News Acad. Sci. Kazakhstan*, vol. 1, pp. 45–57, 2011.

[4] J. J. Heys, *Chemical and Biomedical Engineering Calculations Using Python*. 2017 John Wiley & Sons,Inc, 2017.

[5] T. Moustafa, “Short reference on Kinetic Modeling of Catalytic Reactions,” 2002.

[6] F. H. M. Dekker, A. Bliek, F. Kapteijn, and J. A. Moulijn, “Analysis of mass and heat transfer in transient experiments over heterogeneous catalysts,” *Chem. Eng. Sci.*, vol. 50, no. 22, pp. 3573–3580, 1995.

[7] T. Engel, G. Ertl, H. P. D.D. Eley, and B. W. Paul, “Elementary Steps in the Catalytic Oxidation of Carbon Monoxide on Platinum Metals,” *Adv. Catal.*,

vol. Volume 28, pp. 1–78, 1979.

[8] E. A. Redekop, G. S. Yablonsky, D. Constales, P. A. Ramachandran, J. T. Gleaves, and G. B. Marin, “Elucidating complex catalytic mechanisms based on transient pulse-response kinetic data,” *Chem. Eng. Sci.*, vol. 110, pp. 20–30, 2014.

[9] P. Azadi, G. Brownbridge, I. Kemp, S. Mosbach, J. S. Dennis, and M. Kraft, “Microkinetic modeling of the Fischer-Tropsch synthesis over cobalt catalysts,” *ChemCatChem*, vol. 7, no. 1, pp. 137–143, 2015.

[10] B. Golman, “Transient kinetic analysis of multipath reactions: An educational module using the IPython software package,” *Educ. Chem. Eng.*, vol. 15, pp. 1–18, 2016.

[11] J. T. Gleaves, J. R. Ebner, and T. C. Kuechler, “Temporal Analysis of Products (TAP) — A Unique Catalyst Evaluation System with Submillisecond Time Resolution,” *Catal. Rev.*, vol. 30, no. 1, pp. 49–116, 1988.

[12] J. Pérez-Ramírez and E. V. Kondratenko, “Evolution, achievements, and perspectives of the TAP technique,” *Catal. Today*, vol. 121, no. 3–4, pp. 160–169, 2007.

[13] J. T. Gleaves, G. S. Yablonskii, P. Phanawadee, and Y. Schuurman, “TAP-2:

An interrogative kinetics approach," *Appl. Catal. A Gen.*, vol. 160, no. 1, pp. 55–88, 1997.

[14] B. Zou, M. P. Duduković, and P. L. Mills, "Modeling of evacuated pulse micro-reactors," *Chem. Eng. Sci.*, vol. 48, no. 13, pp. 2345–2355, 1993.

[15] M. Olea, M. Kunitake, T. Shido, K. Asakura, and Y. Iwasawa, "Temporal Analysis of Products (TAP) Study of the Adsorption of CO, O2, and CO2 on a Au/Ti(OH)4* Catalyst.," *Bull. Chem. Soc. Jpn.*, vol. 74, pp. 255–265, 2001.

[16] P. Tantake, P. Phanawadee, Y. Boonnumpha, and J. Limtrakul, "Comparison between regression analysis and moment analysis for transport and kinetic parameter estimation in TAP experiments under a non-ideal inlet condition," *Catal. Today*, vol. 121, no. 3–4, pp. 261–268, 2007.

[17] M. Kobayashi and H. Kobayashi, "Transient response method in heterogeneous catalysis," *Catal. Rev. Sci. Eng.*, vol. 36, no. 10, pp. 139–176, 1974.

[18] Y. J. Mergler, J. Hoebink, and B. E. Nieuwenhuys, "CO Oxidation over a Pt/CoOx/SiO2 Catalyst : A Study Using Temporal Analysis of Products," *J. Catal.*, vol. 313, pp. 305–313, 1997.

[19] H. S. Fogler, *Elements of chemical reaction engineering*, vol. 42. 2010.

[20] J. T. Gleaves, G. Yablonsky, X. Zheng, R. Fushimi, and P. L. Mills, "Temporal analysis of products (TAP)-Recent advances in technology for kinetic analysis of multi-component catalysts," *J. Mol. Catal. A Chem.*, vol. 315, no. 2, pp. 108–134, 2010.

[21] J. S. Rieck and A. T. Bell, "Influence of adsorption and mass transfer effects on temperature-programmed desorption from porous catalysts," *J. Catal.*, vol. 85, no. 1, pp. 143–153, 1984.

[22] S. C. van der Linde, T. a. Nijhuis, F. H. M. Dekker, F. Kapteijn, and J. a. Moulijn, "Mathematical treatment of transient kinetic data: Combination of parameter estimation with solving the related partial differential equations," *Appl. Catal. A Gen.*, vol. 151, no. 1, pp. 27–57, 1997.

[23] V. F. van Kan J., Segal A., "Numerical Methods in Scientific Computing,"vol 12. July, 2014.

[24] A. Kumar, X. Zheng, M. P. Harold, and V. Balakotaiah, "Microkinetic modeling of the NO + H2 system on Pt/Al 2O3 catalyst using temporal analysis of products," *J. Catal.*, vol. 279, no. 1, pp. 12–26, 2011.

[25] K. Morgan *et al.*, "TAP studies of CO oxidation over CuMnOX and Au/CuMnOX catalysts," *J. Catal.*, vol. 276, no. 1, pp. 38–48, 2010.

