

HEAT TRANSFER ENHANCEMENT DUE TO SURFACE OXIDATION OF METHANE
ON PLATINUM

By

SUNGSIK KIM

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2011

© 2011 Sungsik Kim

To my family

ACKNOWLEDGMENTS

I thank my parents to support me to study at University of Florida and sister to help me. Moreover, I thank my friend, Gobong to recommend me this school. And, I thank my advisor, Dr. David W. Mikolaitis to give me a chance to research methane under enhancement of heat transfer and to teach me numerical programs and chemical kinetics program. Finally, I thank my advisor to lead me to catalytic combustion field.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS.....	4
LIST OF TABLES.....	7
LIST OF FIGURES.....	8
ABSTRACT	11
CHAPTER	
1 INTRODUCTION	13
1.1 Review of Catalytic Combustion	13
1.1.1 Surface Reaction	14
1.1.2 Chemical Kinetics	15
1.1.3 Methane	16
1.1.4 Catalytic Methane Research	17
1.2 Motivation of Current Study	20
2 EXPERIMENTAL METHOD.....	22
2.1 Modeling Method	22
2.2 Simulation Method	27
2.3 Heat Transfer Calculation Method	30
2.4 Fuel Consumption.....	32
3 RESULTS AND DISCUSSION	39
3.1 Analysis of Fuel Consumption and Mole Fraction.....	39
3.1.1 New Mechanism with Faster Deactivation Reaction.....	39
3.1.2 New Mechanism with Faster Dissociation Reaction	40
3.2 Analysis of Heat Transfer.....	40
3.2.1 New Mechanism with Faster Deactivation Reaction.....	40
3.2.2 New Mechanism with Faster Dissociation Reaction	42
4 SUMMARY	53
APPENDIX	
A ORIGINAL CATCOMB CODE	55
B MODIFIED CATCOMB CODE FOR LOW IGNITION BRANCH	60
C HEAT TRANSFER IN CONVENTIONAL MECHANISM	95

D	HEAT TRANSFER IN NEW MECHANISM WITH FASTER DEACTIVATION.....	96
E	ENHANCEMENT OF HEAT TRANSFER IN NEW MECHANISM WITH FASTER DEACTIVATION	97
F	HEAT TRANSFER IN LOW IGNITION BRANCH IN NEW MECHANISM WITH FASTER DISSOCIATION.....	100
G	HEAT TRANSFER IN HIGH IGNITION BRANCH IN NEW MECHANISM.....	104
H	ENHANCEMENT OF HEAT TRANSER.....	106
	LIST OF REFERENCES	115
	BIOGRAPHICAL SKETCH.....	117

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 Detailed surface mechanism of Deutschmann	33
2-2 NASA coefficients of H ₂ O	33
2-3 NASA coefficients of H ₂ O*	34
2-4 NASA coefficients of CO ₂	34
2-5 NASA coefficients of CO ₂ *	34
2-6 Enthalpy of species at 700K..	34
2-7 Surface reactions in new mechanism	35
2-8 Gas reactions in new mechanism with faster deactivation reaction.....	35
2-9 Gas reactions in new mechanism with the faster dissociation reaction	36

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1 Surface reaction process	21
2-1 Reaction path of $H(s) + OH(s)$ in Deutschmann mechanism.....	36
2-2 Reaction path of $OH(s) + OH(s)$ and $H_2O(s)$ in Deutschmann mechanism.....	36
2-3 Reaction path of $CO(s) + O(s)$ and $CO_2(s)$ in Deutschmann mechanism	37
2-4 Reaction path of reaction of $H(s) + O(s)$ in modeling.....	37
2-5 Reaction path of $CO(s) + O(s)$ in modeling.....	37
2-6 Stagnation flow	38
2-7 Energy balance.....	38
3-1 Mole fraction of species in Deutschmann at 900K.....	44
3-2 Axial velocity profile in Deutschmann at 900K.....	44
3-3 Temperature profile in Deutschmann at 900K	44
3-4 Density profile in Deutschmann at 900K.....	45
3-5 Specific enthalpy in Deutschmann at 900K	45
3-6 Mole fraction of species in new mechanism with faster deactivation at 900K.....	46
3-7 Axial velocity profile in new mechanism with faster deactivation at 900K.....	46
3-8 Temperature profile in new mechanism with faster deactivation at 900K	46
3-9 Density profile in new mechanism with faster deactivation at 900K.....	47
3-10 Specific enthalpy in new mechanism with faster deactivation at 900K	47
3-11 Mole fraction of species in new mechanism with faster dissociation at 900K.....	48
3-12 Axial velocity profile in new mechanism with faster dissociation at 900K	48
3-13 Temperature profile in new mechanism with faster dissociation at 900K	48
3-14 Density profile in new mechanism with faster dissociation at 900K.....	49
3-15 Specific enthalpy in new mechanism with faster dissociation at 900K.....	49

3-16	Heat transfer to non-reacting flow in Deutschmann.....	50
3-17	Heat transfer to non-reacting flow in new mechanism with faster deactivation...	50
3-18	Heat transfer to non-reacting flow in conventional and new mechanism	51
3-19	Enhancement of heat transfer in new mechanism with faster deactivation	51
3-20	Heat transfer to non-reacting flow in new mechanism with faster dissociation ...	52
3-21	Enhancement of heat transfer in new mechanism with faster dissociation.....	52
C-1	Heat flux to non-reacting flow in Deutschmann	95
D-1	Heat flux to non-reacting flow in new mechanism with faster deactivation	96
E-1	Heat flux to non-reacting flow from 320K to 560K in new mechanism with faster deactivation	97
E-2	Heat flux to non-reacting flow from 570K to 800K in new mechanism with faster deactivation	98
E-3	Heat flux to non-reacting flow from 820K to 1100K in new mechanism with faster deactivation	99
F-1	Heat flux to non-reacting flow from 320K to 605K in low ignition branch in new mechanism.....	100
F-2	Heat flux to non-reacting flow from 610K to 905K in low ignition branch in new mechanism.....	101
F-3	Heat flux to non-reacting flow from 910K to 1015.2K in low ignition branch in new mechanism.....	102
F-4	Heat flux to non-reacting flow from 916K to 937K in low ignition branch in new mechanism.....	103
G-1	Heat flux to non-reacting flow from 616.5K to 910K in high ignition branch in new mechanism.....	104
G-2	Heat flux to non-reacting flow from 915K to 1080K in high ignition branch in new mechanism.....	105
H-1	Enhancement of heat transfer from 310K to 455K in low ignition branch in new mechanism.....	106
H-2	Enhancement of heat transfer from 460K to 605K in low ignition branch in new mechanism.....	107

H-3	Enhancement of heat transfer from 610K to 755K in low ignition branch in new mechanism.....	108
H-4	Enhancement of heat transfer from 610K to 905K in low ignition branch in new mechanism.....	109
H-5	Enhancement of heat transfer from 910K to 1015K in low ignition branch in new mechanism.....	110
H-6	Enhancement of heat transfer from 616.5K to 760K in high ignition branch in new mechanism.....	111
H-7	Enhancement of heat transfer from 765K to 910K in high ignition branch in new mechanism.....	112
H-8	Enhancement of heat transfer from 915K to 985K in high ignition branch in new mechanism.....	113
H-9	Enhancement of heat transfer from 990K to 1080K in high ignition branch in new mechanism.....	114

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

HEAT TRANSFER ENHANCEMENT DUE TO SURFACE OXIDATION OF METHANE ON PLATINUM

By

Sungsik Kim

August 2011

Chair: Mikolaitis W, David
Major: Mechanical Engineering

A study of methane is conducted on the conditions of premixed lean methane air mixtures with platinum surface as a catalyst in one dimensional stagnation flow in steady state to understand enhancement of heat transfer under a new surface mechanism of methane oxidation. Premixed methane/air mixtures flowing onto platinum surface of various surface temperatures are simulated. Enhancement of heat transfer mechanism of catalytic methane oxidation is modeled based on GRI30 mechanism for gas reactions and Deutschmann mechanism for surface reactions. In the proposed surface mechanism, it is assumed that activated H_2O and CO_2 desorbs directly from platinum surface without elementary surface reaction and that activated H_2O and CO_2 are dissociated or deactivated by third body reactions. Thus, three elementary surface reactions are replaced in Deutschmann mechanism and four elementary reactions are added in GRI30 mechanism. Deutschmann mechanism and new catalytic methane mechanism are simulated and heat flux to non-reaction flow is calculated to see how much heat is required to maintain a given surface temperature. By comparing to conventional catalytic methane combustion, new mechanism shows different phenomena in ignition temperature, fuel consumption at each surface temperature,

most importantly, heat transfer to non-reacting flow where there is no reactions due to enhancement of heat transfer by adsorption and desorption. In addition, heat transfer to non-reacting flow and enhancement of heat transfer are different with different rates of dissociation and deactivation reactions. Because of insufficient information about heat transfer to non-reacting flow, more research on enhancement of heat transfer is required to prove this mechanism.

CHAPTER 1 INTRODUCTION

1.1 Review of Catalytic Combustion

Catalytic combustion becomes more an issue in that it reduces emission of NO_x because of lower operating temperature. Generally, it works by providing an alternative mechanism involving a different transition state and lower activation energy. The advantages of the catalytic combustion⁹ is that it does not require the presence of a flame, nor an ignition source like a spark or pilot flame, but there is a minimum inlet gas temperature required to have a sufficiently high catalyst activity to achieve complete combustion. Second, it is operated at low temperature in which NO_x would not be formed. Thus, it is cleaner combustion process. Finally, it may enable the design of more compact furnaces and reactors heated by combustion reactions to be contemplated.

Catalytic combustion is a heterogeneous reaction that consists of two phase reaction which is a gas-solid phase reaction. Surface reactions are important in many combustion applications such as in wall recombination process during auto ignition, in coal combustion, in soot formation and oxidation, in catalytic combustion or in metal combustion. Rate of surface reaction varies with surface solid. It means that rate of surface reaction can be significantly increased or decreased by the catalyst¹³ that is substance attached to surface which affects rate of reaction without being consumed. If catalyst speed the reaction, it is called positive catalysts whereas if it slows the reaction, it is called inhibitors. And this phenomenon that rate of surface reaction is influenced by catalyst is called catalysis.

1.1.1 Surface Reaction

An early and important discovery in the history of surface catalysis was the observation by Faraday that molecules must first become attached to, or adsorb on, a surface before they can react. Figure 1-1 shows mechanism of surface reaction. First, gas molecules are adsorbed onto surface which is called adsorption, and surface reaction of molecules occurs on surface. After reaction occurs, products escape from surface so called desorption. According to Gardiner, the overall process of gas-solid reactions⁷ can be divided into several sub-processes.

- 1) Transport of the reactant molecule to the surface by convection and/or diffusion.
- 2) Adsorption of the reactant molecules on the surface.
- 3) Elementary reaction steps, involving various combinations of adsorbed molecules the surface itself, and gas-phase molecules.
- 4) Desorption of the product molecules from the surface.
- 5) Transport of the product molecules away from the surface by convection and/or diffusion.

This is well-known as the Langmuir-Hinshelwood mechanism in the modern treatment of surface reaction.

The mutual attraction between an approaching molecules and a surface can be attributed to two types of interactions. In the early 1900s, Langmuir first investigated the process of adsorption and developed the ideas of sticking and trapping. There are two main types of adsorption, and they are distinctly different. Physisorption is the forces that are of a physical nature and the adsorption by physisorption is relatively weak. There is no direct chemical bond between the adsorbate and surface. The adsorbate is held by physical forces such as van der Waals forces. And chemisorption is considerably stronger and the adsorbed molecules are attached to the surface by

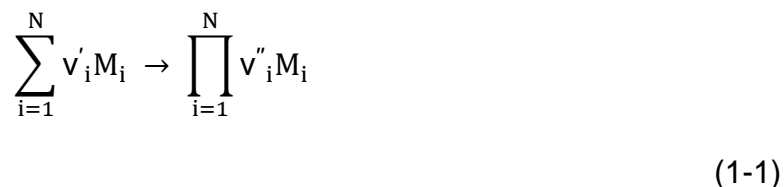
valence forces of the same type as those occurring between bound atoms in molecules. It occurs when the adsorbate and surface have a direct chemical bond causing the sharing of electrons. The concept of chemisorptions was developed by Taylor⁷, Keier and Roginsky⁷, Kummer and Emmett⁷, Constable⁷ and many others.

The Langmuir-Rideal mechanism⁷ or Rideal-Eley mechanism⁷ represents the reaction between a gas molecule and an adsorbed molecule. In the precursor mechanism⁷, species B is adsorbed on the metal catalyst surface, species A has just a momentary residence on the surface without forming a bond on the surface, and the reaction product AB is immediately formed.

In the desorption of product species, molecules requires sufficient energy to overcome the bond strength between the adsorbed species and the surface. If the desorption process of product species does not occur quickly at the catalytic surface, product species can become saturated to stop the surface reaction process.

1.1.2 Chemical Kinetics

All chemical reactions have different rate of reaction under same conditions. It is affected by concentrations the chemical compounds, temperature, pressure, presence of a catalyst or inhibitor, and radiative effects. One-step chemical reaction of arbitrary complexity can be represented by stoichiometric equation⁷.



Rate of reaction (RR)⁷ of a chemical product species is proportional to the products of the concentrations of the reacting chemical species.

$$RR = \frac{dC_{\text{product}}}{dt} = -\frac{dC_{\text{reactant}}}{dt} = k \prod_{i=1}^N (C_{M_i})^{v_i} \quad (1-2)$$

The coefficient k is the proportionality constant called the specific reaction-rate constant. For a given chemical reaction, k is independent of the concentrations C_{M_i} (kmol/m³) and depends only on the temperature. The Swedish chemist and physicist Svante Arrhenius (1859-1927) stated that only those molecules that possess energy greater than a certain amount E_a will react and these high-energy, active molecules lead to products. Following is Arrhenius law⁷.

$$k = AT^b \exp\left(-\frac{E_a}{R_u T}\right) \quad (1-3)$$

Where AT^b is the collision frequency and the exponential term is called the Boltzmann factor that represents the fraction of collisions that have energy levels greater than the activation energy E_a and it is assumed to include the effect of the collision terms, the steric factor associated with the orientation of the colliding molecules, and the mild temperature dependence of the pre-exponential factor. The values of A , b whose value generally is between 0 and 1, and E_a are based on the nature of the elementary reaction.

1.1.3 Methane

Methane¹² is well-known as natural gas. It is the principle component of natural gas. It is discovered and isolated by Alessandro Volta between 1776 and 1778. It is attractive as a fuel because it reduces pollution and maintains a clean and healthy environment and is abundant and is secure source of energy. Moreover, burning methane¹² produces less carbon dioxide for each unit of heat released compared to

other hydrocarbon fuels. It is used as a vehicle fuel and currently methane rocket research is being conducted by NASA¹⁴.

1.1.4 Catalytic Methane Research

Olaf Deutschmann¹ studied catalytic combustion and conversion of methane numerically in one dimensional flow configurations. In the catalytic combustion of methane, reactions of C1 and C2 species are included in homogeneous reactive flow, and surface reactions of catalytic methane oxidation includes 10 surface species and 26 reactions. CH₄ – air mixtures flow slowly onto a heated platinum foil. when catalytic ignition temperature is reached (depending on CH₄ – air ratio) because of the exothermic surface reactions that release heat, catalyst temperature rises rapidly and when heat from surface reactions is over a critical value, the reaction becomes self-accelerating and a new stationary state controlled by mass transport. And after ignition, the global process is controlled by diffusion of reactants toward the catalyst and products desorbed from the catalyst when there are free surface sites on platinum for methane and oxygen. Since temperature is increased for uncovered surface site to ignite, the ignition temperature is increased as methane-oxygen ratio is decreased. Further increases of the foil temperature after ignition results in homogeneous reaction ignition because of the different gradient of mole fraction of methane between on the surface and near the surface. In the catalytic conversion of methane, hydrocarbon mechanism that consists of 618 elementary of 54 chemical species including reactions of C1-C4 systems is used. Two models were made for catalytic conversion of methane with different surface mechanisms. In the first model, methane conversion to ethane and to ethylene continues since oxygen does not affect the production of the catalytic CH₃· even though consumption of oxygen is fast so oxidation of methane is over. In the

second model, it is concluded that the conversion process is almost over after all of oxygen is consumed.

Olaf Deutschmann² analyzed heterogeneous oxidation of methane on a platinum foil to simulate the experiments of Williams et al⁶. It is concluded that as surface temperature is increased by supplying electric power, ignition occurs around 600 °C. Because surface temperature is different from temperature of a non-catalytic surface, it shows that heat generated by surface reactions is important. Also, it is found that gas-phase reactions do not occur significantly due to the low temperature of the gases on the foil and that after ignition the power to the foil can be decreased to values below the ignition power without extinguishing the flame. Moreover, near the stoichiometric mixture, the chemical energy release is large enough to maintain the system ignited and auto thermal behavior is found.

Olaf Deutschmann³ investigated hydrogen assisted catalytic combustion of methane on platinum. It is made of experiment and two numerical simulations. In the experiment, methane/hydrogen/air mixtures flow through platinum coated honeycomb monoliths. An ignited pure hydrogen/air flow catalytically, and then methane is fed slowly with increasing its amount. It is concluded that the light-off temperature of oxidation of methane decreases with increasing hydrogen content and that light-off temperature increase with increasing hydrogen feed. In the simulation of stagnation flow on to platinum foil, oxidation of hydrogen can easily cause a temperature where oxidation of methane begins in hydrogen assisted catalytic ignition. However, too high temperature from hydrogen addition may damage catalyst and the surrounding technical device. In another simulation of flow via a single channel of honeycomb

monolith, when a lean hydrogen/air mixture is fed, the catalyst ignites and all the hydrogen is consumed and this leads to a rapid increase of monolith temperature whereas no significant amount of methane is converted without hydrogen addition and the catalyst temperature remains at 300K. Therefore, all the CH_4 is completely consumed even for very low methane concentration with temperature over 800K because light-off of methane combustion will occur immediately. Finally, it is concluded that hydrogen addition to the initial mixture makes catalytic combustion of methane on platinum light-off.

F. Moallemi⁴ analyzed catalytic combustion of methane air mixtures on platinum and palladium surface to see effects of operating temperature conditions on combustion, heat transfer efficiency and pollutant formation. It is found that surface temperature of Pd is higher than Pt with similar fuel concentration, and Pd yields higher flow rate. Moreover, Pd catalyst causes higher methane slippages which means methane leaves the surface faster than from a Pt catalyst. Therefore, it is concluded that ignition occurs easily in Pd catalyst at lower flow rate and flashback is occurred during the ignition period at higher flow rate.

C.A. Henry, D. Mikolaitis, P. Szedlacsek, and D.W. Hahn⁵ studied heat transfer under catalytic combustion of methane. It is focused on effect of heterogeneous chemistry on heat transfer enhancement. Heat transfer is measured under the condition of catalytic methane combustion using a concentric tube reactor with the catalytic reaction occurring in the annular space and a non reaction, cooling flow passing through the center tube. In order to evaluate the local heat transfer flux to the reacting flow stream along the axial direction, detailed measurements of the cooling flow axial

temperature profile are combined with a Langmuir-Hinshelwood mechanism for surface chemistry, and both local and global energy and species conservation. It is found that there is enhancements of 275% with respect to non-reacting convective heat flux for the fuel rich catalytic combustion of methane. Therefore, it is concluded that there is significant partitioning of the enthalpy of combustion in reacting cooling system.

1.2 Motivation of Current Study

Many catalytic methane researches have been conducted as explained above. There are many mechanisms for catalytic combustion. However, those are not focused on heat transfer to non-reacting flow. Those concentrate on combustion process. Moreover, in classic convention heat transfer of reacting flow in catalytic combustion, heat release from surface reactions due to adsorption and desorption are not considered. Therefore, conventional mechanisms are not well matched with respect to heat transfer to non-reacting flow. It is needed to develop new mechanisms that considers heat release from surface reactions due to adsorption and desorption of species to / from surface in heat transfer of reacting flow in catalytic combustion. An idea of new mechanism of catalytic methane is started from the research of C.A. Henry, D. Mikolaitis, P. Szedlacsek, and D.W. Hahn⁵ that there is enhancement heat transfer to cooling flow due to surface reactions in catalytic methane. New mechanism of catalytic methane also considers heat transfer to non-reacting flow and heat from adsorption and desorption. It is based on Olaf Deutschmann mechanism^{2,10} for surface reactions and GRI30 mechanism¹⁰ for gas reactions. Three reactions are replaced in surface reactions and four gas reactions are added in GRI30 mechanism¹⁰ by considering enhancement of heat transfer.

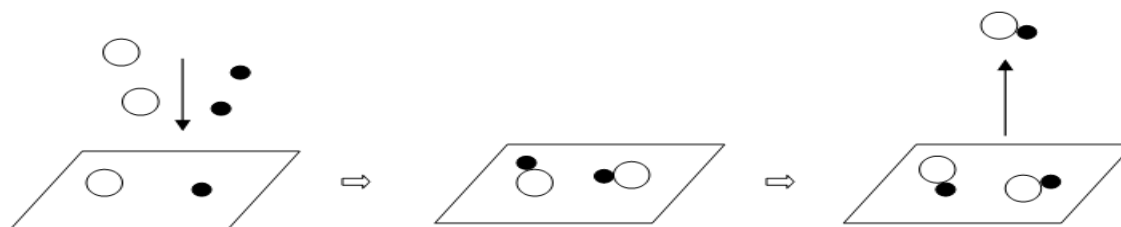
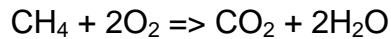


Figure 1-1. Surface reaction process

CHAPTER 2 EXPERIMENTAL METHOD

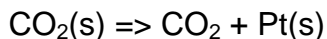
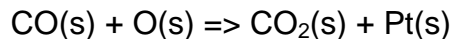
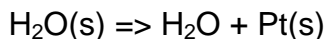
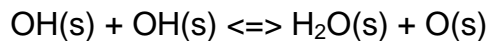
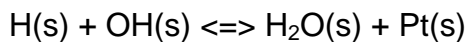
2.1 Modeling Method

Oxidation of methane with catalyst and that of with non-catalyst is different. With catalyst, some elementary reactions occur on the surface. Following is overall stoichiometric global reaction of methane with complete combustion



In general, it consists of many elementary reactions, and if methane is not completely burned, unwanted products such as NO_x , SO_x , and CO are formed in gas reactions.

In catalytic combustion, Deutschmann mechanism^{2,11} of methane is well known. New mechanism of enhancement of heat transfer is focused on H_2O and CO_2 because much energy is obtained from H_2O and CO_2 after surface reactions. Desorption of H_2O and CO_2 in Deutschmann mechanism^{2,11} occurs through some elementary reactions on surface. Detailed Deutschmann mechanism^{2,11} is shown in Table 2-1. In conventional catalytic combustion of methane based on Deutschmann mechanism^{2,11}, H_2O and CO_2 are desorbed by elementary reactions of



where " \Rightarrow " indicates irreversible reaction and " \rightleftharpoons " denotes reversible reaction.

Unlike conventional catalytic combustion, mechanism of enhancement heat transfer is started from the assumption that activated H_2O and CO_2 are desorbed directly from surface without sub-reactions in Deutschmann mechanism^{2,11} if certain amount of energy that exceeds activation energies of activated H_2O and CO_2 . This mechanism is based on GRI30 mechanism¹⁰ that is optimized mechanism and designed to model natural gas combustion including NO formation and reburn chemistry for gas reactions and Deutschmann mechanism^{2,11} for surface reactions. Therefore, some gas reactions are added in GRI30 mechanism¹⁰ and some surface reactions are replaced according to the assumption. To be specific, in modeling mechanism, activated H_2O whose symbol is H_2O^* is formed by irreversible reactions of $\text{H}(\text{S}) + \text{OH}(\text{S})$ and $\text{OH}(\text{S}) + \text{OH}(\text{S})$ and activated H_2O is dissociated into either $\text{H} + \text{OH}$ or $\text{H}_2\text{O} + \text{M}$. Moreover, activated CO_2 whose symbol is CO_2^* is formed by the reaction of irreversible $\text{CO}(\text{S}) + \text{O}(\text{S})$ and it is dissociated into either $\text{CO} + \text{O}$ or $\text{CO}_2 + \text{M}$.

Surface reactions that forms H_2O^*

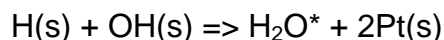
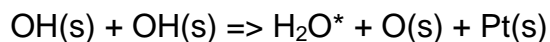
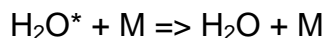
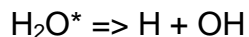


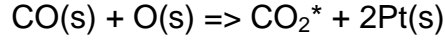
Figure 2-4 shows detailed reaction path of the reactions in modeling.

Dissociation of H_2O^*



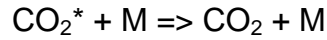
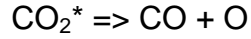
Here, M indicates third body reaction. Activated H_2O goes to stable state by transferring excess energy to M.

Surface reaction that forms CO_2^*



This reaction path is shown in Figure 2-5.

Dissociation of CO_2^*



Activated CO_2 becomes stable by giving energy to third body.

In order to attain activation energies of modeled reactions, it is assumed that the activation energies of modeled reactions are enthalpy difference between H_2O and H_2O^* , and CO_2 and CO_2^* . Moreover, two modeled reactions which are $\text{H(s)} + \text{OH(s)} \rightleftharpoons \text{H}_2\text{O(s)} + \text{PT(s)}$ and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2\text{(s)} + \text{PT(s)}$ are considered in calculation. Before calculating the activation energies, it is assumed that state of activated H_2O and CO_2 are located at the barrier of surface reaction of $\text{H(s)} + \text{OH(s)} \rightleftharpoons \text{H}_2\text{O(s)} + \text{PT(s)}$ and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2\text{(s)} + \text{PT(s)}$ respectively. Figure 1-5 and 1-6 shows positions of activated H_2O and CO_2 . Plus, it is assumed that entropy of H_2O and H_2O^* is the same likewise CO_2 and CO_2^* . Enthalpy of activated H_2O and CO_2 are calculated by NASA polynomials with 7 NASA coefficients. Enthalpy difference between H_2O and CO_2 and H_2O^* and CO_2^* is hold constant in calculation. NASA Polynomials Equations¹⁵ are below.

$$\frac{H}{RT} = a_1 + a_2 \frac{T}{2} + a_3 \frac{T^2}{3} + a_4 \frac{T^3}{4} + a_5 \frac{T^4}{5} + \frac{a_6}{T} \quad (2-1)$$

$$\frac{S}{R} = a_1 \ln T + a_2 T + a_3 \frac{T^2}{2} + a_4 \frac{T^3}{3} + a_5 \frac{T^4}{4} + a_7 \quad (2-2)$$

$$\frac{C_p}{R} = a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4$$

(2-3)

where H is enthalpy that is defined as $H(T) = \Delta H_f(298K) + (H(T) - H(298K))$ in which ΔH_f is formation of enthalpy that is the heat evolved when 1 mole of the substance is formed from its elements in their respective standard state temperature of 298.15K, R is universal gas constant, T(K) is temperature, S(kJ/kmol/K) is entropy, C_p (kJ/Kmol/K) is specific heat. $a_1, a_2, a_3, a_4, a_5, a_6$ and a_7 are the numerical coefficients supplied in NASA thermodynamics files. According to the assumptions, enthalpy of H(s) and OH(s) is calculated by NASA Polynomials of Enthalpy at 700K and by adding the activation energy of the reaction $H(s) + OH(s) \rightleftharpoons H_2O(s) + PT(s)$ that is 17400 [J/mol], the state of activated H_2O is attained. As for CO_2^* , state of $CO(s) + O(s)$ is calculated by NASA Polynomials of Enthalpy at 700K, and by using the activation energy of the reaction of $CO(s) + O(s) \Rightarrow CO_2(s) + PT(s)$ that is 105000 [J/mol], the enthalpy of activated CO_2^* is attained. By comparing enthalpy of activation energies of H_2O and CO_2 with NASA coefficient of activated H_2O and CO_2 which is based on H_2O and CO_2 , NASA coefficient of H_2O^* and CO_2^* are obtained. In order to set NASA coefficients of activated H_2O and CO_2 with the same entropy, a_1, a_2, a_3, a_4 , and a_5 must be same as those of H_2O and CO_2 respectively. Therefore a_6 of activated H_2O and CO_2 is set to -2.909093622×10^4 and -3.213429830×10^4 in low temperature range which ranges from 200K to 1000K and -2.900429710×10^4 and -3.213429830×10^4 in high temperature range which ranges from 1000K to 3500K respectively based on state of H_2O and CO_2 . Table 2-2, 2-3, 2-4, and 2-5 show NASA coefficients of $H_2O, H_2O^*, CO_2, CO_2^*$.

According to the assumption, activation energies of reactions of $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2 \text{Pt(s)}$ and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2 \text{Pt(s)}$ are easily set to 17400[J/mol] and 105000[J/mol] respectively since these are located at energy barrier of $\text{H(s)} + \text{OH(s)}$ and CO(s) and O(s) . Activation energies of the reactions of $\text{OH(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + \text{O(s)} + \text{Pt(s)}$ is calculated by heat of combustion that is enthalpy difference between products and reactants. see Figure 2-4 and 2-5.

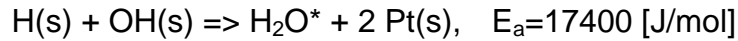
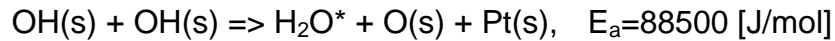


Table 2-6 shows enthalpy of species calculated by NASA polynomials of enthalpy at 700K.

Constants of A and b in AT^b called Boltzmann factor that represents collision frequency in Arrhenius law of above reactions are equal to those of the reaction of $\text{OH(s)} + \text{OH(s)} \rightleftharpoons \text{H}_2\text{O(s)} + \text{O(s)}$, $\text{H(s)} + \text{OH(s)} \rightleftharpoons \text{H}_2\text{O(s)} + \text{Pt(s)}$, and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2\text{(s)} + \text{Pt(s)}$ that is $A = 3.70 \times 10^{21}$ and $b = 0$ since reactants in three cases are the same. Summary of modeling surface reactions is shown in Table 2-7.

Activation energies of dissociation reactions are assumed that those values are zero with different collision frequency because there is no data for collision frequencies and activation energies of dissociation of H_2O^* and CO_2^* . H_2O^* is dissociated quickly into either $\text{H} + \text{OH}$ or $\text{H}_2\text{O} + \text{M}$ with different rate and CO_2^* does the same. In the modeling, it is assumed that the third body reaction, $\text{H}_2\text{O}^* + \text{M} \Rightarrow \text{H}_2\text{O} + \text{M}$, is much faster than dissociation of $\text{H}_2\text{O}^* \Rightarrow \text{H} + \text{OH}$ since $\text{H}_2\text{O}^* \Rightarrow \text{H} + \text{OH}$ is endothermic and $\text{H}_2\text{O}^* + \text{M} \Rightarrow \text{H}_2\text{O} + \text{M}$ is exothermic. The same as CO_2^* . Thus value A for first

dissociation reaction is set to 1.00×10^2 . And value A for third body reaction is 3.00×10^8 . Also, value A for $\text{CO}_2^* \Rightarrow \text{CO} + \text{O}$ is 1.00×10^2 . A for the reaction of $\text{CO}_2^* + \text{M} \Rightarrow \text{CO}_2 + \text{M}$ is set to 3.00×10^8 . Dissociation reactions are added in GRI30 mechanism¹⁰ additionally. Summary of modeled dissociation and deactivation reactions are shown in Table 2-8.

In addition, opposite case that dissociation reaction is much faster than deactivation reaction by the third body reaction is simulated. See Table 2-9.

In summary, three sub-reactions are replaced in Deutschmann mechanism^{2,11} of surface reactions and four dissociation reactions are added in GRI30 mechanism¹⁰ for gas reactions.

2.2 Simulation Method

Numerical simulation is conducted by Python with CANTERA¹⁷ that has been being developed by Dr. David G. Goodwin in Division Engineering and Applied Science in California Institution of Technology. It is being developed since 1997. It is capable of simulating thermodynamics properties, transport properties, chemical equilibrium, homogeneous and heterogeneous chemistry, reactor networks, one dimensional flames, electrochemistry, and reaction path diagrams. Catalytic combustion demo that is catcomb.py is used to simulate the modeling catalytic combustion and the conventional catalytic combustion. It is also modified to get solution for low ignition branch. GRI30 mechanism¹⁰ is used for gas reactions and Deutschmann mechanism^{2,11} is used for surface reactions. GRI30 mechanism¹⁰ consists of 325 elementary reactions and 53 species. 24 elementary surface reactions and 11 species on the surface in Deutschmann mechanism^{2,11}. Stagnation flow is set up in the catalytic combustion in CANTERA¹⁷.

Premixed Methane / air mixtures flow through configuration onto platinum surface. The length between an inlet and platinum surface is $L = 10\text{cm}$ and the area of the platinum surface is one square meter. Figure 2-6 shows the stagnation flow¹⁵. Pressure, inlet temperature, surface temperature, mass flow rate of mixtures, type of transport and composition of methane are variable. The lean mixtures are composed of 0.050 mole of methane, 0.21 mole of oxygen, 0.78 mole of nitrogen and 0.01 mole of argon as an initial condition. Reactive flow reacts on the interface of surface and gas.

Premixed methane/ air mixtures with 300K flow in axial direction at the rate of 0.06 cm/s which is equal to 0.06 kg/m²/s under atmospheric pressure, and radical effect is neglected. Surface temperature is increased by outer heat source such as electric power from 330K to 1100K and heat flux to non-reacting flow is detected.

Mass and energy balance equations⁷ are considered for steady state one dimensional flow in the simulation.

$$\frac{\partial T}{\partial t} = -u \frac{\partial T}{\partial x} + \frac{1}{\rho C_p} \frac{\partial}{\partial x} \left(\rho \lambda \frac{\partial T}{\partial x} \right) - \frac{\sum \dot{\omega}_i h_i M_{w_i}}{\rho C_p} \quad (2-4)$$

$$\frac{\partial Y_i}{\partial t} = -u \frac{\partial Y_i}{\partial x} + \frac{1}{\rho} \frac{\partial}{\partial x} \left(\rho D_i \frac{\partial Y_i}{\partial x} \right) + \frac{\dot{\omega}_i}{\rho} \quad (2-5)$$

Where $T(\text{k})$ is temperature, $u(\text{m/s})$ is axial velocity, $\rho(\text{kg/m}^3)$ is density of flow, C_p (J/mol/K) is specific heat, $\lambda(\text{W/m/K})$ is thermal conductivity, $\dot{\omega}_i$ (mol/m³) is net production rate of i th species, h_i is mass basis specific enthalpy, M_{w_i} is mole fraction of i th species, Y_i is mass fraction, and D_i (m²/s) is diffusion coefficient of i th species. Left hand side of each equation is zero because it is steady state.

Surface coverage equation is used to calculate surface coverage. Surface coverage⁸ is the ratio of the number of adsorbate atoms per unit area to the number of surface atoms per unit area generally. The equation (from Deutschmann¹) used for calculation of surface coverage is following.

$$\frac{d\theta_i}{dt} = \frac{\dot{s}_i}{\Gamma} \quad (2-6)$$

Where θ_i is coverage of ith species, t is time, \dot{s}_i is production rate of ith species on surface, and Γ is surface site density of the catalyst that is platinum with 2.7360×10^{-9} mol/cm² in this case. The equation is integrated in 1 second to get the surface coverage on platinum surface.

Production rate¹⁶ is the different between creation and destruction rate of species.

$$\dot{\omega}_i = \dot{C}_i - \dot{D}_i \quad (2-7)$$

$$\dot{C}_i = \sum_i^N (v_i'' k_{f,i} + v_i' k_{b,i}) \quad (2-8)$$

$$\dot{D}_i = \sum_i^N (v_i' k_{f,i} + v_i'' k_{b,i}) \quad (2-9)$$

Where \dot{C}_i is creation rate of ith species, \dot{D}_i is destruction rate of species, v_i' and v_i'' are stoichiometric coefficient of reactant and product of ith species respectively, $k_{f,i}$ is forward rate of reaction of ith species, and $k_{b,i}$ is reverse rate of reaction of ith species.

In this simulation, hydrogen/air mixtures are calculated first to use as the initial estimate for the methane/air mixtures by using above equations. Temperature, axial velocity, specific enthalpy, density, mole fractions of each species profiles at each grid, and surface coverage are obtained from the simulation.

2.3 Heat Transfer Calculation Method

Heat flux to non-reacting flow is obtained to see how much heat transfers from 300K to 1100K. Heat from the surface is obtained by two different ways. First approach is by energy balance⁵ in the configuration. In energy balance in the configuration, heat on the surface is heat that entering to the configuration minus summation of heat released from inlet to platinum surface to vertical direction. See Figure 2-7.

$$\dot{Q}_{in} = \dot{Q}_{loss} + \dot{Q}_{out} \quad (2-10)$$

$$\dot{Q}_{surface} = \rho_o u_o h_o - \sum_i (\rho_i u_i - \rho_{i+1} u_{i+1}) \left(\frac{h_i + h_{i+1}}{2} \right) \quad (2-11)$$

Where $\dot{Q}_{surface}$ (w/m²) is heat flux on the surface, ρ_o (kg/m³) is density of methane/air mixture at 300K before reaction occurs, u_o (m/s) is velocity of mixture entering the configuration which is same as mass flow rate that is 0.06(kg/s), and h_o (J/Kmol) is mass basis specific enthalpy of mixtures at 300K without reactions. And, i indicates point of grid in axial direction between inlet and surface. First term on the right side means heat going into the configuration and second term on the right side shows heat going out from the configuration in vertical direction. Figure 2-7 shows scheme of direction of heat.

Another approach is to calculate directly enhancement of heat transfer.

Enhancement of heat transfer is heat transfer that is obtained by product of rate of activated species leaving surface and their excess mole specific enthalpy.

$$\dot{Q}_{\text{enhancement},i} = \dot{C}_i \bar{h}_i \quad (2-12)$$

\dot{C}_i (Kmol/s/m²) is creation rate of *i*th species on the surface by modeled surface reaction that is calculated by rate of reaction, and \bar{h}_i (KJ/Kmol) is molar basis specific enthalpy. H₂O* is formed by reactions OH(s) + OH(s) => H₂O* + O(s) + Pt(s) with 88500[J/mol] of activation energy and H(s) + OH(s) => H₂O* + 2 Pt(s) with 17400[J/mol] of activation energy. CO₂* is formed by reaction of CO(s) + O(s) => CO₂* + 2 PT(s) with 105000 [J/mol] of activation energy. Therefore, \bar{h}_i is 88500[J/mol], 17400[J/mol], and 105000[J/mol] respectively in this case. Since \bar{h}_i is enthalpy difference between activated species such as H₂O* and CO₂* and stable species such as H₂O and CO₂, this equation shows excess energy from each modeled surface reaction when compared to conventional mechanism.