[26] "Reference Guide, Integration and ODEs." [Online]. Available:

https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.ode.html. [Accessed: 14-Jan-2014].

Appendices

Appendix A. Computer code for simulation of transient responses of non-reacting species

```
%matplotlib inline
import numpy as np
from scipy.integrate import ode
import matplotlib.pyplot as plt
from ipywidgets import *
from IPython.display import clear_output, display, HTML
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
def ff(tau, y, params):
# specify ODEs to solve
# allocate array of time derivatives
    dcdt = np.zeros(len(y))
# unpack parameters
    ka, kd, nz = params
# Allocate arrays of gaseous components and surface intermediates
    Ca= np.zeros(nz)
#  Axial distribution of gas-phase reacting components
    Ca[0:nz]  = y[0:nz]
    delta_z = 1./float(nz + 1)
# set the delta function on interval 2*Δt
    delta_t =0.0001
    if tau > 2*delta_t:
        Delta = 0.0
    else:
        Delta = (1./(2*delta_t)) * (1 + np.sin((np.pi/delta_t)*(tau-delta_t/2)))
# boundary condition at reactor inlet
    Ca_0  = -Ca[1]/3. + 4.*Ca[0]/3. +  Delta*2*delta_z/3.
    if Ca_0 < 0.0:
        Ca_0 = 0.0
# boundary condition at reactor outlet
    CaN1 = 0.0
# differential equation at first internal point
    dcdt[0] = (Ca[1] - 2.*Ca[0] + Ca_0)/(delta_z**2)
# differential equation at last internal point
    dcdt[nz-1] = (CaN1 - 2.*Ca[nz-1] + Ca[nz-2])/(delta_z**2)
# differential equations at 2,...,nz-1 internal points
    for i in range (1, nz-1):
        dcdt[i] = (Ca[i+1] - 2.*Ca[i] + Ca[i-1])/(delta_z**2)
    return dcdt
def summation(tt):
    SUM = 0.0
    n = 100000
    for i in range(n):
        SUM += ((-1)**i)*(2*i+1)*np.exp(-((i+0.5)**2)*(np.pi**2)*tt)
    return SUM*np.pi
# Main program
def fmain (ka, kd, cbox_plot, cbox_data):
# set header
    h4.visible=True
# specify the number of internal points
    N = 100
# calculate the length of spatial interval
    delta_z = 1./float(N + 1)
# total number of points = number of internal points +
#    two boundary points
    ntpt = N + 2
# specify uniform grid of internal points
    z = np.linspace(delta_z, 1.-delta_z, N)
```

```
# specify the grid of all axial points, i.e. internal points + boundary points
    zz = np.zeros(ntpt)
    zz[0] = 0.
    zz[1:N+1] = z[0:N]
    zz[N+1] = 1.
# combine parameters to pass to the function f, params = [...,...]
    params = [ka, kd, N]
# allocate memory and set initial conditions
    NN = 1*N
    y = np.zeros(NN)
    y[0] = 0.
# set the time range for integration
    tau_start = 0.00001
    tau_final = 5
# specify the time interval for data collection
    delta_tau = 0.05
# calculate the number of time steps required with an additional
# step for initial condition
    num_steps = int(np.floor((tau_final - tau_start)/delta_tau) + 1)
# Prepare plots
    fig, ax = plt.subplots()
# Specify the ODE integrator parameters:
# The vode integrator with backward differentiation formula is used to solve
# the system of stiff ODEs.
# The absolute tolerance is atol, the relative tolerance is rtol,
# the number of steps is nsteps and the first-step size is first_step
    rint = ode(ff).set_integrator('vode', method='bdf', nsteps=50000,
                                  atol = 1.0e-06, rtol = 1.0e-06, first_step=1.0e-6)
    rint.set_initial_value(y, tau_start).set_f_params(params)
# Create vectors to store components axial distributions at specified time steps
    tau = np.zeros((num_steps, 1))
    F = np.zeros(num_steps)
    C_at_tau = np.zeros((num_steps, ntpt))
# Save initial values
    tau[0] = tau_start
    C_at_tau [0,1:N+1] = y [0:N]
    F[0] = 0.0
# Integrate the system of ODEs across each delta_t timestep
    kk = 1
    while rint.successful() and kk < num_steps:
        rint.integrate(rint.t + delta_tau)
#   store the results of integration:  time
        tau[kk] = rint.t
#    concentration
        C_at_tau[kk,1:N+1] = rint.y[0:N]
        C_at_tau[kk,N+1]   = 0.
        C_at_tau[kk,0]     = -rint.y[1]/3. + 4.*rint.y[0]/3.
#    exit flow rate
        F [kk] = -(-4*rint.y[N-1]+rint.y[N-2])/(2.*delta_z)
#   plot the transient response of dimensionless exit flow rate
        clear_output(wait=True)
        ax.cla()
        ax.plot(tau[0:kk], F[0:kk], color="blue", linestyle="solid", linewidth=2)
        ax.set_ylabel(r'Dimensionless exit flow rate, $\bar F_A$  [-]')
        ax.set_xlabel('Dimensionless time, $tau$ [-]')
        ax.set_title('Diffusion + Adsorption: ka = %s, kd = %s' %(ka, kd))
        ax.set_xlim ([0.0, 5.0])
        ax.set_ylim ([0.0, 2.0])
        display(fig)
```

```
# end of time cycle
        kk += 1

    plt.close()
#
# prepare the file name
#
    file_name1 = 'Ka'+str(ka)+'Kd'+str(kd)
    file_name = file_name1.replace(".", "_")
# print and save plot if required
    if  cbox_plot:
        t = np.arange(0.0, 5.0, 0.01)
        Fa = np.zeros (len(t))
        Fa = summation(t)
# plot the dimensionless exit flow rate
        plt.plot (t, Fa, 'ro', label = 'analytical')
        file_namep = file_name+'.pdf'
        plt.plot (tau, F, 'b-', label = 'numerical')
        plt.xlabel('Dimensionless time, $tau$ [-]')
        plt.ylabel(r'Dimensionless exit flow rate, $\bar F_A$ [-]')
        plt.title ('Transient response, Diffusion + Adsorption: ka = %s, kd = %s \n' %(ka, kd))
        plt.legend(loc='upper right')
        plt.xlim ([0.0, 5.0])
        plt.ylim ([0.0, 2.0])
        plt.savefig(file_namep,dpi = 600)
        plt.show()
        print ('Plot is saved in the file < %s >.' %(file_namep))
# save simulation results as a csv file if required
    if  cbox_data:
        file_named = file_name+'.csv'
        with open(file_named,"w") as oname:
            oname.write(" ntpt ,  %5.0f \n" % ( ntpt))
            oname.write(" ka   ,  %7.3f, kd ,  %7.3f  \n" % ( ka, kd))
            oname.write("\n Exit Flow Rate \n")
            for it in range  (0, kk):
                oname.write(" %6.4f  , %10.5e \n" % (tau[it], F[it]))
        print ('Data are saved successfully in the file < %s >.' %(file_named))
    return
# Specify widgets
h1=widgets.HTML(value="<b> Only diffusion  in one-zone TAP reactor</b>")
h2=widgets.HTML("<br> ")
h3=widgets.HTML("<i> Specify adsorption/desorption parameters</i>")
display(h1)
display(h2)
display(h3)
#
form = widgets.VBox()
ka = widgets.BoundedFloatText(description="ka:",
                              value=5.0, min=0.0, max=30.0,
                              padding = 4)
kd = widgets.BoundedFloatText(description="kd:",
                              value=5.0, min=0.0, max=30.0,
                              padding = 4)
button = widgets.Button(description="Click to Run Program",color="red",background_color= "lightgray")
checkbox1 = widgets.Checkbox(description="Save plot ", value=False)
checkbox2 = widgets.Checkbox(description="Save data ", value=False)
form.children = [ka, kd, checkbox1, checkbox2, button]
display(form)
#
h4 = widgets.HTML("<br><i> Simulation started, be patient!</i><br>")
display(h4)
h4.visible=False
```

```
#
def on_button_clicked(b):
    fmain(ka.value, kd.value, checkbox1.value, checkbox2.value)
    h4.visible=False
#
button.on_click(on_button_clicked)
```

Appendix B. Computer code for simulation of transient responses of irreversible adsorption

```
%matplotlib inline
import numpy as np
from scipy.integrate import ode
import matplotlib.pyplot as plt
from ipywidgets import *
from IPython.display import clear_output, display, HTML
def ff(tau, y, params):
# specify ODEs to solve
# allocate array of time derivatives
    dcdt = np.zeros(len(y))
# unpack parameters
    ka, kd, nz = params
# Allocate arrays of gaseous components and surface intermediates
    Ca= np.zeros(nz)
    Sa = np.zeros(nz)
# Split y array into arrays of gaseous components and surface intermediates:
#  Axial distribution of gas-phase reacting components
    Ca[0:nz]  = y[0:nz]
#  Axial distribution of surface intermediates
    delta_z = 1./float(nz + 1)
# set the delta function on interval 2*Δt
    delta_t =0.0001
    if tau > 2*delta_t:
        Delta = 0.0
    else:
        Delta = (1./(2*delta_t)) * (1 + np.sin((np.pi/delta_t)*(tau-delta_t/2)))
# boundary condition at reactor inlet
    Ca_0  = -Ca[1]/3. + 4.*Ca[0]/3. +  Delta*2*delta_z/3.
    if Ca_0 < 0.0:
        Ca_0 = 0.0
# boundary condition at reactor outlet
    CaN1 = 0.0
# differential equation at first internal point
    dcdt[0] = (Ca[1] - 2.*Ca[0] + Ca_0)/(delta_z**2) - ka*Ca[0] + kd*Sa[0]
# differential equation at last internal point
    dcdt[nz-1] = (CaN1 - 2.*Ca[nz-1] + Ca[nz-2])/(delta_z**2) - ka*Ca[nz-1] + kd*Sa[nz-1]
# differential equations at 2,...,nz-1 internal points
    for i in range (1, nz-1):
        dcdt[i] = (Ca[i+1] - 2.*Ca[i] + Ca[i-1])/(delta_z**2) - ka*Ca[i] + kd*Sa[i]
    return dcdt
def summation(tt):
    SUM = 0.0
    n = 100000
    ka=5
    for i in range(n):
        SUM += ((-1)**i)*(2*i+1)*np.exp(-((i+0.5)**2)*(np.pi**2)*tt)
    return SUM*(np.pi)*np.exp(-ka*tt)
# Main program
#
def fmain (ka, kd, cbox_plot, cbox_data):
# set header
    h4.visible=True
# specify the number of internal points
    N = 100
# calculate the length of spatial interval
    delta_z = 1./float(N + 1)
# total number of points = number of internal points +
#    two boundary points
    ntpt = N + 2
# specify uniform grid of internal points
    z = np.linspace(delta_z, 1.-delta_z, N)
# specify the grid of all axial points, i.e. internal points + boundary points
```

```
# specify the grid of all axial points, i.e. internal points + boundary points
    zz = np.zeros(ntpt)
    zz[0] = 0.
    zz[1:N+1] = z[0:N]
    zz[N+1] = 1.
# combine parameters to pass to the function f, params = [...,...]
    params = [ka, kd, N]
# allocate memory and set initial conditions
    NN = 1*N
    y = np.zeros(NN)
    y[0] = 0.
# set the time range for integration
    tau_start = 0.00001
    tau_final = 5.0
# specify the time interval for data collection
    delta_tau = 0.05
# calculate the number of time steps required with an additional
# step for initial condition
    num_steps = int(np.floor((tau_final - tau_start)/delta_tau) + 1)
# Prepare plots
    fig, ax = plt.subplots()
# Specify the ODE integrator parameters:
# The vode integrator with backward differentiation formula is used to solve
# the system of stiff ODEs.
# The absolute tolerance is atol, the relative tolerance is rtol,
# the number of steps is nsteps and the first-step size is first_step
    rint = ode(ff).set_integrator('vode', method='bdf', nsteps=50000,
                                  atol = 1.0e-06, rtol = 1.0e-06, first_step=1.0e-6)
    rint.set_initial_value(y, tau_start).set_f_params(params)
# Create vectors to store components axial distributions at specified time steps
    tau = np.zeros((num_steps, 1))
    F = np.zeros(num_steps)
    C_at_tau = np.zeros((num_steps, ntpt))
# Save initial values
    tau[0] = tau_start