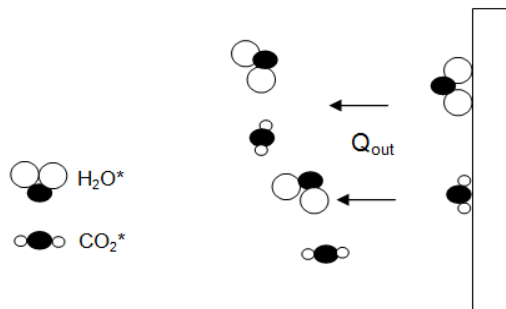


Figure 2-8. Enhancement of heat transfer

2.4 Fuel Consumption

Fuel consumption is calculated with increasing surface temperature to see temperature where fuel is all burnt. New mechanism shows different fuel consumption at each temperature. Detail is explained in Chapter 3. It is calculated by product of ratio of mass fraction of fuel at the surface to initial mass fraction of fuel at $z=0\text{m}$ and $z=0.1\text{m}$. Equation is following.

$$\frac{\text{Initial Mass Fraction of Fuel} - \text{Mass Fraction of Fuel On The Surface}}{\text{Initial Mass Fraction of Fuel}} \times 100 \quad (2-13)$$

Briefly, modeling, simulation, heat transfer, and fuel consumption calculation method are explained in this chapter. New mechanism is modeled with basis on Deutschmann mechanism^{2,11} for surface reactions and GRI30 mechanism¹⁰ for gas reactions with replacing and adding elementary reactions. And solution is obtained by mass, energy balance and surface coverage equations in simulation. In addition, By using solutions at each temperature, heat transfer on the right side of surface where there is no reaction, enhanced heat transfer that is additional heat from surface reactions compared to conventional catalytic combustion, and fuel consumption are calculated to see the difference between conventional mechanism and new mechanism. The results are explained in Chapter 3.

Table 2-1. Detailed surface mechanism of Deutschmann

Deutschmann mechanism ^{2,11} of surface		A	b	E _a
Reactions		(cm mol s)		(J/mol)
(1)	$\text{H}_2 + 2 \text{ Pt(s)} \Rightarrow 2 \text{ H(s)}$	4.45×10^{10}	0.5	0
(2)	$2 \text{ H(s)} \Rightarrow \text{H}_2 + 2 \text{ Pt(s)}$	3.70×10^{21}	0	67400
(3)	$\text{H} + \text{ Pt(s)} \Rightarrow \text{ H(s)}$	1.00	0	0
(4)	$\text{O}_2 + 2 \text{ Pt(s)} \Rightarrow 2 \text{ O(s)}$	1.80×10^{21}	-0.5	0
(5)	$\text{O}_2 + 2 \text{ Pt(s)} \Rightarrow 2 \text{ O(s)}$	2.30×10^{-2}	0	0
(6)	$2 \text{ O(s)} \Rightarrow \text{O}_2 + 2 \text{ Pt(s)}$	3.70×10^{21}	0	213200
(7)	$\text{O} + \text{ Pt(s)} \Rightarrow \text{ O(s)}$	1.0	0	0
(8)	$\text{H}_2\text{O} + \text{ Pt(s)} \Rightarrow \text{ H}_2\text{O(s)}$	7.50×10^{-1}	0	0
(9)	$\text{H}_2\text{O(s)} \Rightarrow \text{ H}_2\text{O} + \text{ Pt(s)}$	1.00×10^{13}	0	40300
(10)	$\text{OH} + \text{ Pt(s)} \Rightarrow \text{ OH(s)}$	1.0	0	0
(11)	$\text{OH(s)} \Rightarrow \text{ OH} + \text{ Pt(s)}$	1.0×10^{13}	0	192800
(12)	$\text{H(s)} + \text{ O(s)} \rightleftharpoons \text{ OH(s)} + \text{ Pt(s)}$	3.70×10^{21}	0	11500
(13)	$\text{H(s)} + \text{ OH(s)} \rightleftharpoons \text{ H}_2\text{O(s)} + \text{ Pt(s)}$	3.70×10^{21}	0.5	17400
(14)	$\text{OH(s)} + \text{ OH(s)} \rightleftharpoons \text{ H}_2\text{O(s)} + \text{ O(s)}$	3.70×10^{21}	0	48200
(15)	$\text{CO} + \text{ Pt(s)} \Rightarrow \text{ CO(s)}$	1.618×10^{20}	0.5	0
(16)	$\text{CO(s)} \Rightarrow \text{ CO} + \text{ Pt(s)}$	1.00×10^{13}	0	125500
(17)	$\text{CO}_2\text{(s)} \Rightarrow \text{ CO}_2 + \text{ Pt(s)}$	1.00×10^{13}	0	20500
(18)	$\text{CO(s)} + \text{ O(s)} \Rightarrow \text{ CO}_2\text{(s)} + \text{ Pt(s)}$	3.70×10^{21}	0	105000
(19)	$\text{CH}_4 + 2 \text{ Pt(s)} \Rightarrow \text{ CH}_3\text{(s)} + \text{ H(s)}$	4.6334×10^{20}	0.5	0
(20)	$\text{CH}_3\text{(s)} + \text{ Pt(s)} \Rightarrow \text{ CH}_2\text{(s)} + \text{ H(s)}$	3.70×10^{21}	0	20000
(21)	$\text{CH}_2\text{(s)} + \text{ Pt(s)} \Rightarrow \text{ CH(s)} + \text{ H(s)}$	3.70×10^{21}	0	20000
(22)	$\text{CH(s)} + \text{ Pt(s)} \Rightarrow \text{ C(s)} + \text{ H(s)}$	3.70×10^{21}	0	20000
(23)	$\text{C(s)} + \text{ O(s)} \Rightarrow \text{ CO(s)} + \text{ Pt(s)}$	3.70×10^{21}	0	62800
(24)	$\text{CO(s)} + \text{ Pt(s)} \Rightarrow \text{ C(s)} + \text{ O(s)}$	1.00×10^{18}	0	184000

Table 2-2. NASA coefficients of H₂O

NASA Coefficients of H ₂ O	Low Temperature Range 200K - 1000K	High Temperature Range 1000K – 3500K
a ₁	4.198640560×10^0	3.033992490×10^0
a ₂	$-2.036434100 \times 10^{-3}$	$2.176918040 \times 10^{-3}$
a ₃	$6.520402110 \times 10^{-6}$	$-1.640725180 \times 10^{-7}$
a ₄	$-5.487970620 \times 10^{-9}$	$-9.704198700 \times 10^{-11}$
a ₅	$1.771978170 \times 10^{-12}$	$1.682009920 \times 10^{-14}$
a ₆	$-3.029372670 \times 10^{-4}$	$-3.000429710 \times 10^{-4}$
a ₇	$-8.490322080 \times 10^{-1}$	4.966770100×10^0

Table 2-3. NASA coefficients of H₂O*

NASA Coefficients of H ₂ O*	Low Temperature Range 200K - 1000K	High Temperature Range 1000K – 3500K
a ₁	4.198640560 x 10 ⁰	3.033992490 x 10 ⁰
a ₂	-2.036434100 x 10 ⁻³	2.176918040 x 10 ⁻³ ,
a ₃	6.520402110 x 10 ⁻⁶	-1.640725180 x 10 ⁻⁷
a ₄	-5.487970620 x 10 ⁻⁹	-9.704198700 x 10 ⁻¹¹
a ₅	1.771978170 x 10 ⁻¹²	1.682009920 x 10 ⁻¹⁴
a ₆	-2.909093622 x 10 ⁺⁴	-2.900429710 x 10 ⁺⁴
a ₇	-8.490322080 x 10 ⁻¹	4.966770100 x 10 ⁰

Table 2-4. NASA coefficients of CO₂

NASA Coefficients of CO ₂	Low Temperature Range 200K - 1000K	High Temperature Range 1000K – 3500K
a ₁	2.356773520 x 10 ⁰	3.857460290 x 10 ⁰
a ₂	8.984596770 x 10 ⁻³	4.414370260 x 10 ⁻³
a ₃	-7.123562690 x 10 ⁻⁶	-2.214814040 x 10 ⁻⁶
a ₄	2.459190220 x 10 ⁻⁹	5.234901880 x 10 ⁻¹⁰
a ₅	-1.436995480 x 10 ⁻¹³	-4.720841640 x 10 ⁻¹⁴
a ₆	-4.837196970 x 10 ⁺⁴	-4.875916600 x 10 ⁻¹⁴
a ₇	9.901052220 x 10 ⁰	2.271638060 x 10 ⁰

Table 2-5. NASA coefficients of CO₂*

NASA Coefficients of CO ₂ *	Low Temperature Range 200K - 1000K	High Temperature Range 1000K – 3500K
a ₁	2.356773520 x 10 ⁰	3.857460290 x 10 ⁰
a ₂	8.984596770 x 10 ⁻³	4.414370260 x 10 ⁻³
a ₃	-7.123562690 x 10 ⁻⁶	-2.214814040 x 10 ⁻⁶
a ₄	2.459190220 x 10 ⁻⁹	5.234901880 x 10 ⁻¹⁰
a ₅	-1.436995480 x 10 ⁻¹³	-4.720841640 x 10 ⁻¹⁴
a ₆	-3.213429830 x 10 ⁺⁴	-3.213429830 x 10 ⁻¹⁴
a ₇	9.901052220 x 10 ⁰	2.271638060 x 10 ⁰

Table 2-6. Enthalpy of species at 700K. H is calculated by (H/RT)*RT where R is gas constant that is 8.314[J/mol/K] and T is temperature.

Species	H/RT	H [J/mol]
H(s)	-5.61807343242547	-32696.0638
OH(s)	-35.1834824436407	-204760.8311
CO(s)	-41.1300311398638	-239368.5552
O(s)	-41.1300311398638	-103372.7744
Pt(s)	0	0
H ₂ O*	-37.3930957669273	-217620.3387
H ₂ O	-39.111367881213	-227620.3388
CO ₂ *	-41.364298169368	-240731.9425
CO ₂	-64.5609715979394	-375731.9425

Table 2-7. Surface reactions in new mechanism

	Surface Reaction Mechanism Of Modeling	A (cm mol s)	b	E _a (J/mol)
(1)	$\text{H}_2 + 2 \text{Pt(s)} \Rightarrow 2 \text{H(s)}$	4.45×10^{10}	0.5	0
(2)	$2 \text{H(s)} \Rightarrow \text{H}_2 + 2 \text{Pt(s)}$	3.70×10^{21}	0	67400
(3)	$\text{H} + \text{Pt(s)} \Rightarrow \text{H(s)}$	1.00	0	0
(4)	$\text{O}_2 + 2 \text{Pt(s)} \Rightarrow 2 \text{O(s)}$	1.80×10^{21}	-0.5	0
(5)	$\text{O}_2 + 2 \text{Pt(s)} \Rightarrow 2 \text{O(s)}$	2.30×10^{-2}	0	0
(6)	$2 \text{O(s)} \Rightarrow \text{O}_2 + 2 \text{PT(s)}$	3.70×10^{21}	0	213200
(7)	$\text{O} + \text{Pt(s)} \Rightarrow \text{O(s)}$	1.0	0	0
(8)	$\text{H}_2\text{O} + \text{Pt(s)} \Rightarrow \text{H}_2\text{O(s)}$	7.50×10^{-1}	0	0
(9)	$\text{H}_2\text{O(s)} \Rightarrow \text{H}_2\text{O} + \text{Pt(s)}$	1.00×10^{13}	0	40300
(10)	$\text{OH} + \text{Pt(s)} \Rightarrow \text{OH(s)}$	1.0	0	0
(11)	$\text{OH(s)} \Rightarrow \text{OH} + \text{Pt(s)}$	1.0×10^{13}	0	192800
(12)	$\text{H(s)} + \text{O(s)} \rightleftharpoons \text{OH(s)} + \text{Pt(s)}$	3.70×10^{21}	0	11500
(13)	$\text{CO} + \text{Pt(s)} \Rightarrow \text{CO(s)}$	1.618×10^{20}	0.5	0
(14)	$\text{CO(s)} \Rightarrow \text{CO} + \text{Pt(s)}$	1.00×10^{13}	0	125500
(15)	$\text{CO}_2\text{(s)} \Rightarrow \text{CO}_2 + \text{Pt(s)}$	1.00×10^{13}	0	20500
(16)	$\text{CH}_4 + 2 \text{Pt(s)} \Rightarrow \text{CH}_3\text{(s)} + \text{H(s)}$	4.6334×10^{20}	0.5	0
(17)	$\text{CH}_3\text{(s)} + \text{Pt(s)} \Rightarrow \text{CH}_2\text{(s)s} + \text{H(s)}$	3.70×10^{21}	0	20000
(18)	$\text{CH}_2\text{(s)s} + \text{Pt(s)} \Rightarrow \text{CH(s)} + \text{H(s)}$	3.70×10^{21}	0	20000
(19)	$\text{CH(s)} + \text{Pt(s)} \Rightarrow \text{C(s)} + \text{H(s)}$	3.70×10^{21}	0	20000
(20)	$\text{C(s)} + \text{O(s)} \Rightarrow \text{CO(s)} + \text{Pt(s)}$	3.70×10^{21}	0	62800
(21)	$\text{CO(s)} + \text{Pt(s)} \Rightarrow \text{C(s)} + \text{O(s)}$	1.00×10^{18}	0	184000
(22)	$\text{OH(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + \text{O(s)} + \text{Pt(s)}$	3.70×10^{21}	0	88500
(23)	$\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2 \text{PT(s)}$	3.70×10^{21}	0	17400
(24)	$\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2 \text{PT(s)}$	3.70×10^{21}	0	105000

Table 2-8. Gas reactions in new mechanism with faster deactivation reaction

	Dissociation Reactions in Modeling	A (cm mol s)	b	E _a (J/mol)
(1)	$\text{H}_2\text{O}^* \Rightarrow \text{H} + \text{OH}$	1.00×10^2	0	0
(2)	$\text{H}_2\text{O}^* + \text{M} \Rightarrow \text{H}_2\text{O} + \text{M}$	3.00×10^8	0	0
(3)	$\text{CO}_2^* \Rightarrow \text{CO} + \text{O}$	1.00×10^2	0	0
(4)	$\text{CO}_2^* + \text{M} \Rightarrow \text{CO}_2 + \text{M}$	3.00×10^8	0	0

Table 2-9. Gas reactions in new mechanism with the faster dissociation reaction

Dissociation Reactions in Modeling		A (cm mol s)	b	E _a (J/mol)
(1)	$\text{H}_2\text{O}^* \Rightarrow \text{H} + \text{OH}$	3.00×10^3	0	0
(2)	$\text{H}_2\text{O}^* + \text{M} \Rightarrow \text{H}_2\text{O} + \text{M}$	0.10×10^1	0	0
(3)	$\text{CO}_2^* \Rightarrow \text{CO} + \text{O}$	3.00×10^3	0	0
(4)	$\text{CO}_2^* + \text{M} \Rightarrow \text{CO}_2 + \text{M}$	0.10×10^1	0	0

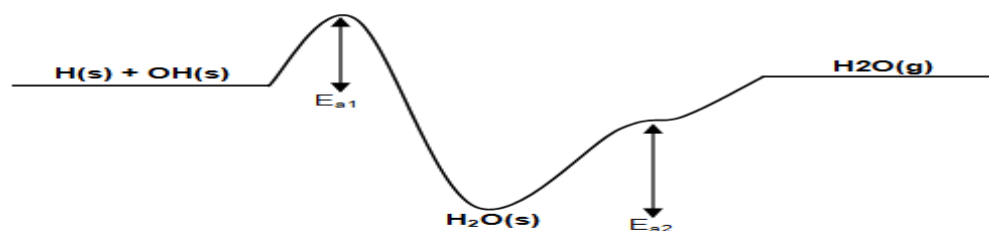


Figure 2-1. Reaction path of $\text{H(s)} + \text{OH(s)}$ in Deutschmann mechanism. E_{a1} indicates activation energy of reaction of $\text{H(s)} + \text{OH(s)}$ and E_{a2} indicates activation energy of desorption of reaction of $\text{H}_2\text{O(s)}$. ($E_{a1}=17400\text{J/mol}$, and $E_{a2}=40300\text{J/mol}$)

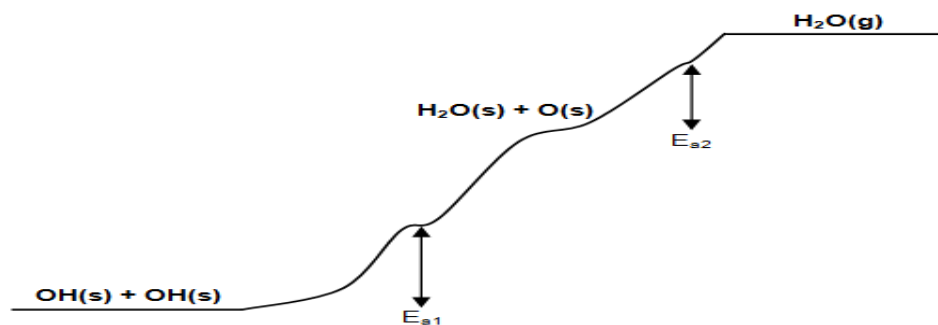


Figure 2-2. Reaction path of $\text{OH(s)} + \text{OH(s)}$ and $\text{H}_2\text{O(s)}$ in Deutschmann mechanism. E_{a1} is 48200J/mol and E_{a2} is 40300J/mol .

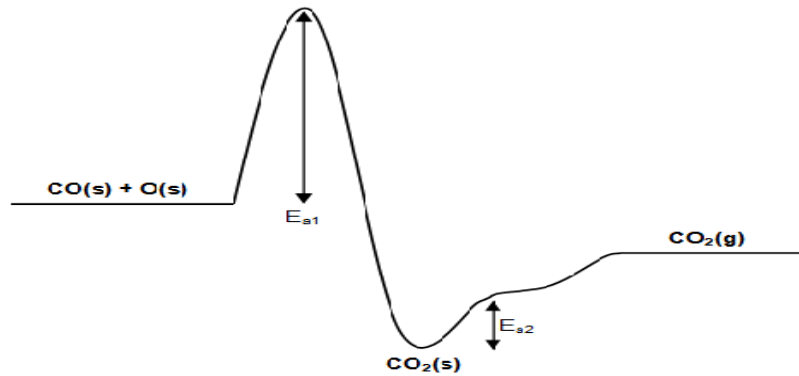


Figure 2-3. Reaction path of $\text{CO(s)} + \text{O(s)}$ and $\text{CO}_2\text{(s)}$ in Deutschmann mechanism. E_{a1} is 105000J/mol and E_{a2} is 20500J/mol.