    C_at_tau [0,1:N+1] = y [0:N]
    F[0] = 0.0
# Integrate the system of ODEs across each delta_t timestep
    kk = 1
    while rint.successful() and kk < num_steps:
        rint.integrate(rint.t + delta_tau)
#   store the results of integration:  time
        tau[kk] = rint.t
#    concentration
        C_at_tau[kk,1:N+1] = rint.y[0:N]
        C_at_tau[kk,N+1]   = 0.
        C_at_tau[kk,0]     = -rint.y[1]/3. + 4.*rint.y[0]/3.
#    exit flow rate
        F [kk] = -(-4*rint.y[N-1]+rint.y[N-2])/(2.*delta_z)
#   plot the transient response of dimensionless exit flow rate
        clear_output(wait=True)
        ax.cla()
        ax.plot(tau[0:kk], F[0:kk], color="blue", linestyle="solid", linewidth=2)
        ax.set_ylabel(r'Dimensionless exit flow rate, $\bar F_A$  [-]')
        ax.set_xlabel('Dimensionless time, $tau$ [-]')
        ax.set_title('Diffusion + Adsorption: ka = %s, kd = %s' %(ka, kd))
        ax.set_xlim ([0.0, 5.0])
        ax.set_ylim ([0.0, 2.0])
        display(fig)
# end of time cycle
        kk += 1
```

```
    plt.close()
# prepare the file name
    file_name1 = 'Ka'+str(ka)+'Kd'+str(kd)
    file_name = file_name1.replace(".", "_")
# print and save plot if required
    if  cbox_plot:
#        Comparison of numerical solution with analytical one
        t = np.arange(0.0, 5.0, 0.01)
        Fa = np.zeros (len(t))
        Fa = summation(t)
# plot the dimensionless exit flow rate
        plt.plot (t, Fa, 'ro', label = 'analytical')
        file_namep = file_name+'.pdf'
        plt.plot (tau, F, 'b-', label = 'numerical')
        plt.xlabel('Dimensionless time, $tau$ [-]')
        plt.ylabel(r'Dimensionless exit flow rate, $\bar F_A$ [-]')
        plt.title ('Transient response, Diffusion + Adsorption: ka = %s, kd = %s \n' %(ka, kd))
        plt.legend(loc='upper right')
        plt.xlim ([0.0, 5.0])
        plt.ylim ([0.0, 2.0])
        plt.savefig(file_namep,dpi = 600)
        plt.show()
        print ('Plot is saved in the file < %s >.' %(file_namep))
# save simulation results as a csv file if required
    if  cbox_data:
        file_named = file_name+'.csv'
        with open(file_named,"w") as oname:
            oname.write(" ntpt ,  %5.0f \n" % ( ntpt))
            oname.write(" ka   ,  %7.3f, kd ,  %7.3f  \n" % ( ka, kd))
            oname.write("\n Exit Flow Rate \n")
            for it in range  (0, kk):
                oname.write(" %6.4f  , %10.5e \n" % (tau[it], F[it]))
        print ('Data are saved successfully in the file < %s >.' %(file_named))
    return
# Specify widgets
h1=widgets.HTML(value="<b> Diffusion and irreversible adsorption in one-zone TAP reactor</b>")
h2=widgets.HTML("<br> ")
h3=widgets.HTML("<i> Specify adsorption/desorption parameters</i>")
display(h1)
display(h2)
display(h3)
#
form = widgets.VBox()
ka = widgets.BoundedFloatText(description="ka:",
                              value=5.0, min=0.0, max=30.0,
                              padding = 4)
kd = widgets.BoundedFloatText(description="kd:",
                              value=5.0, min=0.0, max=30.0,
                              padding = 4)
button = widgets.Button(description="Click to Run Program",color="red",background_color= "lightgray")
checkbox1 = widgets.Checkbox(description="Save plot ", value=False)
checkbox2 = widgets.Checkbox(description="Save data ", value=False)
form.children = [ka, kd, checkbox1, checkbox2, button]
display(form)
#
h4 = widgets.HTML("<br><i> Simulation started, be patient!</i><br>")
display(h4)
h4.visible=False
#
```

```
#
def on_button_clicked(b):
    fmain(ka.value, kd.value, checkbox1.value, checkbox2.value)
    h4.visible=False
#
button.on_click(on_button_clicked)
```

Appendix C. Computer code for simulation of transient responses of reversible adsorption

```
%matplotlib inline
import numpy as np
from scipy.integrate import ode
import matplotlib.pyplot as plt
from ipywidgets import *
#from notebook import widgets
from IPython.display import clear_output, display, HTML
#
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#
def ff(tau, y, params):
# specify ODEs to solve
# allocate array of time derivatives
    dcdt = np.zeros(len(y))
# unpack parameters
    ka, kd, nz = params
# Allocate arrays of gaseous components and surface intermediates
    Ca= np.zeros(nz)
    Sa = np.zeros(nz)
# Split y array into arrays of gaseous components and surface intermediates:
#  Axial distribution of gas-phase reacting components
    Ca[0:nz]  = y[0:nz]
#  Axial distribution of surface intermediates
    Sa[0:nz]  = y[nz:2*nz]
    delta_z = 1./float(nz + 1)
# set the delta function on interval 2*Δt
    delta_t =0.0001
    if tau > 2*delta_t:
        Delta = 0.0
    else:
        Delta = (1./(2*delta_t)) * (1 + np.sin((np.pi/delta_t)*(tau-delta_t/2)))
# boundary condition at reactor inlet
    Ca_0  = -Ca[1]/3. + 4.*Ca[0]/3. +  Delta*2*delta_z/3.
    if Ca_0 < 0.0:
        Ca_0 = 0.0
# boundary condition at reactor outlet
    CaN1 = 0.0
# differential equation at first internal point
    dcdt[0] = (Ca[1] - 2.*Ca[0] + Ca_0)/(delta_z**2) - ka*Ca[0] + kd*Sa[0]
# differential equation at last internal point
    dcdt[nz-1] = (CaN1 - 2.*Ca[nz-1] + Ca[nz-2])/(delta_z**2) - ka*Ca[nz-1] + kd*Sa[nz-1]
# differential equations at 2,...,nz-1 internal points
    for i in range (1, nz-1):
#
        dcdt[i] = (Ca[i+1] - 2.*Ca[i] + Ca[i-1])/(delta_z**2) - ka*Ca[i] + kd*Sa[i]
#
    for i in range (0, nz):
        dcdt[nz+i] =  ka*Ca[i] - kd*Sa[i]
    return dcdt
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# Main program

def fmain (ka, kd, cbox_plot, cbox_data):
# set header
    h4.visible=True
# specify the number of internal points
    N = 100
# calculate the length of spatial interval
    delta_z = 1./float(N + 1)
# total number of points = number of internal points +
#    two boundary points
    ntpt = N + 2
```

```
# specify uniform grid of internal points
    z = np.linspace(delta_z, 1.-delta_z, N)
# specify the grid of all axial points, i.e. internal points + boundary points
    zz = np.zeros(ntpt)
    zz[0] = 0.
    zz[1:N+1] = z[0:N]
    zz[N+1] = 1.
# combine parameters to pass to the function f, params = [...,...]
    params = [ka, kd, N]
# allocate memory and set initial conditions
    NN = 2*N
    y = np.zeros(NN)
    y[0] = 0.
# set the time range for integration
    tau_start = 0.00001
    tau_final = 5.0
# specify the time interval for data collection
    delta_tau = 0.05
# calculate the number of time steps required with an additional
# step for initial condition
    num_steps = int(np.floor((tau_final - tau_start)/delta_tau) + 1)
# Prepare plots
    fig, ax = plt.subplots()
# Specify the ODE integrator parameters:
# The vode integrator with backward differentiation formula is used to solve
# the system of stiff ODEs.
# The absolute tolerance is atol, the relative tolerance is rtol,
# the number of steps is nsteps and the first-step size is first_step
    rint = ode(ff).set_integrator('vode', method='bdf', nsteps=50000,
                                  atol = 1.0e-06, rtol = 1.0e-06, first_step=1.0e-6)
    rint.set_initial_value(y, tau_start).set_f_params(params)
# Create vectors to store components axial distributions at specified time steps
    tau = np.zeros((num_steps, 1))
    F = np.zeros(num_steps)
    C_at_tau = np.zeros((num_steps, ntpt))