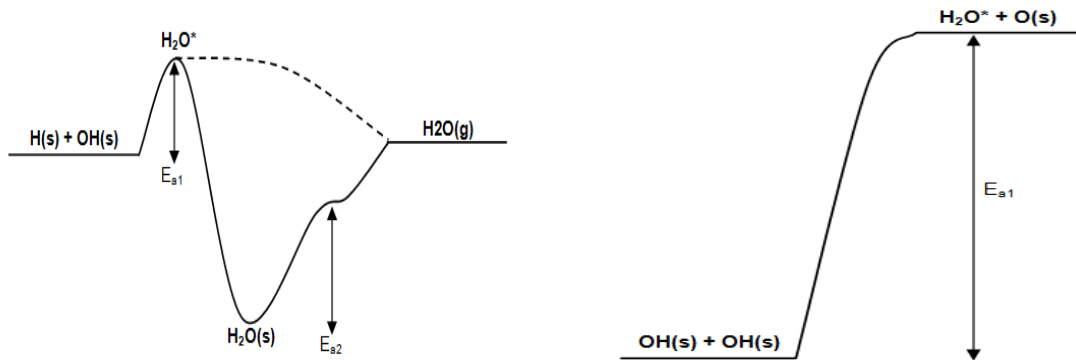


Figure 2-4. Left picture shows reaction path of reaction of $\text{H(s)} + \text{O(s)}$ in modeling. E_{a1} indicates activation energy of reaction $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}$. Dashed line show reaction path of modeling. Line is conventional reaction path. Right figure shows reaction path of reaction of $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2\text{pt(s)}$. E_{a1} indicates activation energy of above reaction.

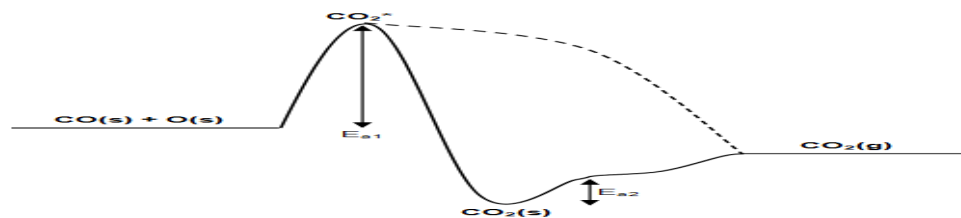


Figure 2-5. Reaction path of $\text{CO(s)} + \text{O(s)}$ in modeling. Dashed line shows reaction path of modeling. E_{a1} is activation energy of reaction of $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2\text{Pt(s)}$, and the line shows conventional reaction path.

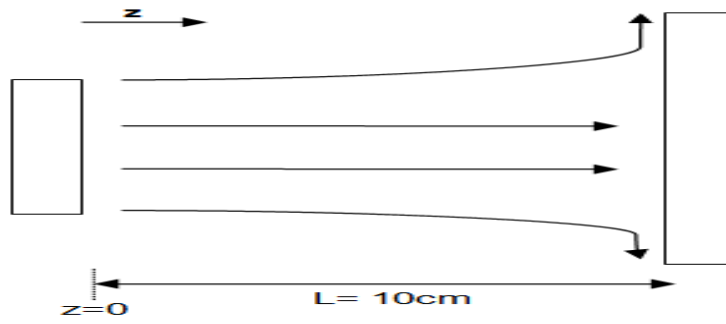


Figure 2-6. Stagnation flow

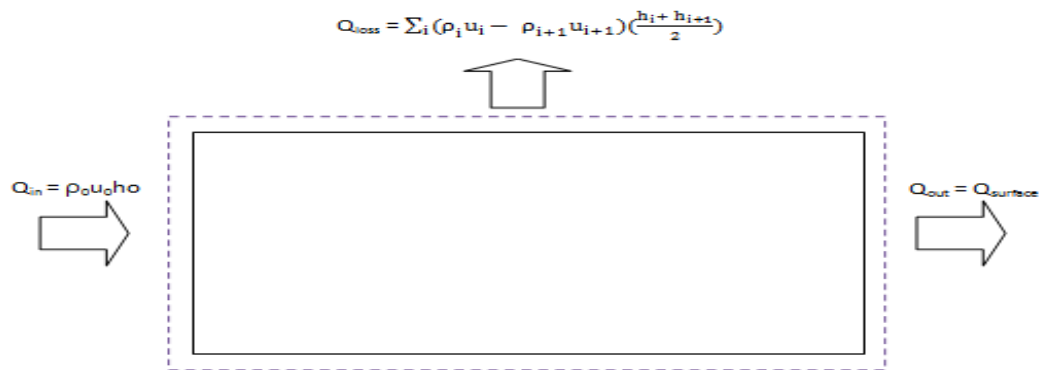


Figure 2-7. Energy balance

CHAPTER 3 RESULTS AND DISCUSSION

The methods of modeling, simulation, heat transfer, enhancement of heat transfer and fuel consumption are explained in Chapter 2. In Chapter 3, heat transfer and enhancement of heat transfer and fuel consumption and behavior of new mechanism will be explained. In short, new mechanism shows different fuel consumption at each temperature, mole fraction, heat transfer, and behavior.

3.1 Analysis of Fuel Consumption and Mole Fraction

Simulation results show that fuel consumption and mole fraction of new mechanism are different from those of conventional catalytic methane mechanism at certain temperature. It is because new mechanism considers cooling heat transfer to non-reacting flow. And H_2O^* and CO_2^* are formed in new mechanism. Moreover, in the new mechanism more fuel is consumed at lower temperature than conventional mechanism.

3.1.1 New Mechanism with Faster Deactivation Reaction

In case of conventional catalytic methane, fuel consumption is 55.66% at 900K whereas 63.59% at 900K in new mechanism of catalytic methane with faster deactivation reactions. Therefore, Fuel is consumed well in new mechanism under the same conditions such as initial mixture composition, pressure and gas inlet temperature. Thereby, much more heat transfers to non-reacting flow at the same temperature.

It is shown that mole fraction is different in new mechanism. Mole fraction of H_2O^* and CO_2^* are added, and CH_4 is consumed quickly and much more H_2O and CO_2 are formed as products.

3.1.2 New Mechanism with Faster Dissociation Reaction

In case of new mechanism with faster dissociation reaction, fuel consumption is 99.35% at 900K whereas all of methane consumed around 1100K in conventional case. This case shows the most fuel consumption among the mechanisms. Therefore, the most fuel is consumed in new mechanism with faster dissociation reactions under the same conditions such as initial mixture composition, pressure and gas inlet temperature. Thereby, the most heat transfers to non-reacting flow at the same temperature.

Compared to those two mechanisms, CH_4 is consumed more quickly and the most H_2O and CO_2 are formed as products at the same temperature.

3.2 Analysis of Heat Transfer

Heat flux to non-reacting flow is calculated by global energy balance equation⁵ by increasing surface temperature from 300K to 1100K.

3.2.1 New Mechanism with Faster Deactivation Reaction

In conventional catalytic methane, it shows that surface reactions start to occur at 760K since heat transfer to non-reacting flow starts to increase with increasing surface temperature while some surface reactions start at 750K in new mechanism. Conventional catalytic methane shows that ignition occurs around 930K because since 930K heat is released from reactions whereas new mechanism of catalytic methane shows that ignition occurs around 913K. Result of heat transfer to non-reacting flow shows that in conventional catalytic methane with respect to heat transfer, heat should be added by about 930K to make ignition occur, when surface temperature is over 930K, heat is started to released to non-reacting flow. The Figure 3-16 shows heat flux to the non-reacting flow.

In new mechanism, the shape of graph is similar to that in conventional catalytic methane. Detailed heat flux is shown in Figure 3-17. However, heat transfer to non-reacting flow is slightly larger than that in conventional catalytic mechanism. It is because there is excess heat transfer from adsorption and desorption of species such as H_2O^* and CO_2^* . When compared to conventional catalytic mechanism, it shows that heat transfer to non-reacting flow is similar to that of conventional catalytic mechanism by 730K. This means that surface reactions that form H_2O^* and CO_2^* are inactive.

However, slight difference of heat transfer to non-reacting flow starts to show since 730K and it is larger as surface temperature is increased since surface reactions of $\text{OH(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + \text{O(s)} + \text{Pt(s)}$, $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2 \text{Pt(s)}$, and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2 \text{Pt(s)}$ start to occurs around 730K. After 1070K, heat transfer to non-reacting flow is suddenly increased as surface temperature is increased since at high temperature, gas reactions are dominant. See Figure 3-17 and Figure 3-18.

Excess heat from the surface reaction that forms H_2O^* and CO_2^* is almost zero by 700K. After 700K excess heat from surface reactions starts to generate. Then, excess heat is suddenly increases as surface temperature is increased. Therefore, enhancement of heat transfer due to surface reactions results in slight difference of heat transfer to non-reacting flow between new mechanism and conventional catalytic mechanism. This means that much heat can be obtained from the reactions of $\text{OH(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + \text{O(s)} + \text{Pt(s)}$, $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2 \text{Pt(s)}$, and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2 \text{Pt(s)}$. At 990K enhancement of heat transfer to non-reacting flow starts to decrease and after 1070K it is suddenly drop with increasing surface temperature since at high temperature gas reactions are dominant. See Figure 3-19.

3.2.2 New Mechanism with Faster Dissociation Reaction

Unlike heat transfer of conventional catalytic methane, new mechanism with faster dissociation reaction consists of high and low ignition branch. Detailed heat flux is shown in Figure 3-20. Heat transfer to non-reacting flow is calculated by increasing surface temperature from 300K to 1100K.

In the low ignition branch, surface reaction is dominant but weak. Therefore, radicals such as H, O, OH, H₂O, CO₂ come off from the surface. As surface temperature increases from 300K, more heat is needed to light off in low ignition branch. Heat transfer between 918K and 936K shows unusual phenomenon. Much more heat is needed to maintain surface temperature. This has not been discovered in this study. More experimental research is needed to find the reason. At 1015.2K, small amount of heat is needed. This means that ignition occurs at this temperature, and reactions suddenly jump to the high ignition branch. In high branch, surface temperature should be cooled down to maintain combustion. More decreasing surface temperature leads to extinguishing flame at 616.5K with jumping to low branch. The reason why the amount of heat released from surface is suddenly increased from 1075K to 1080K is that even though gas reaction is dominant at this temperature, there is some of surface reaction, but at 1080K, there is no surface reaction at all. By repeating increasing and decreasing surface temperature, it shows auto-thermal phenomenon. Combustion circulates from 1015.2K to 616.2K in the high branch and from 616.2K to 1015.2K in the low branch. When compared to heat transfer of conventional catalytic methane, it shows that much more complete combustion from 616.6K to 1015.2K after ignition occurs.

Excess heat from the surface reaction that forms H₂O* and CO₂* is almost zero in low branch whereas in high branch much heat is released from the surface. This means

that much heat can be obtained from the reactions of $\text{OH(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + \text{O(s)} + \text{Pt(s)}$, $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2 \text{Pt(s)}$, and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2 \text{Pt(s)}$. Figure 3-21 shows enhancement of heat transfer.

Heat flux to non-reacting flow by global energy balance is different from that by enhancement of heat transfer. It shows that enhancement of heat flux is three times larger than heat transfer by global energy balance. Thus, according to this mechanism, more heat is obtained from surface reactions compared to conventional mechanism. Finally, more experimental works are required to get collision frequency and activation energy of modeled dissociation reactions and to correct new mechanism.