# Save initial values
    tau[0] = tau_start
    C_at_tau [0,1:N+1] = y [0:N]
    F[0] = 0.0
# Integrate the system of ODEs across each delta_t timestep
    kk = 1
    while rint.successful() and kk < num_steps:
        rint.integrate(rint.t + delta_tau)
#   store the results of integration:time
        tau[kk] = rint.t
#    concentration
        C_at_tau[kk,1:N+1] = rint.y[0:N]
        C_at_tau[kk,N+1]   = 0.
        C_at_tau[kk,0]     = -rint.y[1]/3. + 4.*rint.y[0]/3.
#    exit flow rate
        F [kk] = -(-4*rint.y[N-1]+rint.y[N-2])/(2.*delta_z)
#   plot the transient response of dimensionless exit flow rate
        clear_output(wait=True)
        ax.cla()
        ax.plot(tau[0:kk], F[0:kk], color="blue", linestyle="solid", linewidth=2)
        ax.set_ylabel(r'Dimensionless exit flow rate, $\bar F_A$  [-]')
        ax.set_xlabel('Dimensionless time, $tau$ [-]')
        ax.set_title('Diffusion + Adsorption: ka = %s, kd = %s' %(ka, kd))
        ax.set_xlim ([0.0, 5.0])
        ax.set_ylim ([0.0, 2.0])
        display(fig)
# end of time cycle
        kk += 1
#
```

```
# end of time cycle
        kk += 1
#
    plt.close()
# prepare the file name
    file_name1 = 'Ka'+str(ka)+'Kd'+str(kd)
    file_name = file_name1.replace(".", "_")
# print and save plot if required
    if  cbox_plot:
#        Comparison of numerical solution with analytical one
        plt.plot(Tau,Fa,'ro', label='analytical')
        file_namep = file_name+'.pdf'
        plt.plot (tau, F, 'b-', label = 'numerical')
        plt.xlabel('Dimensionless time, $tau$ [-]')
        plt.ylabel(r'Dimensionless exit flow rate, $\bar F_A$ [-]')
        plt.title ('Transient response, Diffusion + Adsorption: ka = %s, kd = %s \n' %(ka, kd))
        plt.legend(loc='upper right')
        plt.xlim ([0.0, 5.0])
        plt.ylim ([0.0, 2.0])
        plt.savefig(file_namep,dpi = 600)
        plt.show()
        print ('Plot is saved in the file < %s >.' %(file_namep))
# save simulation results as a csv file if required
    if  cbox_data:
        file_named = file_name+'.csv'
        with open(file_named,"w") as oname:
            oname.write(" ntpt ,  %5.0f \n" % ( ntpt))
            oname.write(" ka   ,  %7.3f, kd ,  %7.3f  \n" % ( ka, kd))
            oname.write("\n Exit Flow Rate \n")
            for it in range  (0, kk):
                oname.write(" %6.4f  , %10.5e \n" % (tau[it], F[it]))
        print ('Data are saved successfully in the file < %s >.' %(file_named))
    return
# Specify widgets
h1=widgets.HTML(value="<b> Diffusion and reversible adsorption in one-zone TAP reactor</b>")
h2=widgets.HTML("<br> ")
h3=widgets.HTML("<i> Specify adsorption/desorption parameters</i>")
display(h1)
display(h2)
display(h3)
#
form = widgets.VBox()
ka = widgets.BoundedFloatText(description="ka:",
                              value=5.0, min=0.0, max=30.0,
                              padding = 4)
kd = widgets.BoundedFloatText(description="kd:",
                              value=5.0, min=0.0, max=30.0,
                              padding = 4)
button = widgets.Button(description="Click to Run Program",color="red",background_color= "lightgray")
checkbox1 = widgets.Checkbox(description="Save plot ", value=False)
checkbox2 = widgets.Checkbox(description="Save data ", value=False)
form.children = [ka, kd, checkbox1, checkbox2, button]
display(form)
#
h4 = widgets.HTML("<br><i> Simulation started, be patient!</i><br>")
display(h4)
h4.visible=False
#
def on_button_clicked(b):
    fmain(ka.value, kd.value, checkbox1.value, checkbox2.value)
    h4.visible=False
#
button.on_click(on_button_clicked)
```

Appendix D. Computer code for simulation of transient responses of Langmuir-Hinshelwood and Eley-Rideal reaction mechanism

```
%matplotlib inline
import numpy as np
from scipy.integrate import ode
import matplotlib.pyplot as plt
from ipywidgets import *
#from notebook import widgets
from IPython.display import clear_output, display, HTML
def ff(t, y, params):
# specify ODEs to solve
# allocate array of time derivatives
    yt = np.zeros(len(y))
# unpack parameters
    k1a, k1d, k2a, k2d, k3, k4a, k4d, k5, nz, Nk, Dk, Rho, e, A, L= params
# Allocate arrays of gaseous components and surface intermediates
    c_CO = np.zeros(nz)
    s_CO = np.zeros(nz)
    c_O  = np.zeros(nz)
    s_O  = np.zeros(nz)
    c_CO2= np.zeros(nz)
    s_OM = np.zeros(nz)
# Split y array into arrays of gaseous components and surface intermediates:
#  Axial distribution of gas-phase reacting components
    c_CO[0:nz]  = y[0:nz]
    c_O[0:nz] = y[nz:2*nz]
    c_CO2[0:nz]=y[2*nz:3*nz]
#  Axial distribution of surface intermediates
    s_CO[0:nz]  = y[3*nz:4*nz]
    s_O[0:nz] = y[4*nz:5*nz]
    s_OM[0:nz]= y[5*nz:6*nz]
    dz = L/float(nz + 1)
# set the delta function on interval 2*Δt
    dt =0.001
    if t > 2*dt:
        delta = 0.0
    else:
        delta = (1./(2*dt)) * (1 + np.sin((np.pi/dt)*(t-dt/2)))
# boundary condition at reactor inlet
    c_CO0  = -c_CO[1]/3. + 4.*c_CO[0]/3.+ (delta*2*dz*Nk)/(3.*A*e*Dk)
    c_O0  = -c_O[1]/3. + 4.*c_O[0]/3. + (delta*2*dz*Nk)/(3.*A*e*Dk)
    c_CO20 = 0 #-c_CO2[1]/3. + 4.*c_CO2[0]/3.
    if c_CO0 < 0.0 and c_O0 < 0:
        c_CO0 = 0.0
        c_O0 = 0.0
# boundary condition at reactor outlet
    c_CON1 = 0.0
    c_ON1 = 0.0
    c_CO2N1 = 0
    for i in range (0, nz):
        r_CO =  - k1a*c_CO[i] + k1d*s_CO[i] - k5*c_CO[i]*s_OM[i]
        r_O =   - k2a*c_O[i] + k2d*s_O[i]**2 - k4a*c_O[i] + k4d*s_OM[i]**2
        r_CO2 =    - k3*s_CO[i]*s_O[i] - k5*c_CO[i]*s_OM[i]
        if (i==0):
            dc_CO = Dk*(c_CO[1] - 2.*c_CO[0] + c_CO0)/(dz**2) + r_CO*(Rho/e)
            dc_O = Dk*(c_O[1] - 2.*c_O[0] + c_O0)/(dz**2) + r_O*(Rho/e)
            dc_CO2 = Dk*(c_CO2[1] - 2.*c_CO2[0] + c_CO20)/(dz**2) + r_CO2*(Rho/e)
        elif (i==nz-1):
            dc_CO =  Dk*(c_CON1 - 2.*c_CO[nz-1] + c_CO[nz-2])/(dz**2) + r_CO*(Rho/e)
            dc_O = Dk*(c_ON1 - 2.*c_O[nz-1] + c_O[nz-2])/(dz**2) + r_O*(Rho/e)
            dc_CO2 = Dk*(c_CO2N1 - 2.*c_CO2[nz-1] + c_CO2[nz-2])/(dz**2) + r_CO2*(Rho/e)
        else:
            dc_CO = Dk*(c_CO[i+1] - 2.*c_CO[i] + c_CO[i-1])/(dz**2) + r_CO*(Rho/e)
            dc_O = Dk*(c_O[i+1] - 2.*c_O[i] + c_O[i-1])/(dz**2) + r_O*(Rho/e)
            dc_CO2 = Dk*(c_CO2[i+1] - 2.*c_CO2[i] + c_CO2[i-1])/(dz**2) + r_CO2*(Rho/e)

        ds_CO = k1a*c_CO[i] - k1d*s_CO[i] - k3*s_CO[i]*s_O[i]
        ds_O = 2*k2a*c_O[i] -2*k2d*s_O[i]**2 - k3*s_CO[i]*s_O[i]
        ds_OM= 2*k4a*c_O[i] - 2*k4d*(s_OM[i])**2 - k5*c_CO[i]*s_OM[i]
```

```
        yt[i] = dc_CO
        yt[nz+i] = dc_O
        yt[2*nz+i] = dc_CO2
        yt[3*nz+i] = ds_CO
        yt[4*nz+i] = ds_O
        yt[5*nz+i] = ds_OM
    return yt
# Main program
def fmain (k1a, k1d, k2a, k2d, k3, k4a, k4d, k5, cbox_plot, cbox_data):
# set header
    h4.visible=True
# specify the number of internal points
    nz = 50
# specify simulation environment
    Dk=34
    Rho=1.7
    e=0.38
    A=3.1416 # Area is taken from D=2 cm
    Nk=1e6
    L=2.54
    dz = L/float(nz + 1)
# total number of points = number of internal points +
#    two boundary points
    ntpt = nz + 2
# specify uniform grid of internal points
    z = np.linspace(dz, L-dz, nz)
# specify the grid of all axial points, i.e. internal points + boundary points
    zz = np.zeros(ntpt)
    zz[0] = 0.
    zz[1:nz+1] = z[0:nz]
    zz[nz+1] = L
# combine parameters to pass to the function f, params = [...,...]
    params = [k1a, k1d, k2a, k2d, k3,k4a, k4d,k5, nz, Nk, Dk, Rho, e, A, L]
# allocate memory and set initial conditions
    NN = 6*nz
    y0 = np.zeros(NN)
    y0[0] = 0.
# set the time range for integration
    t_start = 0.0001
    t_final = 1
# specify the time interval for data collection
    delta_t = 0.01
# calculate the number of time steps required with an additional
# step for initial condition
    num_steps = int(np.floor((t_final - t_start)/delta_t) + 1)
# Prepare plots
    fig, ax = plt.subplots()
# Specify the ODE integrator parameters:
# The vode integrator with backward differentiation formula is used to solve
# the system of stiff ODEs.
# The absolute tolerance is atol, the relative tolerance is rtol,
# the number of steps is nsteps and the first-step size is first_step
    rint = ode(ff).set_integrator('vode', method='bdf', nsteps=50000,
                                  atol = 1.0e-06, rtol = 1.0e-06, first_step=1.0e-06)
    rint.set_initial_value(y0, t_start).set_f_params(params)
# Create vectors to store components axial distributions at specified time steps
    t = np.zeros((num_steps, 1))
    C_at_t = np.zeros((num_steps, NN))
    F_CO2 = np.zeros(num_steps)
    F_CO = np.zeros(num_steps)