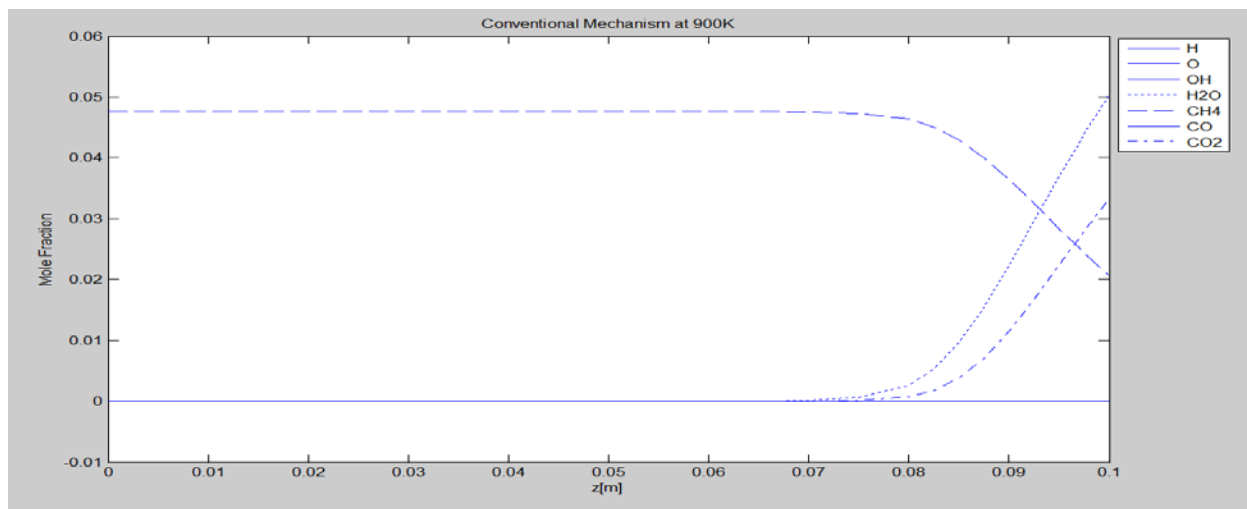


Figure 3-1. Mole fraction of species in Deutschmann at 900K

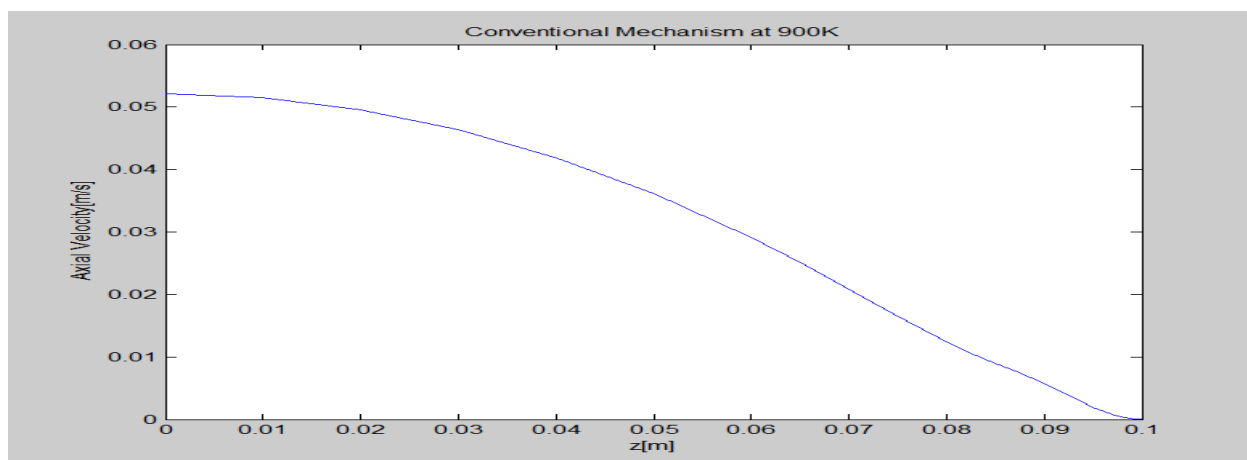


Figure 3-2. Axial velocity profile in Deutschmann at 900K

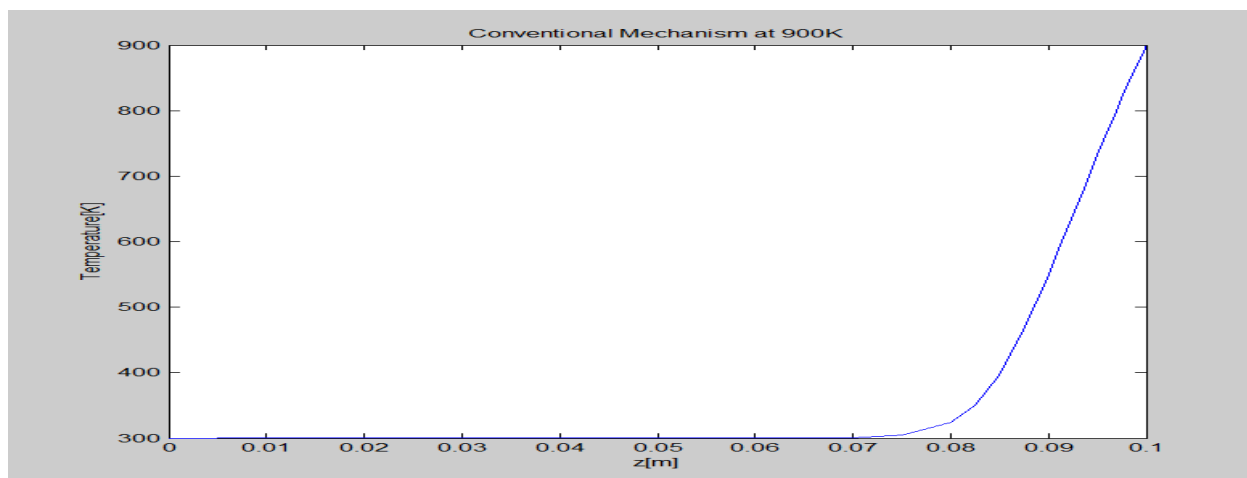


Figure 3-3. Temperature profile in Deutschmann at 900K

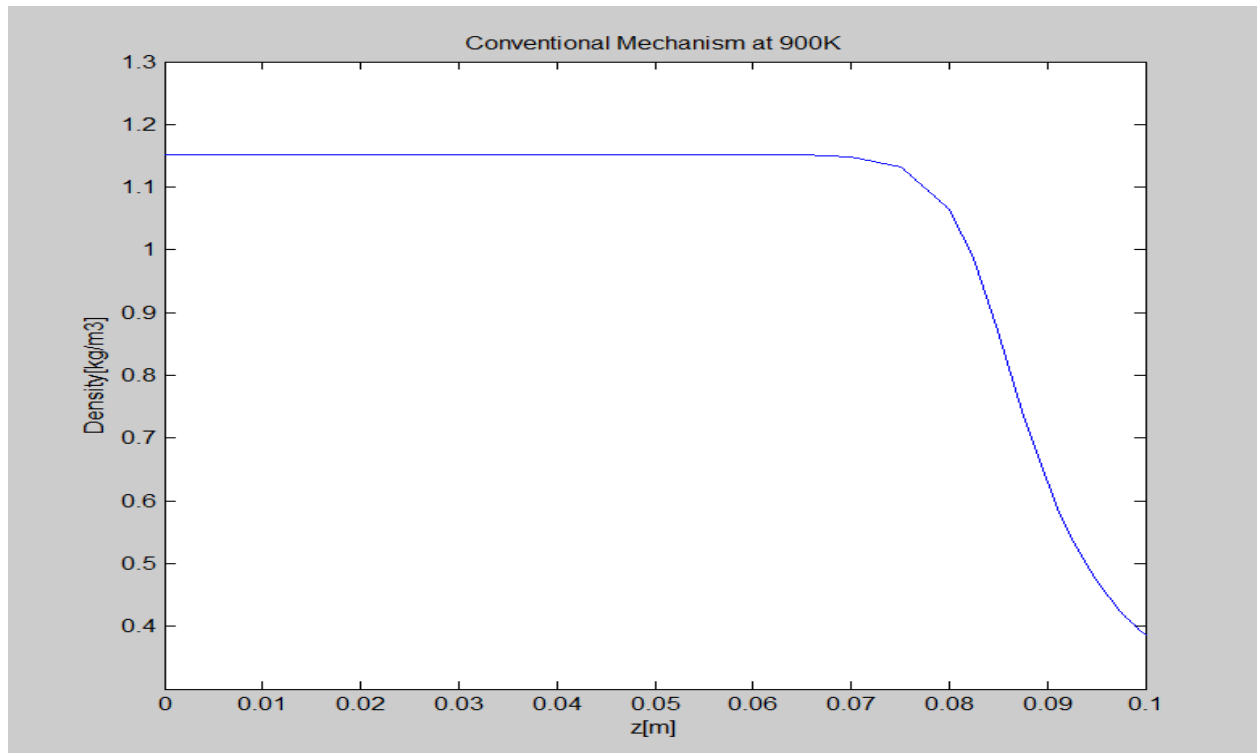


Figure 3-4. Density profile in Deutschmann at 900K

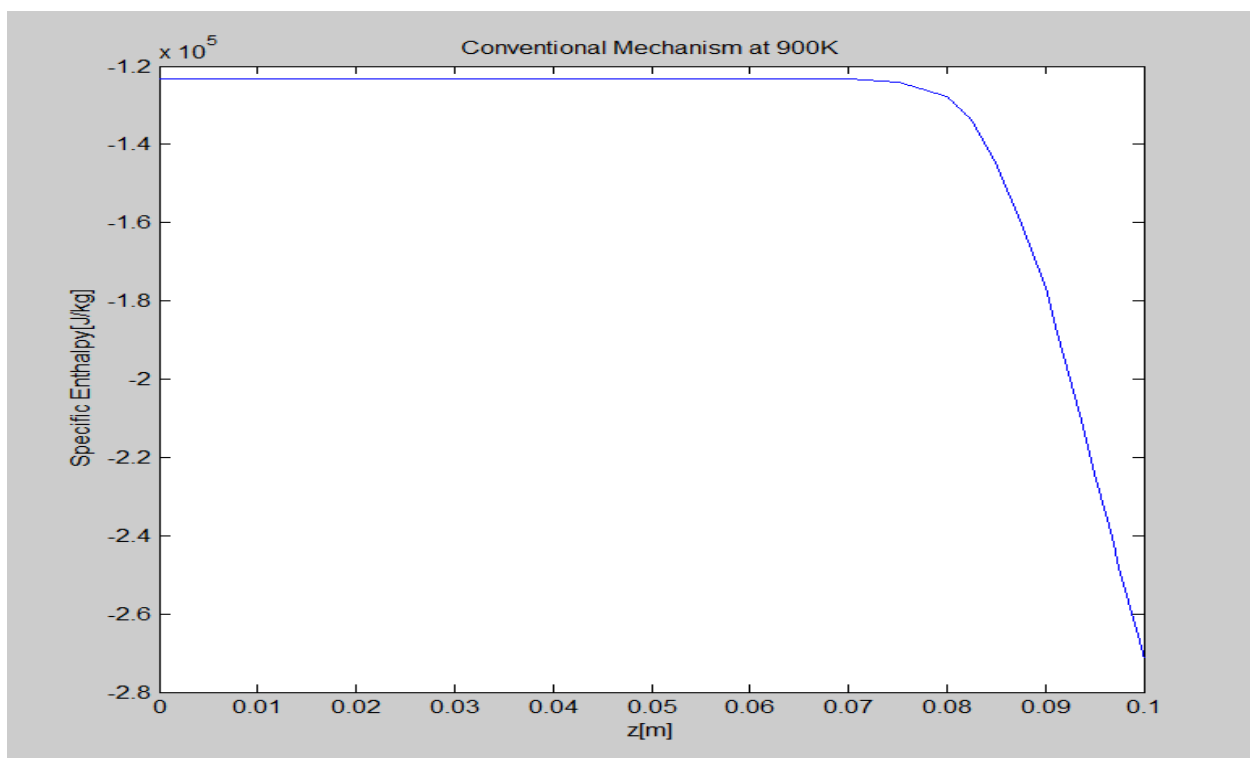


Figure 3-5. Specific enthalpy in Deutschmann at 900K

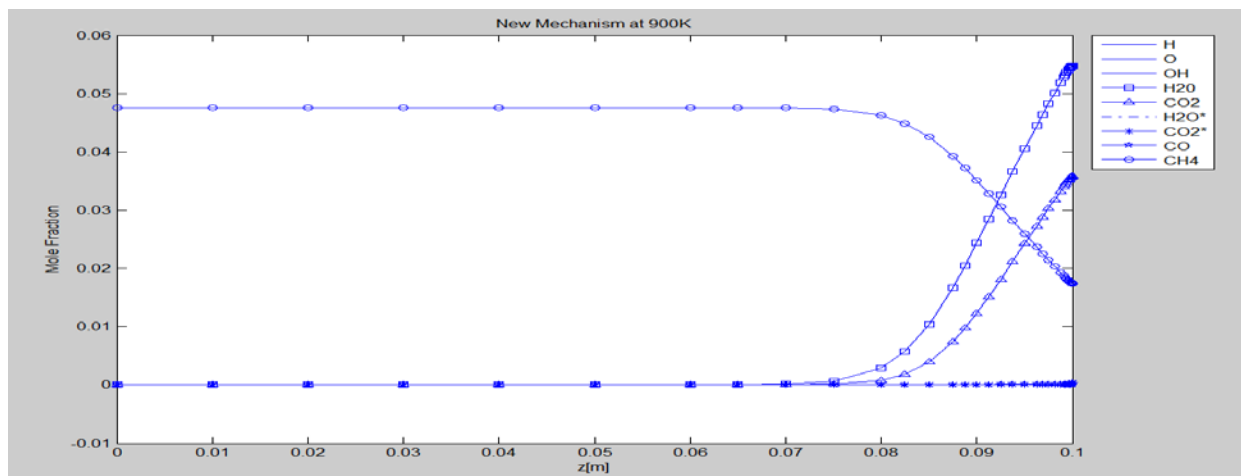


Figure 3-6. Mole fraction of species in new mechanism with faster deactivation at 900K

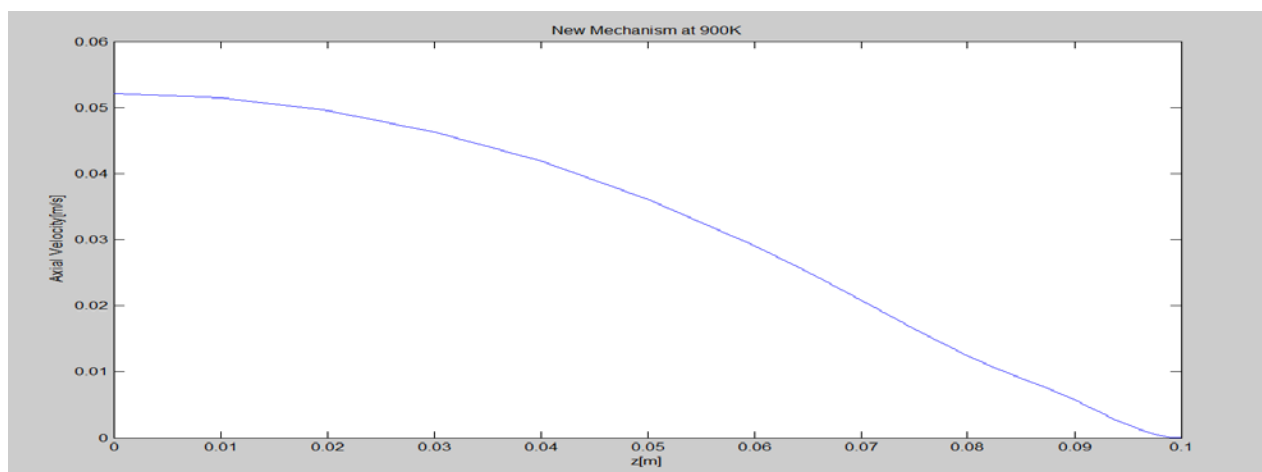


Figure 3-7. Axial velocity profile in new mechanism with faster deactivation at 900K

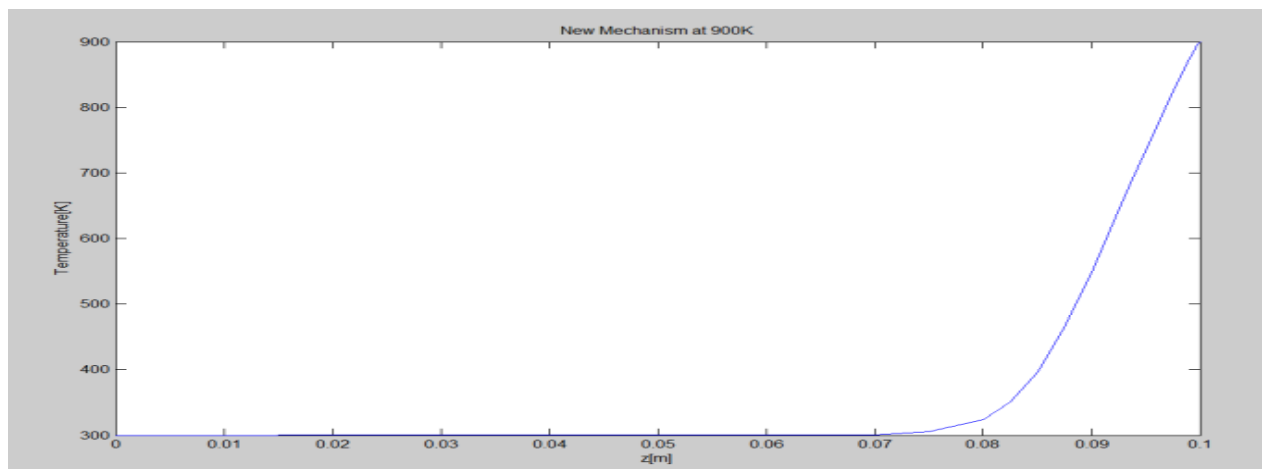


Figure 3-8. Temperature profile in new mechanism with faster deactivation at 900K

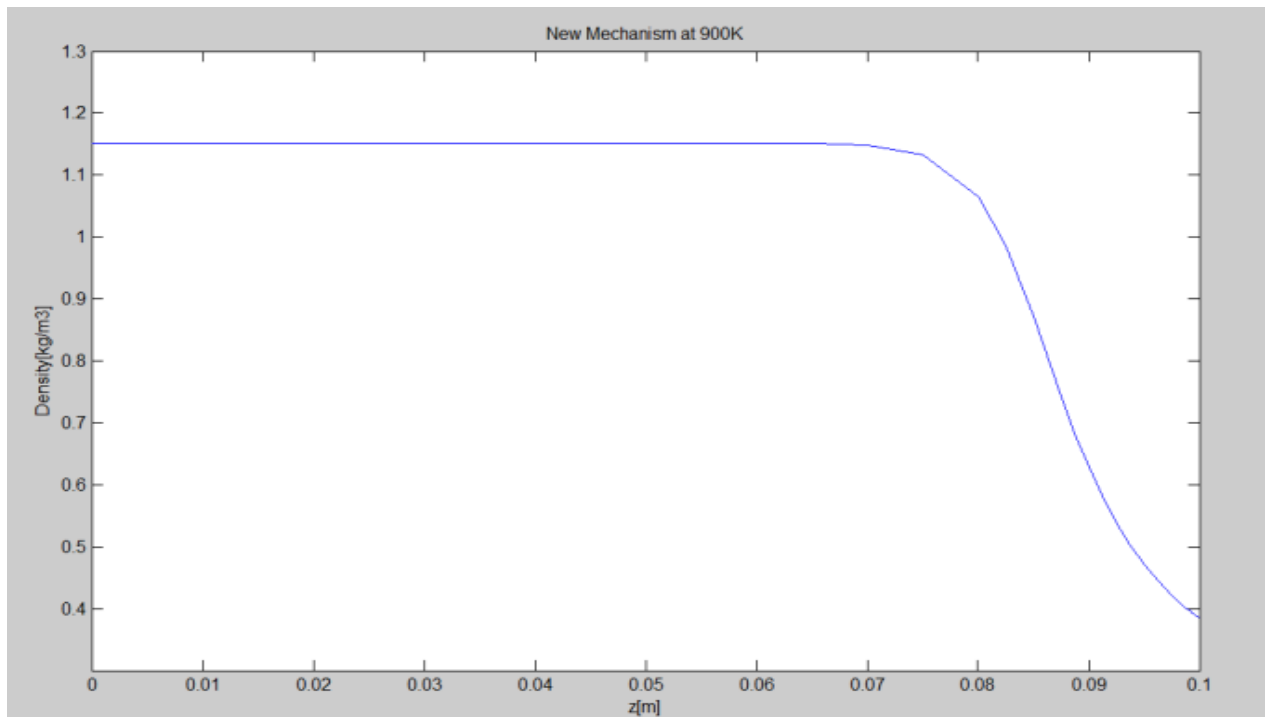


Figure 3-9. Density profile in new mechanism with faster deactivation at 900K

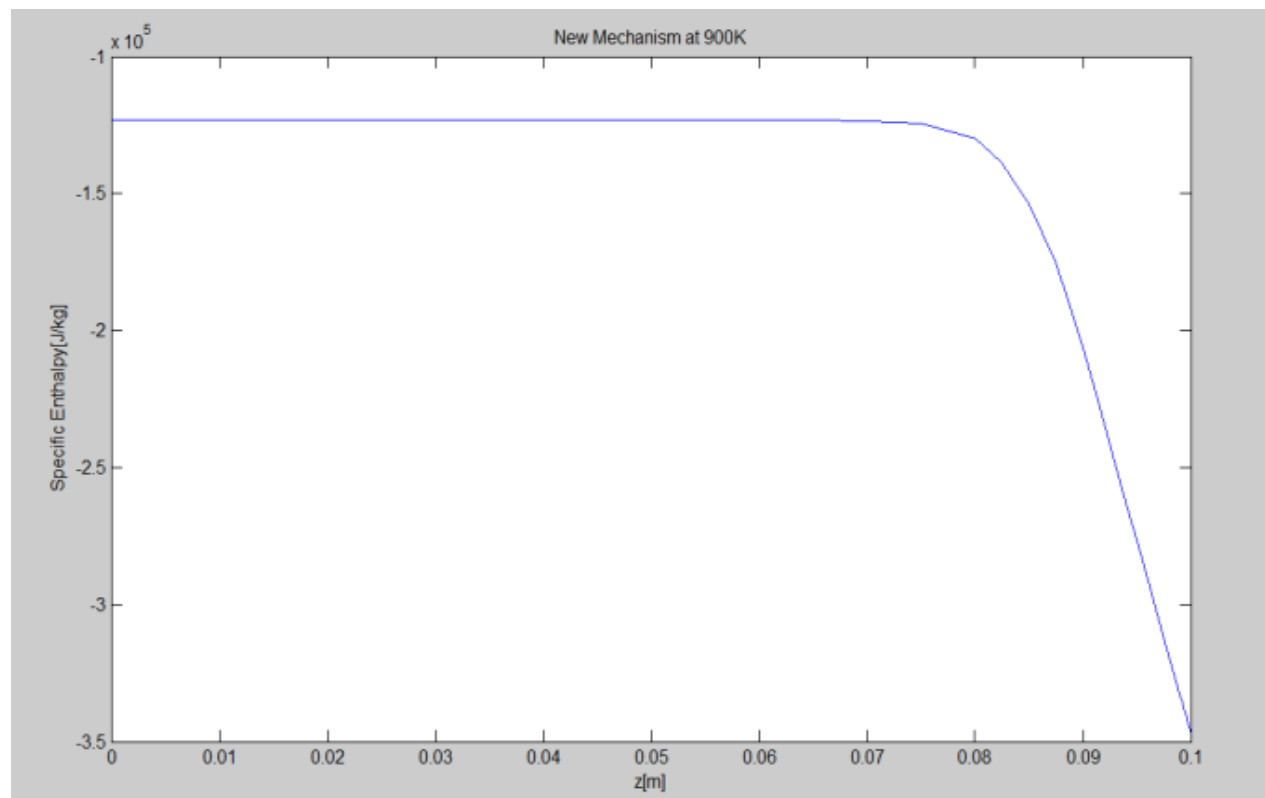


Figure 3-10. Specific enthalpy in new mechanism with faster deactivation at 900K

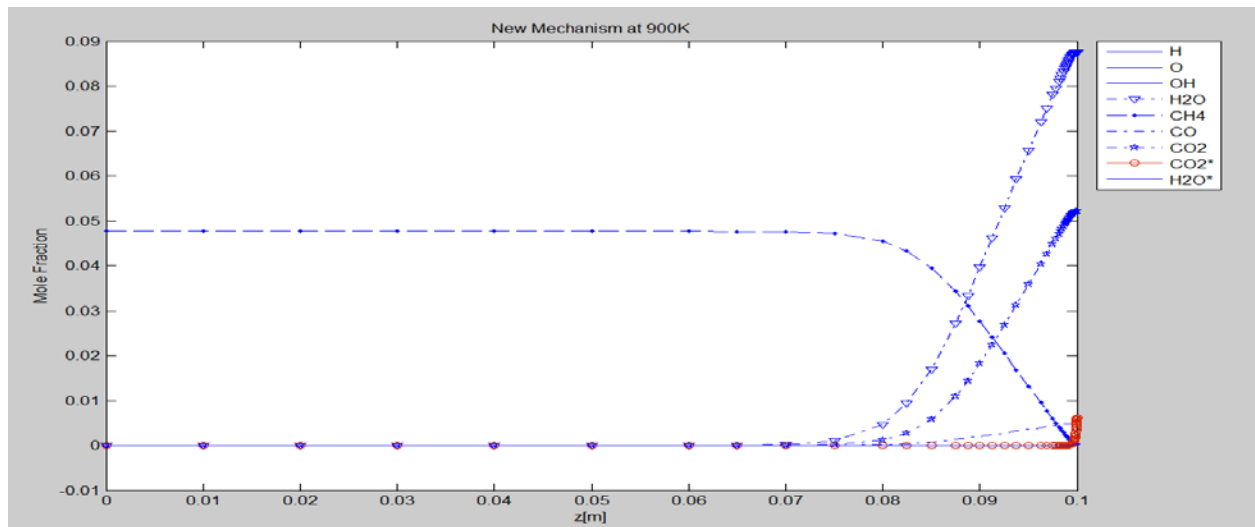


Figure 3-11. Mole fraction of species in new mechanism with faster dissociation at 900K

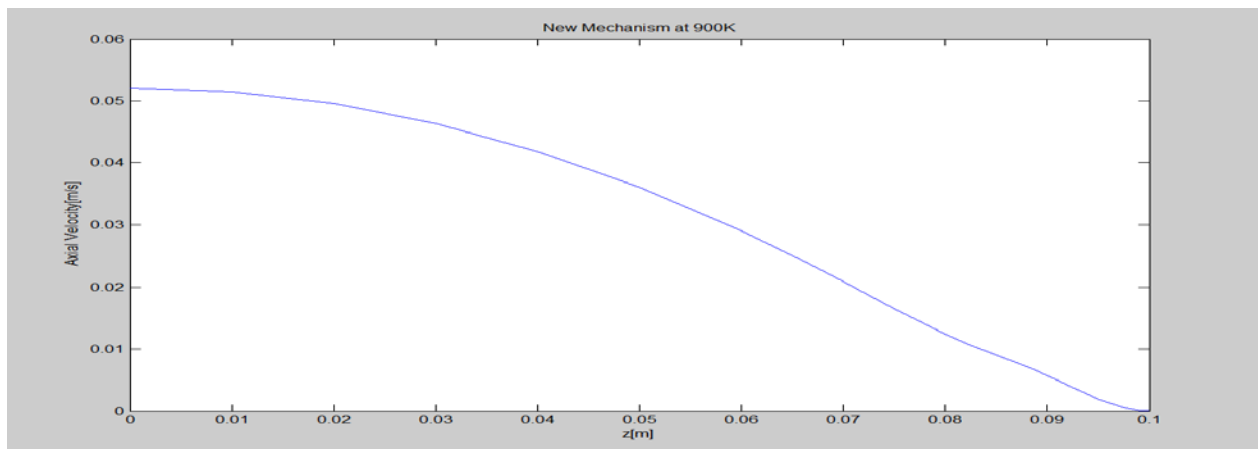


Figure 3-12. Axial velocity profile in new mechanism with faster dissociation at 900K

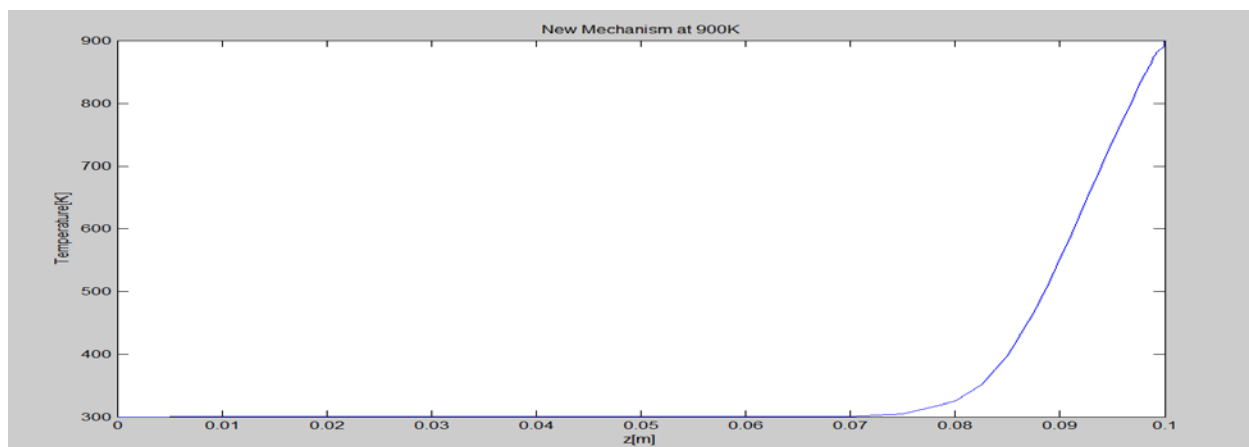


Figure 3-13. Temperature profile in new mechanism with faster dissociation at 900K

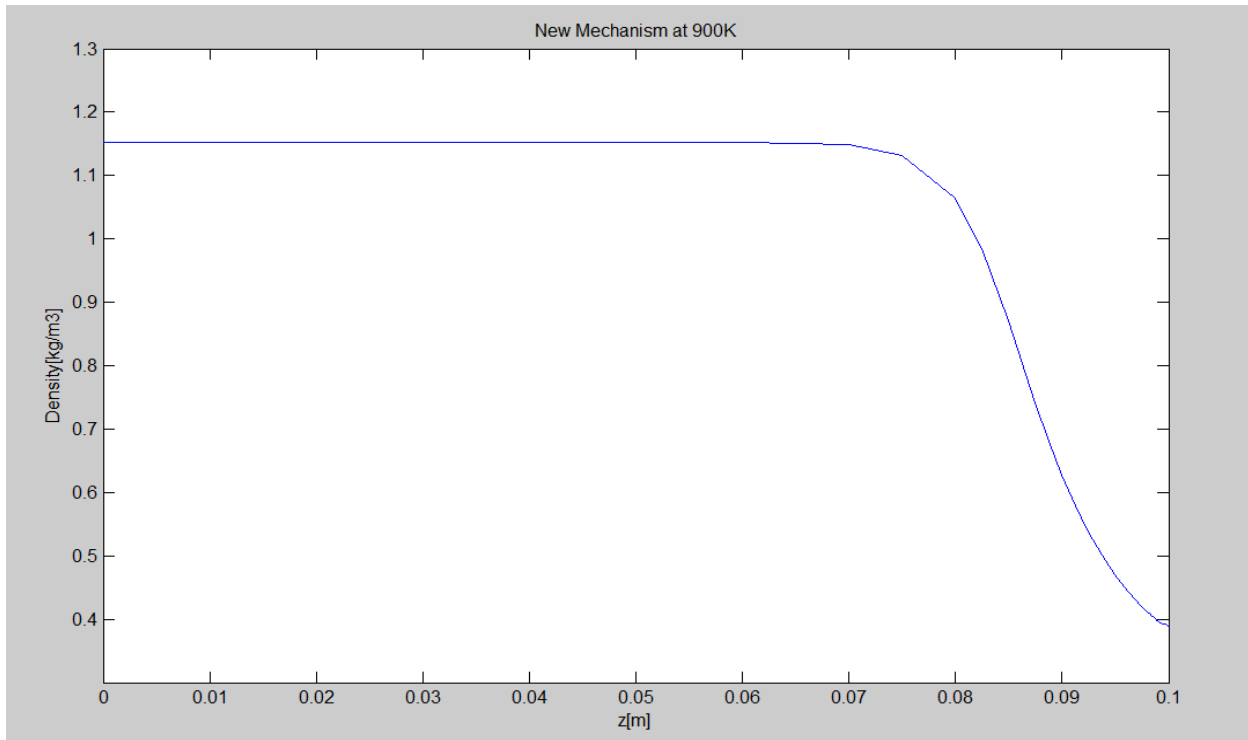


Figure 3-14. Density profile in new mechanism with faster dissociation at 900K

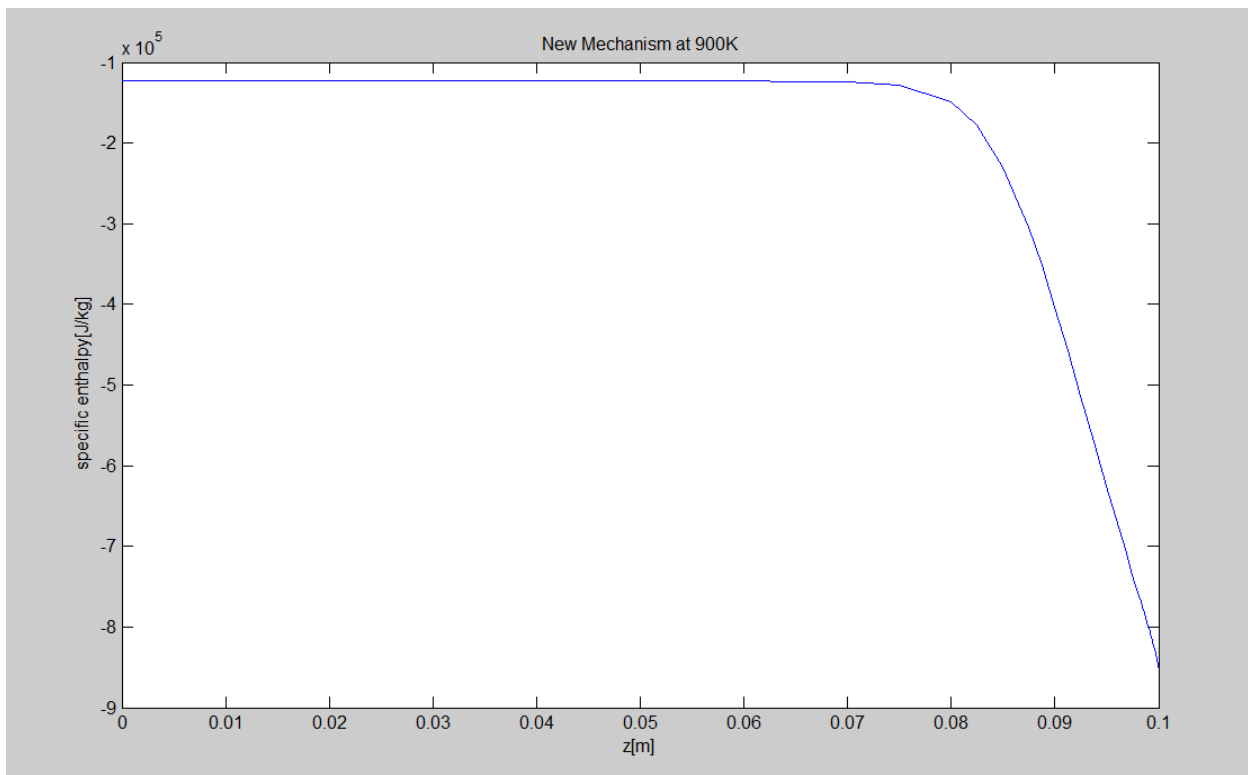


Figure 3-15. Specific enthalpy in new mechanism with faster dissociation at 900K

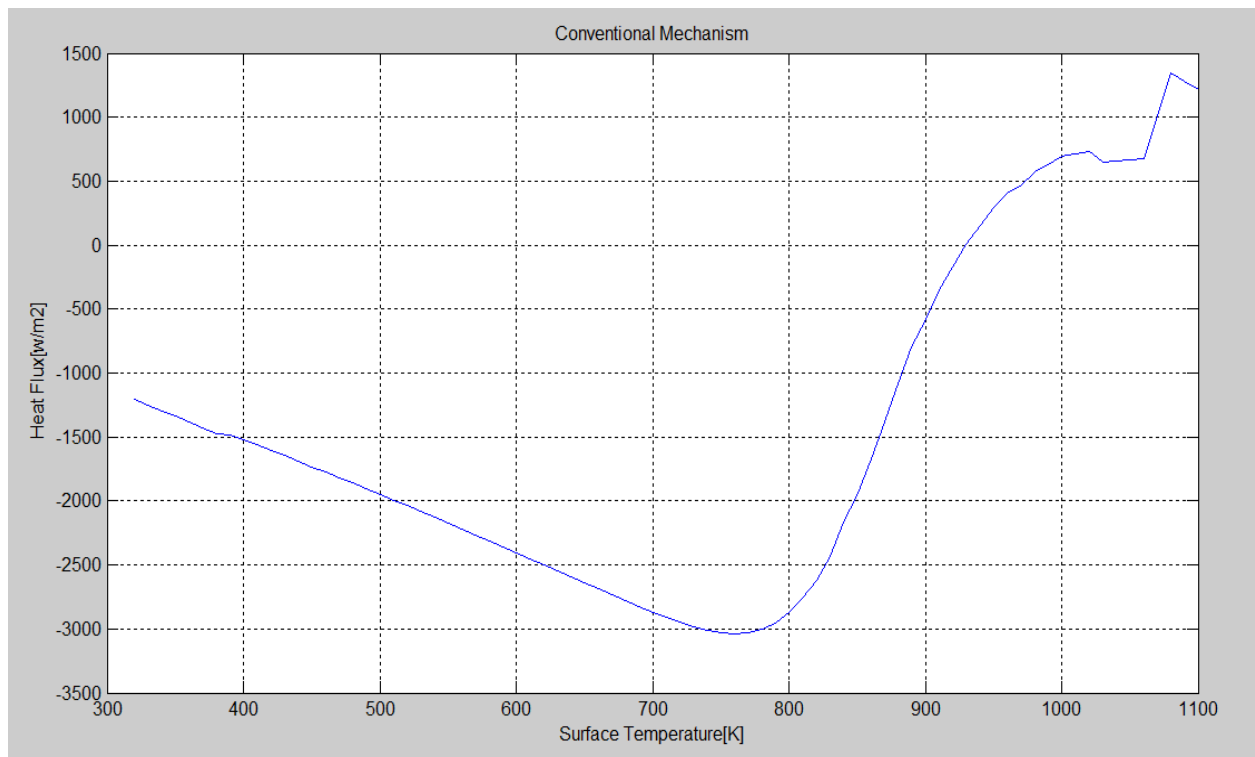


Figure 3-16. Heat transfer to non-reacting flow in Deutschmann

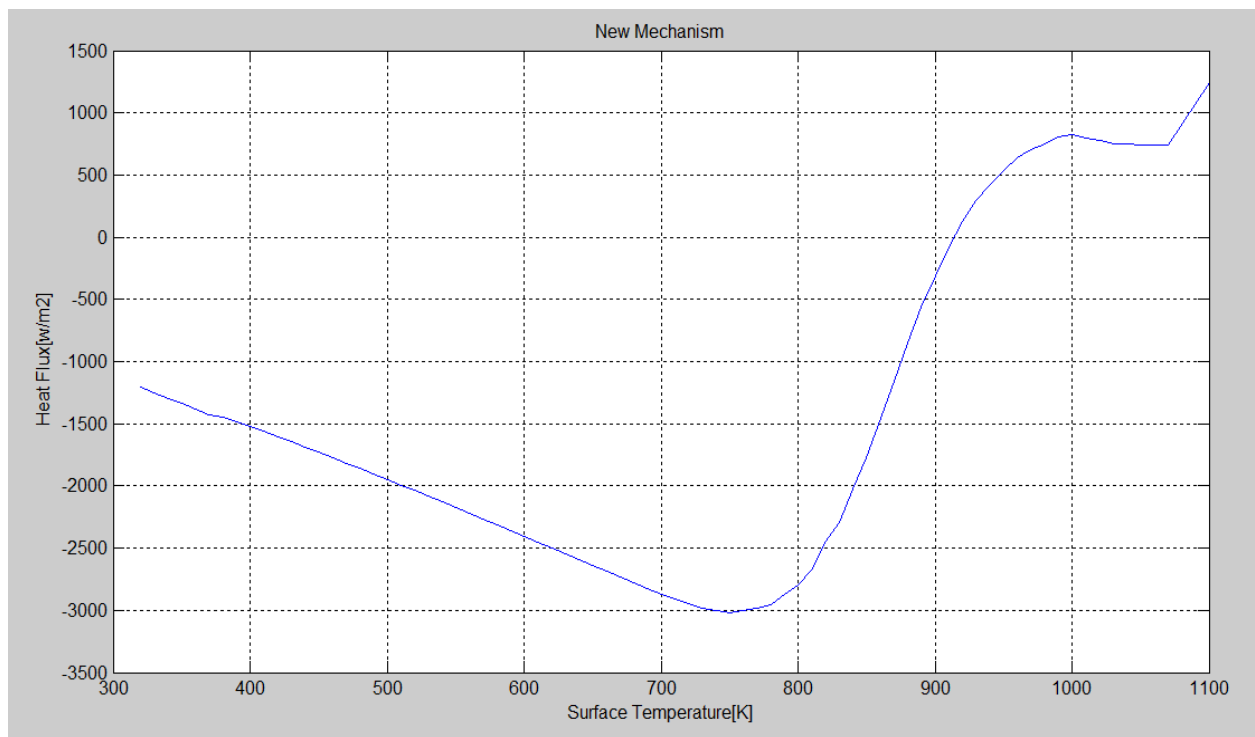


Figure 3-17. Heat transfer to non-reacting flow in new mechanism with faster deactivation

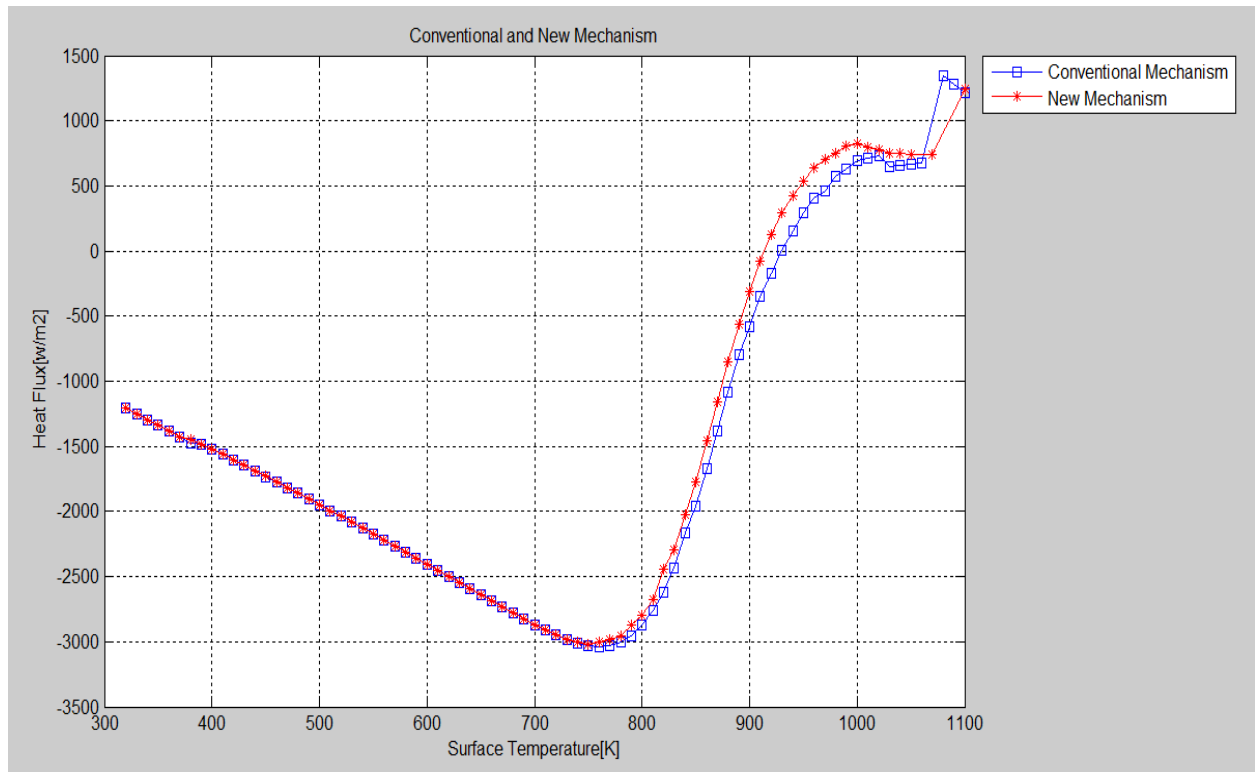


Figure 3-18. Heat transfer to non-reacting flow in conventional and new mechanism

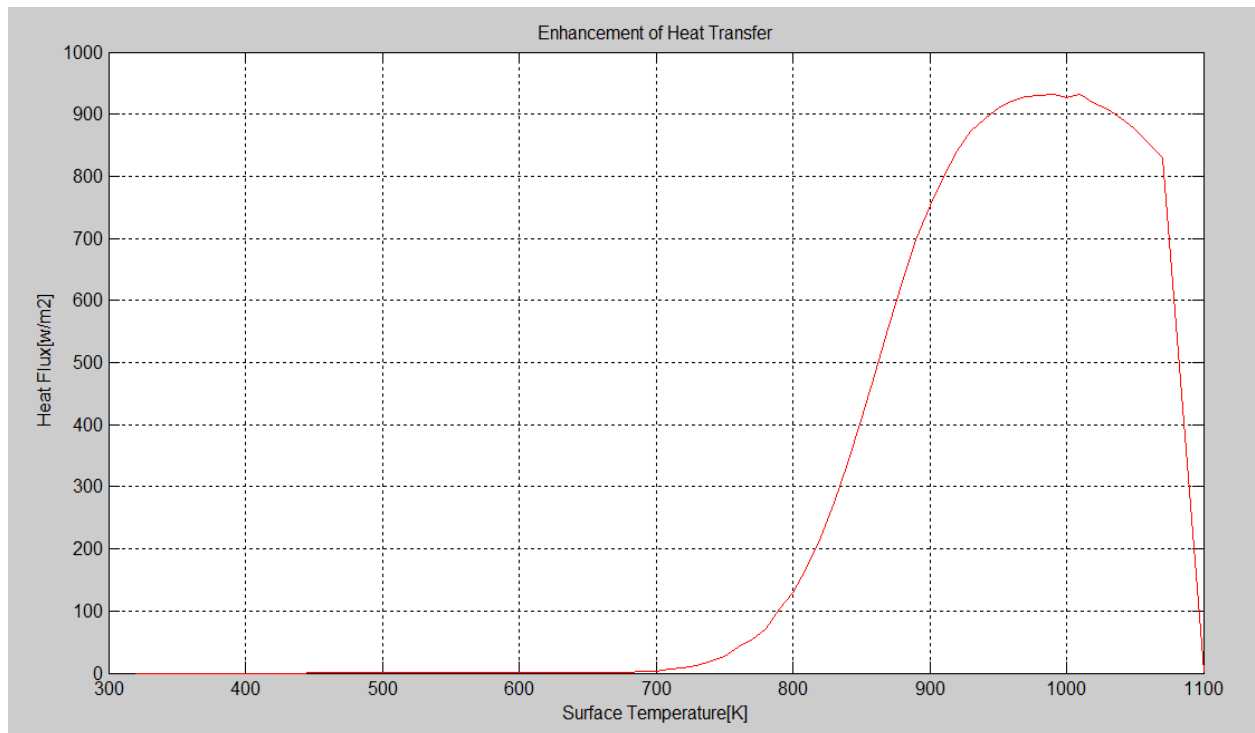


Figure 3-19. Enhancement of heat transfer in new mechanism with faster deactivation