    F_O = np.zeros(num_steps)
# Save initial values
    t[0] = t_start
    C_at_t [0,0:NN] = y0[0:NN]
    F_CO2[0] = 0
    F_CO[0] = 0
    F_O[0] = 0
```

```
# Integrate the system of ODEs across each delta_t timestep
    kk = 1
    while rint.successful() and kk < num_steps:
        rint.integrate(rint.t + delta_t)
#   store the results of integration:  time
        t[kk] = rint.t
        C_at_t[kk,1:nz+1] = rint.y[0:nz]
        C_at_t[kk,nz+1]   = 0.
        C_at_t[kk,0]      = -rint.y[1]/3. + 4.*rint.y[0]/3.
#    exit flow rate
        F_CO[kk] = -((-4*rint.y[nz-1]+rint.y[nz-2])*(A*Dk))/(2.*dz*Nk)
        F_O[kk] = -((-4*rint.y[2*nz-1]+rint.y[2*nz-2])*(A*Dk))/(2.*dz*Nk)
        F_CO2 [kk] = ((-4*rint.y[3*nz-1]+rint.y[3*nz-2])*(A*Dk))/(2.*dz*Nk)
#   plot the transient response of dimensionless exit flow rate
        clear_output(wait=True)
        ax.cla()
        ax.plot(t[0:kk], F_CO[0:kk], color="blue", linestyle="solid", linewidth=2)
        ax.plot(t[0:kk], F_O[0:kk], color="red", linestyle="solid", linewidth=2)
        ax.plot(t[0:kk], F_CO2[0:kk], color="green", linestyle="solid", linewidth=2)
        ax.set_ylabel(r'Exit flow rate, $\bar F/Nk$  [1/s]')
        ax.set_xlabel('Time, $t$ [s]')
        ax.set_title('Diffusion + Adsorption: T = 130, Nk = %s, ' %(Nk))
        ax.set_xlim ([0.0, 1])
        ax.set_ylim ([-1, 15])
        display(fig)
# end of time cycle
        kk += 1
    plt.close()
# prepare the file name
    file_name1 = 'K1a'+str(k1a)+'K1d'+str(k1d)+'K2a'+str(k2a)+'K2d'+str(k2d)+'K3'+str(k3)+'K4a'+str(k4a)+'K4d'+str(k4d)+'K5'
    file_name = file_name1.replace(".", "_")
# print and save plot if required
    if  cbox_plot:
        file_namep = file_name+'.pdf'
        plt.plot (t, F_CO, 'b-', label = 'CO')
        plt.plot (t, F_O, 'r-', label = 'O')
        plt.plot (t, F_CO2, 'g-', label = 'CO2')
        plt.xlabel('Time, $t$ [s]')
        plt.ylabel(r'Exit flow rate, $\bar F/Nk$ [1/s]')
        plt.title ('Transient response: T=130, Nk = %s \n' %(Nk))
        plt.legend(loc='upper right')
        plt.xlim ([0.0, 1])
        plt.ylim ([-1,15])
        plt.savefig(file_namep,dpi = 600)
        plt.show()
        print ('Plot is saved in the file < %s >.' %(file_namep))
# save simulation results as a csv file if required
    if  cbox_data:
        file_named = file_name+'.csv'
        with open(file_named,"w") as oname:
            oname.write(" ntpt ,  %5.0f \n" % ( ntpt))
            oname.write(" k1a  ,  %7.3f, k1d ,  %7.3f, k2a  ,  %7.3f, k2d ,  %7.3f, k3 ,  %7.3f, k4a  ,  %7.3f, k4d ,  %7.3f
            oname.write("\n Exit Flow Rate \n")
            for it in range  (0, kk):
                oname.write(" %6.4f  , %10.5e \n" % (t[it], F_CO2[it]))
        print ('Data are saved successfully in the file < %s >.' %(file_named))
    return
```

```
# Specify widgets
h1=widgets.HTML(value="<b> Diffusion and reversible adsorption in one-zone TAP reactor</b>")
h2=widgets.HTML("<br> ")
h3=widgets.HTML("<i> Specify adsorption/desorption parameters</i>")
display(h1)
display(h2)
display(h3)
form = widgets.VBox()
k1a = widgets.BoundedFloatText(description="k1a:",
                               value=77, min=0.0, max=2000.0,
                               padding = 4)
k1d = widgets.BoundedFloatText(description="k1d:",
                               value=3.09e-1, min=0.0, max=2000.0,
                               padding = 4)
k2a = widgets.BoundedFloatText(description="k2a:",
                               value=77, min=0.0, max=2000.0,
                               padding = 4)
k2d = widgets.BoundedFloatText(description="k2d:",
                               value=2.02e-1, min=0.0, max=2000.0,
                               padding = 4)
k3 = widgets.BoundedFloatText(description="k3:",
                               value=1.54e-1, min=0.0, max=2000.0,
                               padding = 4)
k4a = widgets.BoundedFloatText(description="k4a:",
                               value=18.4, min=0.0, max=2000.0,
                               padding = 4)
k4d = widgets.BoundedFloatText(description="k4d:",
                               value=1.75e-1, min=0.0, max=2000.0,
                               padding = 4)
k5 = widgets.BoundedFloatText(description="k5:",
                               value=3.67e-2, min=0.0, max=2000.0,
                               padding = 4)
button = widgets.Button(description="Click to Run Program",color="red",background_color= "lightgray")
checkbox1 = widgets.Checkbox(description="Save plot ", value=False)
checkbox2 = widgets.Checkbox(description="Save data ", value=False)
form.children = [k1a, k1d, k2a, k2d, k3, k4a, k4d, k5, checkbox1, checkbox2, button]
display(form)
#
h4 = widgets.HTML("<br><i> Simulation started, be patient!</i><br>")
display(h4)
h4.visible=False
#
def on_button_clicked(b):
    fmain(k1a.value, k1d.value, k2a.value, k2d.value, k3.value, k4a.value, k4d.value, k5.value, checkbox1.value, checkbox2.value)
    h4.visible=False

#
button.on_click(on_button_clicked)
```