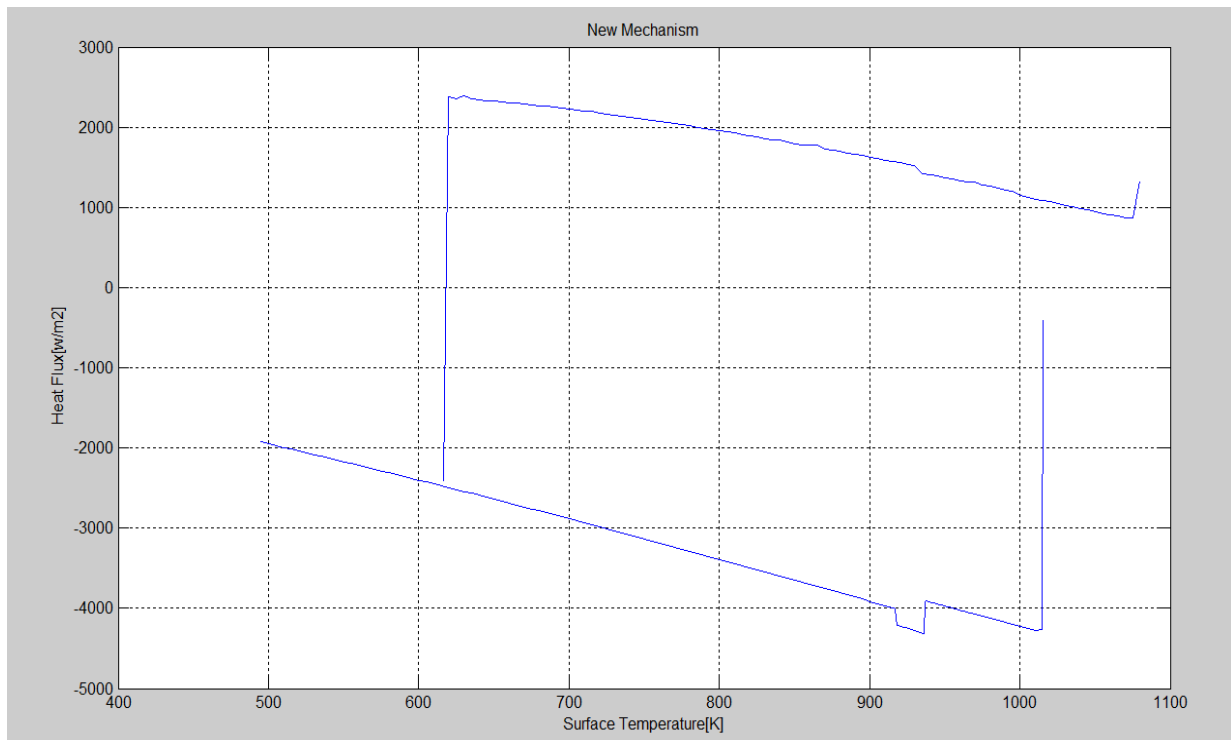


Figure 3-20. Heat transfer to non-reacting flow in new mechanism with faster dissociation

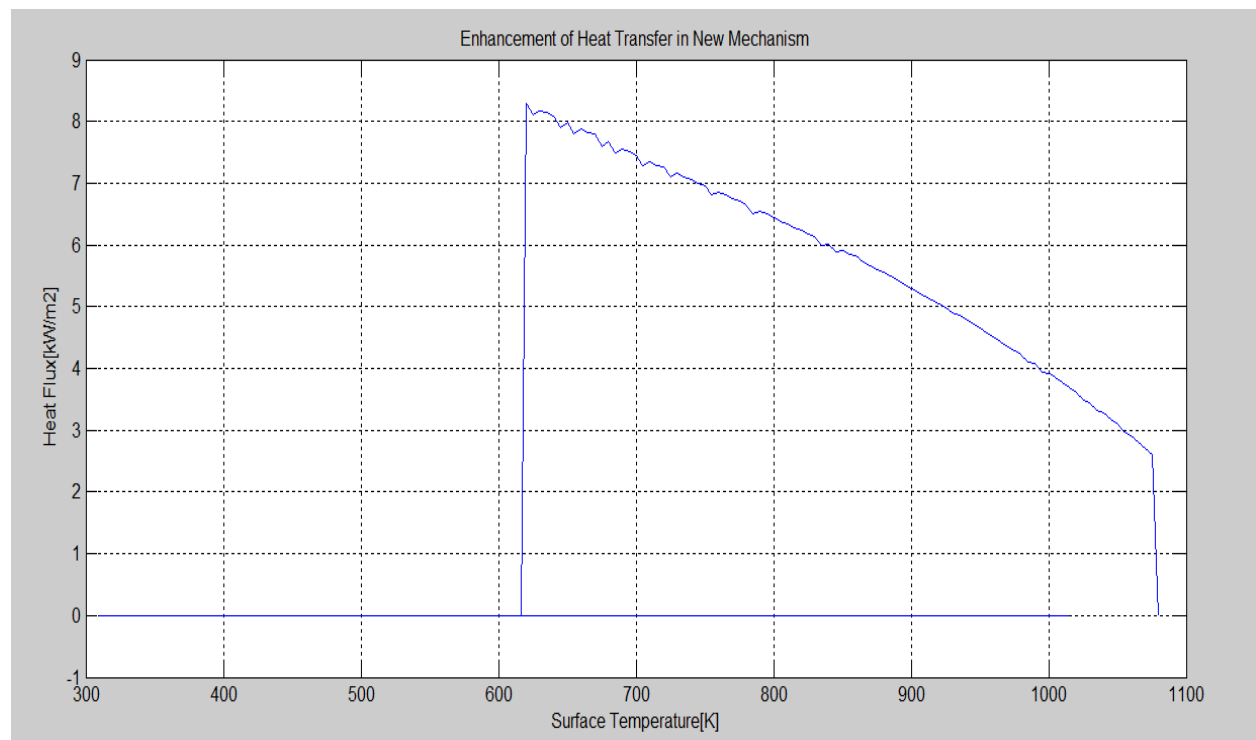
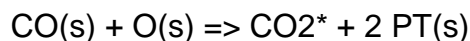
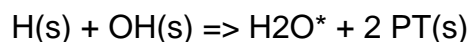
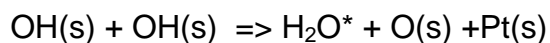


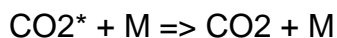
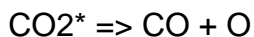
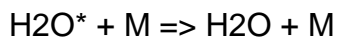
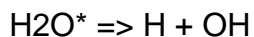
Figure 3-21. Enhancement of heat transfer in new mechanism with faster dissociation

CHAPTER 4 SUMMARY

This mechanism of catalytic methane is postulated mechanism. New mechanism of catalytic methane under enhancement heat transfer is based on GRI30 mechanism¹⁰ for gas reactions and Deutschmann mechanism^{2,11} for surface reaction. This mechanism much more focuses on enhancement of heat transfer than combustion of reacting flow. Activated H_2O and CO_2 are desorbed directly from surface reactions and dissociated or deactivated into either H_2O or $\text{H} + \text{OH}$, and either CO_2 or $\text{CO} + \text{O}$ with different collision frequency but zero activation energy. More experimental work is needed to measure activation energies and collision frequency of dissociation reactions of activated H_2O and CO_2 to correct this mechanism. Main features of new mechanism are following.



Those surface reactions represent that activated H_2O and CO_2 are desorbed directly from surface.



Those gas reactions represents dissociation of activated H_2O and CO_2 with different rate of reaction and zero activation energy in which dissociation reaction occurs automatically without putting energy.

From the simulation, new mechanism shows different path. In New mechanism with faster deactivation, heat flux to non-reacting flow shows similar to conventional catalytic methane mechanism, but slightly more heat transfers to non-reacting flow due to enhancement of heat transfer. Moreover, enhancement of heat transfer starts around 730K and increases with increasing surface temperature by 1000K and then starts to decrease and is suddenly decrease to zero at 1100K since gas reactions are dominant at high temperature. In New Mechanism with faster dissociation, it consists of two different branches that is high and low ignition branch. Heat is put into surface by 1015.2K to ignite, and then reactions suddenly occur with jumping to high branch. After ignition occurs, surface should be cooled down by 616.5K and then jump to the low branch. In order to make automatically circulate combustion process in this region, heating process and cooling process are needed at low branch and high branch respectively.

Amount of heat transfer to non-reacting flow in new mechanism is higher than that in conventional catalytic methane mechanism due to enhancement of heat transfer. And enhancement of heat transfer where excess heat is released from modeled surface reactions is significantly different from heat flux that is calculated by global energy balance equation. This means that classic conventional heat transfer is needed to be corrected under surface reactions. Moreover, heat transfer to non-reacting flow is different in new mechanism according to rate of reactions of dissociation and deactivation. Finally, much more research is required to improve this new mechanism.

APPENDIX A ORIGINAL CATCOMB CODE

```
# CATCOMB -- Catalytic combustion of methane on platinum.
#
# This script solves a catalytic combustion problem. A stagnation flow
# is set up, with a gas inlet 10 cm from a platinum surface at 900
# K. The lean, premixed methane/air mixture enters at ~ 6 cm/s (0.06
# kg/m2/s), and burns catalytically on the platinum surface. Gas-phase
# chemistry is included too, and has some effect very near the
# surface.
#
# The catalytic combustion mechanism is from Deutschman et al., 26th
# Symp. (Intl.) on Combustion, 1996 pp. 1747-1754
#
#
from Cantera import *
from Cantera.OneD import *
#from Cantera.OneD.StagnationFlow import StagnationFlow
import math

#####
#
# Parameter values are collected here to make it easier to modify
# them

p      = OneAtm          # pressure
tinlet = 300.0           # Inlet temperature
tsurf  = 900.0           # surface temperature
mdot   = 0.06            # kg/m^2/s

transport = 'Mix'        # transport model

# We will solve first for a hydrogen/air case to
# use as the initial estimate for the methane/air case

# composition of the inlet premixed gas for the hydrogen/air case
comp1 = 'H2:0.05, O2:0.21, N2:0.78, AR:0.01'

# composition of the inlet premixed gas for the methane/air case
comp2 = 'CH4:0.050, O2:0.21, N2:0.78, AR:0.01'

# the initial grid, in meters. The inlet/surface separation is 10 cm.
initial_grid = [0.0, 0.02, 0.04, 0.06, 0.08, 0.1] # m
```

```

# numerical parameters
tol_ss  = [1.0e-5, 1.0e-9]    # [rtol, atol] for steady-state problem
tol_ts  = [1.0e-4, 1.0e-9]    # [rtol, atol] for time stepping

loglevel = 5                  # amount of diagnostic output
                                # (0 to 5)

refine_grid = 1               # 1 to enable refinement, 0 to
                                # disable

##### create the gas object #####
#
# This object will be used to evaluate all thermodynamic, kinetic,
# and transport properties
#
# The gas phase will be taken from the definition of phase 'gas' in
# input file 'ptcombust.cti,' which is a stripped-down version of
# GRI-Mech 3.0.
gas = importPhase('ptcombust.cti','gas')
gas.set(T = tinlet, P = p, X = comp1)

##### create the interface object #####
#
# This object will be used to evaluate all surface chemical production
# rates. It will be created from the interface definition 'Pt_surf'
# in input file 'ptcombust.cti,' which implements the reaction
# mechanism of Deutschmann et al., 1995 for catalytic combustion on
# platinum.
#
surf_phase = importInterface('ptcombust.cti','Pt_surf', [gas])
surf_phase.setTemperature(tsurf)

# integrate the coverage equations in time for 1 s, holding the gas
# composition fixed to generate a good starting estimate for the
# coverages.
surf_phase.advanceCoverages(1.0)

# create the object that simulates the stagnation flow, and specify an
# initial grid
sim = StagnationFlow(gas = gas, surfchem = surf_phase,
                    grid = initial_grid)

# Objects of class StagnationFlow have members that represent the gas inlet ('inlet')
and the surface ('surface'). Set some parameters of these objects.

```



```

sim.inlet.set(mdot = mdot, T = tinlet, X = comp1)
sim.surface.set(T = tsurf)

# Set error tolerances
sim.set(tol = tol_ss, tol_time = tol_ts)

# Method 'init' must be called before beginning a simulation
sim.init()

# Show the initial solution estimate
sim.showSolution()

# Solving problems with stiff chemistry coupled to flow can require
# a sequential approach where solutions are first obtained for
# simpler problems and used as the initial guess for more difficult
# problems.

# start with the energy equation on (default is 'off')
sim.set(energy = 'on')

# disable the surface coverage equations, and turn off all gas and
# surface chemistry.
sim.surface.setCoverageEqs('off')
surf_phase.setMultiplier(0.0);
gas.setMultiplier(0.0);

# solve the problem, refining the grid if needed, to determine the
# non-reacting velocity and temperature distributions

sim.solve(loglevel, refine_grid)

# now turn on the surface coverage equations, and turn the
# chemistry on slowly
sim.surface.setCoverageEqs('on')
for iter in range(6):
    mult = math.pow(10.0,(iter - 5));
    surf_phase.setMultiplier(mult);
    gas.setMultiplier(mult);
    print 'Multiplier = ',mult
    sim.solve(loglevel, refine_grid);

# At this point, we should have the solution for the hydrogen/air
# problem.

sim.showSolution()

```

```

# Now switch the inlet to the methane/air composition.

sim.inlet.set(X = comp2)

# set more stringent grid refinement criteria
sim.setRefineCriteria(100.0, 0.15, 0.2, 0.0)

# solve the problem for the final time
sim.solve(loglevel, refine_grid)

# show the solution
sim.showSolution()

# save the solution in XML format. The 'restore' method can be used to restart
# a simulation from a solution stored in this form.

sim.save("catcomb.xml", "sol")

# save selected solution components in a CSV file for plotting in
# Excel or MATLAB.

# These methods return arrays containing the values at all grid points
z = sim.flow.grid()
T = sim.T()
u = sim.u()
V = sim.V()

f = open('catcomb.csv', 'w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'Tc(K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames()) + list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    Tc=((T[n]-300)*tsurf)/(tsurf-tinlet) + 300
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], Tc, gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))

# write the surface coverages to the CSV file
writeCSV(f, ['species on the surface', 'converages', 'concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()

```

```

names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n], con[n]])

for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT())
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface'] +list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface'] +list(surf_phase.netProductionRates()))
f.close()

print 'solution saved to catcomb.csv'

# show some statistics
sim.showStats()

```

APPENDIX B
MODIFIED CATCOMB CODE FOR LOW IGNITION BRANCH

```
# CATCOMB -- Catalytic combustion of methane on platinum.
#
# This script solves a catalytic combustion problem. A stagnation flow
# is set up, with a gas inlet 10 cm from a platinum surface at 900
# K. The lean, premixed methane/air mixture enters at ~ 6 cm/s (0.06
# kg/m2/s), and burns catalytically on the platinum surface. Gas-phase
# chemistry is included too, and has some effect very near the
# surface.
#
# The catalytic combustion mechanism is from Deutschman et al., 26th
# Symp. (Intl.) on Combustion, 1996 pp. 1747-1754
#
#
from Cantera import *
from Cantera.OneD import *
#from Cantera.OneD.StagnationFlow import StagnationFlow
import math

#####
#
# Parameter values are collected here to make it easier to modify
# them

p      = OneAtm          # pressure
tinlet = 300.0           # Inlet temperature
tstart = 700.0           # starting temperature
tsurf  = 1016.0          # surface temperature
mdot   = 0.06            # kg/m^2/s
i      = 1.0             # Increment of Surface Temperature
transport = 'Mix'        # transport model

# We will solve first for a hydrogen/air case to
# use as the initial estimate for the methane/air case

# composition of the inlet premixed gas for the hydrogen/air case
comp1   = 'H2:0.05, O2:0.21, N2:0.78, AR:0.01'

# composition of the inlet premixed gas for the methane/air case
comp2   = 'CH4:0.050, O2:0.21, N2:0.78, AR:0.01'

# the initial grid, in meters. The inlet/surface separation is 10 cm.
```

```

initial_grid = [0.0, 0.02, 0.04, 0.06, 0.08, 0.1] # m

# numerical parameters
tol_ss = [1.0e-10, 1.0e-10] # [rtol, atol] for steady-state problem
tol_ts = [1.0e-4, 1.0e-9] # [rtol, atol] for time stepping

loglevel = 2 # amount of diagnostic output
              # (0 to 5)

refine_grid = 1 # 1 to enable refinement, 0 to
                # disable

##### create the gas object #####
#
# This object will be used to evaluate all thermodynamic, kinetic,
# and transport properties
#
# The gas phase will be taken from the definition of phase 'gas' in
# input file 'ptcombust.cti,' which is a stripped-down version of
# GRI-Mech 3.0.
gas = importPhase('ptcombust.cti','gas')
gas.set(T = tinlet, P = p, X = comp1)

##### create the interface object #####
#
# This object will be used to evaluate all surface chemical production
# rates. It will be created from the interface definition 'Pt_surf'
# in input file 'ptcombust.cti,' which implements the reaction
# mechanism of Deutschmann et al., 1995 for catalytic combustion on
# platinum.
#
surf_phase = importInterface('ptcombust.cti','Pt_surf', [gas])

surf_phase.setTemperature(481)
surf_phase.advanceCoverages(1.0)
sim = StagnationFlow(gas = gas, surfchem = surf_phase,
                    grid = initial_grid)
sim.inlet.set(mdot = mdot, T = tinlet, X = comp1)
sim.surface.set(T = 481)
sim.set(tol = tol_ss, tol_time = tol_ts)
sim.init()
sim.showSolution()
sim.set(energy = 'on')
sim.surface.setCoverageEqs('off')

```

```

surf_phase.setMultiplier(0.0);
gas.setMultiplier(0.0);
sim.solve(loglevel, refine_grid)
sim.surface.setCoverageEqs('on')
for iter in range(6):
    mult = math.pow(10.0,(iter - 5));
    surf_phase.setMultiplier(mult);
    gas.setMultiplier(mult);
    print 'Multiplier = ',mult
    sim.solve(loglevel, refine_grid);
sim.showSolution()
sim.inlet.set(X = comp2)
sim.setRefineCriteria(100.0, 0.15, 0.2, 0.0)
sim.solve(loglevel, refine_grid)
sim.showSolution()

surf_phase.setTemperature(500)
surf_phase.advanceCoverages(1.0)
sim = StagnationFlow(gas = gas, surfchem = surf_phase,
                    grid = initial_grid)
sim.inlet.set(mdot = mdot, T = tinlet, X = comp1)
sim.surface.set(T = 500)
sim.set(tol = tol_ss, tol_time = tol_ts)
sim.init()
sim.showSolution()
sim.set(energy = 'on')
sim.surface.setCoverageEqs('off')
surf_phase.setMultiplier(0.0);
gas.setMultiplier(0.0);
sim.solve(loglevel, refine_grid)
sim.surface.setCoverageEqs('on')
for iter in range(6):
    mult = math.pow(10.0,(iter - 5));
    surf_phase.setMultiplier(mult);
    gas.setMultiplier(mult);
    print 'Multiplier = ',mult
    sim.solve(loglevel, refine_grid);
sim.showSolution()
sim.inlet.set(X = comp2)
sim.setRefineCriteria(100.0, 0.15, 0.2, 0.0)
sim.solve(loglevel, refine_grid)
sim.showSolution()

surf_phase.setTemperature(600)
surf_phase.advanceCoverages(1.0)
sim = StagnationFlow(gas = gas, surfchem = surf_phase,

```

```

        grid = initial_grid)
sim.inlet.set(mdot = mdot, T = tinlet, X = comp1)
sim.surface.set(T = 600)
sim.set(tol = tol_ss, tol_time = tol_ts)
sim.init()
sim.showSolution()
sim.set(energy = 'on')
sim.surface.setCoverageEqs('off')
surf_phase.setMultiplier(0.0);
gas.setMultiplier(0.0);
sim.solve(loglevel, refine_grid)
sim.surface.setCoverageEqs('on')
for iter in range(6):
    mult = math.pow(10.0,(iter - 5));
    surf_phase.setMultiplier(mult);
    gas.setMultiplier(mult);
    print 'Multiplier = ',mult
    sim.solve(loglevel, refine_grid);
sim.showSolution()
sim.inlet.set(X = comp2)
sim.setRefineCriteria(100.0, 0.15, 0.2, 0.0)
sim.solve(loglevel, refine_grid)
sim.showSolution()
while tstart < tsurf:
    tstart=tstart+ i
    surf_phase.setTemperature(tstart)
    surf_phase.advanceCoverages(1.0)
    sim = StagnationFlow(gas = gas, surfchem = surf_phase,
        grid = initial_grid)
    sim.inlet.set(mdot = mdot, T = tinlet, X = comp1)
    sim.surface.set(T = tstart)
    sim.set(tol = tol_ss, tol_time = tol_ts)
    sim.init()
    sim.showSolution()
    sim.set(energy = 'on')
    sim.surface.setCoverageEqs('off')
    surf_phase.setMultiplier(0.0);
    gas.setMultiplier(0.0);
    sim.solve(loglevel, refine_grid)
    sim.surface.setCoverageEqs('on')
    for iter in range(6):
        mult = math.pow(10.0,(iter - 5));
        surf_phase.setMultiplier(mult);
        gas.setMultiplier(mult);
        print 'Multiplier = ',mult
        sim.solve(loglevel, refine_grid);

```

```

sim.showSolution()
sim.inlet.set(X = comp2)
sim.setRefineCriteria(100.0, 0.15, 0.2, 0.0)
sim.solve(loglevel, refine_grid)
sim.showSolution()
if tstart == 620:
    sim.save("620.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('620m5.csv', 'w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
            + list(gas.speciesNames()) + list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                + list(gas.moleFractions()) + list(gas.creationRates()))
    writeCSV(f, ['species on the surface', 'coverages', 'concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n], con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f, ['fwdrateconst'] + list(surf_phase.fwdRateConstants()))
    writeCSV(f, ['fwdrateprogress'] + list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f, ['species on the surface'] + list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f, ['nasa enthalpies'] + list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f, ['creationRate on the surface'] + list(surf_phase.creationRates()))
    writeCSV(f, ['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f, ['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()

elif tstart == 625:
    sim.save("625.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()

```



```

u = sim.u()
V = sim.V()
f = open('625m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
        +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n], con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 630:
    sim.save("630.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('630m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()])

```

```

        +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','coverages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()
elif tstart == 635:
    sim.save("635.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('635m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','coverages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))

```

```

writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 640:
sim.save("640.xml","sol")
z = sim.flow.grid()
T = sim.T()
u = sim.u()
V = sim.V()
f = open('640m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
+ list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
sim.setGasState(n)
writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
+list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
rxn=surf_phase.reactionEqn(n)
writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))

```

```

        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 645:
        sim.save("645.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('645m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
            writeCSV(f, ['species on the surface','converages','concentrations'])
            cov = sim.coverages()
            con = surf_phase.concentrations()
            names = surf_phase.speciesNames()
            for n in range(len(names)):
                writeCSV(f, [names[n], cov[n],con[n]])
            for n in range(24):
                rxn=surf_phase.reactionEqn(n)
                writeCSV(f, list([rxn]))
            writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
            writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
            writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
            writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
            writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
            writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
            writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
            f.close()
    elif tstart == 650:
        sim.save("650.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('650m5.csv','w')

```

```

        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):
            writeCSV(f, [names[n], cov[n],con[n]])
        for n in range(24):
            rxn=surf_phase.reactionEqn(n)
            writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 655:
        sim.save("655.xml", "sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('655m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])
        cov = sim.coverages()

```

```

con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n], con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 660:
    sim.save("660.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('660m5.csv', 'w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface', 'converages', 'concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n], con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))

```

```

        writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 665:
        sim.save("665.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('665m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
            writeCSV(f, ['species on the surface','converages','concentrations'])
            cov = sim.coverages()
            con = surf_phase.concentrations()
            names = surf_phase.speciesNames()
            for n in range(len(names)):
                writeCSV(f, [names[n], cov[n],con[n]])
            for n in range(24):
                rxn=surf_phase.reactionEqn(n)
                writeCSV(f, list([rxn]))
            writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
            writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
            writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
            writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
            writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
            writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
            writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
            f.close()

```

```

elif tstart == 670:
    sim.save("670.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('670m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','converages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()
elif tstart == 675:
    sim.save("675.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('675m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))

```



```

for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','converages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()
elif tstart == 680:
    sim.save("680.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('680m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
            + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):

```

```

        writeCSV(f, [names[n], cov[n], con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 685:
        sim.save("685.xml", "sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('685m5.csv', 'w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface', 'converages', 'concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):
            writeCSV(f, [names[n], cov[n], con[n]])
        for n in range(24):
            rxn=surf_phase.reactionEqn(n)
            writeCSV(f, list([rxn]))
            writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
            writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
            writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
            writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))

```

```

        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 690:
        sim.save("690.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('690.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','coverages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):
            writeCSV(f, [names[n], cov[n],con[n]])
        for n in range(24):
            rxn=surf_phase.reactionEqn(n)
            writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 695:
        sim.save("695.xml","sol")
        z = sim.flow.grid()
        T = sim.T()

```

```

u = sim.u()
V = sim.V()
f = open('695.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdRateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdRateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 700:
    sim.save("700.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('700m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
            + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()])

```

```

        +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','coverages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()
elif tstart == 705:
    sim.save("705.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('705m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','coverages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))

```

```

writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 710:
sim.save("710.xml","sol")
z = sim.flow.grid()
T = sim.T()
u = sim.u()
V = sim.V()
f = open('710m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
+ list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
sim.setGasState(n)
writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
+list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
rxn=surf_phase.reactionEqn(n)
writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))

```

```

        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()

.
.
.
.
.
.

elif tstart == 930:
    sim.save("930.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('930m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','converages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()

```

```

elif tstart == 935:
    sim.save("935.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('935m5.csv', 'w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
            + list(gas.speciesNames()) + list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                + list(gas.moleFractions()) + list(gas.creationRates()))
    writeCSV(f, ['species on the surface', 'coverages', 'concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n], con[n]])
    for n in range(24):
        rxn = surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f, ['fwdRateconst'] + list(surf_phase.fwdRateConstants()))
    writeCSV(f, ['fwdRateprogress'] + list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f, ['species on the surface'] + list(gas.speciesNames())
+list(surf_phase.speciesNames()))
    writeCSV(f, ['nasa enthalpies'] + list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f, ['creationRate on the surface'] + list(surf_phase.creationRates()))
    writeCSV(f, ['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f, ['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()

elif tstart == 940:
    sim.save("940.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('940m5.csv', 'w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']

```



```

        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','coverages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()
    elif tstart ==945:
        sim.save("945.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('945m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','coverages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()

```

```

for n in range(len(names)):
    writeCSV(f, [names[n], cov[n], con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()

elif tstart == 950:
    sim.save("950.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('950m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
    + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
    +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','converages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n], con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))

```

```

        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart ==955:
        sim.save("955.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('955m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
            + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                +list(gas.moleFractions()) +list(gas.creationRates()))
            writeCSV(f, ['species on the surface','converages','concentrations'])
            cov = sim.coverages()
            con = surf_phase.concentrations()
            names = surf_phase.speciesNames()
            for n in range(len(names)):
                writeCSV(f, [names[n], cov[n],con[n]])
            for n in range(24):
                rxn=surf_phase.reactionEqn(n)
                writeCSV(f, list([rxn]))
            writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
            writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
            writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
            writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
            writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
            writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
            writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
            f.close()

    elif tstart == 960:

```

```

sim.save("960.xml","sol")
z = sim.flow.grid()
T = sim.T()
u = sim.u()
V = sim.V()
f = open('960m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
        +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart ==965:
    sim.save("965.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('965m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):

```

```

        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):
            writeCSV(f, [names[n], cov[n], con[n]])
        for n in range(24):
            rxn=surf_phase.reactionEqn(n)
            writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()

    elif tstart == 970:
        sim.save("970.xml", "sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('970m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
            writeCSV(f, ['species on the surface','converages','concentrations'])
            cov = sim.coverages()
            con = surf_phase.concentrations()
            names = surf_phase.speciesNames()
            for n in range(len(names)):

```

```

        writeCSV(f, [names[n], cov[n], con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart ==975:
        sim.save("975.xml", "sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('975m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):
            writeCSV(f, [names[n], cov[n], con[n]])
        for n in range(24):
            rxn=surf_phase.reactionEqn(n)
            writeCSV(f, list([rxn]))
            writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
            writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
            writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
            writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))

```

```

        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart == 980:
        sim.save("980.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('980m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])
        cov = sim.coverages()
        con = surf_phase.concentrations()
        names = surf_phase.speciesNames()
        for n in range(len(names)):
            writeCSV(f, [names[n], cov[n],con[n]])
        for n in range(24):
            rxn=surf_phase.reactionEqn(n)
            writeCSV(f, list([rxn]))
        writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
        writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
        writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
        writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
        writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
        writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart ==985:
        sim.save("985.xml","sol")
        z = sim.flow.grid()
        T = sim.T()

```

```

u = sim.u()
V = sim.V()
f = open('985m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
        +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n], con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 990:
    sim.save("990.xml", "sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('990m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()])

```



```

        +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()
elif tstart ==995:
    sim.save("995.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('995m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
    writeCSV(f, ['species on the surface','converages','concentrations'])
    cov = sim.coverages()
    con = surf_phase.concentrations()
    names = surf_phase.speciesNames()
    for n in range(len(names)):
        writeCSV(f, [names[n], cov[n],con[n]])
    for n in range(24):
        rxn=surf_phase.reactionEqn(n)
        writeCSV(f, list([rxn]))

```

```

writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart == 1000:
sim.save("1000.xml", "sol")
z = sim.flow.grid()
T = sim.T()
u = sim.u()
V = sim.V()
f = open('1000m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
+ list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
sim.setGasState(n)
writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
+list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
rxn=surf_phase.reactionEqn(n)
writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))

```

```

        writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
        f.close()
    elif tstart ==1005:
        sim.save("1005.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()
        f = open('1005m5.csv','w')
        writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
                + list(gas.speciesNames())+ list(gas.speciesNames()))
        for n in range(sim.flow.nPoints()):
            sim.setGasState(n)
            writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                    +list(gas.moleFractions()) +list(gas.creationRates()))
            writeCSV(f, ['species on the surface','converages','concentrations'])
            cov = sim.coverages()
            con = surf_phase.concentrations()
            names = surf_phase.speciesNames()
            for n in range(len(names)):
                writeCSV(f, [names[n], cov[n],con[n]])
            for n in range(24):
                rxn=surf_phase.reactionEqn(n)
                writeCSV(f, list([rxn]))
            writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
            writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
            writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
            writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
            writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
            writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
            writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
            f.close()

    elif tstart == 1010:
        sim.save("1010.xml","sol")
        z = sim.flow.grid()
        T = sim.T()
        u = sim.u()
        V = sim.V()

```

```

f = open('1010m5.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
        + list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
            +list(gas.moleFractions()) +list(gas.creationRates()))
writeCSV(f, ['species on the surface','converages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
elif tstart ==1015:
    sim.save("1015.xml","sol")
    z = sim.flow.grid()
    T = sim.T()
    u = sim.u()
    V = sim.V()
    f = open('1015m5.csv','w')
    writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
            + list(gas.speciesNames())+ list(gas.speciesNames()))
    for n in range(sim.flow.nPoints()):
        sim.setGasState(n)
        writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
                +list(gas.moleFractions()) +list(gas.creationRates()))
        writeCSV(f, ['species on the surface','converages','concentrations'])

```

```

cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n], con[n]])
for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
writeCSV(f,['species on the surface'] +list(gas.speciesNames()))
+list(surf_phase.speciesNames()))
writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
writeCSV(f,['destructionRate on the surface']
+list(surf_phase.destructionRates()))
writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
f.close()
else:
    pass
else:
# save selected solution components in a CSV file for plotting in
# Excel or MATLAB.

# These methods return arrays containing the values at all grid points
z = sim.flow.grid()
T = sim.T()
u = sim.u()
V = sim.V()

f = open('catcomb.csv','w')
writeCSV(f, ['z (m)', 'u (m/s)', 'V (1/s)', 'T (K)', 'rho (kg/m3)', 'specific enthalpy
(J/kg)', 'thermal conductivity (w/m2 k)']
+ list(gas.speciesNames())+ list(gas.speciesNames()))
for n in range(sim.flow.nPoints()):
    sim.setGasState(n)
    writeCSV(f, [z[n], u[n], V[n], T[n], gas.density(), gas.enthalpy_mass(),
gas.thermalConductivity()]
+list(gas.moleFractions()) +list(gas.creationRates()))

# write the surface coverages to the CSV file

```

```

writeCSV(f, ['species on the surface','coverages','concentrations'])
cov = sim.coverages()
con = surf_phase.concentrations()
names = surf_phase.speciesNames()
for n in range(len(names)):
    writeCSV(f, [names[n], cov[n],con[n]])

for n in range(24):
    rxn=surf_phase.reactionEqn(n)
    writeCSV(f, list([rxn]))
    writeCSV(f,['fwdrateconst'] +list(surf_phase.fwdRateConstants()))
    writeCSV(f,['fwdrateprogress'] +list(surf_phase.fwdRatesOfProgress()))
    writeCSV(f,['species on the surface'] +list(gas.speciesNames())
+list(surf_phase.speciesNames()))
    writeCSV(f,['nasa enthalpies'] +list(gas.enthalpies_RT()))
+list(surf_phase.enthalpies_RT()))
    writeCSV(f,['creationRate on the surface'] +list(surf_phase.creationRates()))
    writeCSV(f,['destructionRate on the surface'] +list(surf_phase.destructionRates()))
    writeCSV(f,['netproductionRate on the surface']
+list(surf_phase.netProductionRates()))
    f.close()

print 'solution saved to catcomb.csv'

# show some statistics
sim.showStats()

```

APPENDIX C
HEAT TRANSFER IN CONVENTIONAL MECHANISM

Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)
320	-1208.851756	570	-2264.233311	820	-2615.891164
330	-1251.997502	580	-2310.562397	830	-2431.902454
340	-1295.43543	590	-2357.164711	840	-2162.030161
350	-1339.150342	600	-2404.030039	850	-1957.018381
360	-1383.141395	610	-2448.389805	860	-1666.963561
370	-1427.407332	620	-2495.520185	870	-1381.820807
380	-1471.945201	630	-2542.82805	880	-1082.607617
390	-1487.748976	640	-2590.257452	890	-791.3083427
400	-1521.519835	650	-2637.724776	900	-585.4077531
410	-1562.834846	660	-2685.103284	910	-352.3566663
420	-1604.443135	670	-2732.206513	920	-170.6122831
430	-1646.347262	680	-2778.708518	930	6.509903484
440	-1688.510745	690	-2824.327818	940	156.9871032
450	-1731.005429	700	-2868.5167	950	293.3556183
460	-1773.800302	710	-2910.402184	960	407.3807495
470	-1816.895759	720	-2948.938553	970	458.275166
480	-1860.291978	730	-2982.893881	980	572.6410474
490	-1903.988679	740	-3010.380367	990	630.588355
500	-1947.985226	750	-3029.109716	1000	691.1099035
510	-1992.280654	760	-3036.241816	1010	709.0198011
520	-2036.873683	770	-3028.325716	1020	736.4317096
530	-2081.762723	780	-3001.447693	1030	648.6298186
540	-2126.945844	790	-2951.147096	1040	661.4834314
550	-2172.420707	800	-2872.858013	1050	669.5599729
560	-2218.184428	810	-2762.259835	1060	673.8173992

Figure C-1. Heat flux to non-reacting flow in Deutschmann

APPENDIX D
HEAT TRANSFER IN NEW MECHANISM WITH FASTER DEACTIVATION

Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)
320	-1208.832788	570	-2263.846732	820	-2440.335562
330	-1251.997502	580	-2310.46517	830	-2296.148746
340	-1295.437579	590	-2356.707548	840	-2019.691494
350	-1339.150342	600	-2404.095312	850	-1770.920409
360	-1383.141395	610	-2448.9447	860	-1452.912052
370	-1427.407308	620	-2496.087981	870	-1154.64318
380	-1446.507005	630	-2542.798916	880	-855.3941756
390	-1487.748976	640	-2590.210047	890	-563.7074178
400	-1521.519835	650	-2637.643267	900	-309.1216384
410	-1562.834846	660	-2684.96025	910	-78.54027108
420	-1604.443135	670	-2731.911908	920	128.0218575
430	-1646.347262	680	-2778.2529	930	298.1042759
440	-1688.510745	690	-2823.628785	940	423.7855659
450	-1730.903316	700	-2867.320532	950	540.7495625
460	-1773.694658	710	-2908.502428	960	636.3621878
470	-1817.739189	720	-2946.065782	970	700.0737941
480	-1860.140393	730	-2979.352125	980	753.0006626
490	-1903.818217	740	-2999.074347	990	810.2759668
500	-1947.81233	750	-3018.133676	1000	827.6326582
510	-1993.719519	760	-2999.165361	1010	798.2652918
520	-2036.644055	770	-2986.793849	1020	782.830616
530	-2082.060986	780	-2953.301077	1030	751.1825923
540	-2126.652363	790	-2867.003601	1040	749.9976656
550	-2172.927941	800	-2800.923459	1050	743.5418018
560	-2218.0777	810	-2676.991845	1070	738.5903893
				1100	1241.307963

Figure D-1. Heat flux to non-reacting flow in new mechanism with faster deactivation

APPENDIX E
ENHANCEMENT OF HEAT TRANSFER IN NEW MECHANISM WITH FASTER
DEACTIVATION

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
320	6.48E-73	-1.52E-37	-5.16E-61	-1.52E-37
330	1.61E-82	2.21E-42	2.61E-69	2.21E-42
340	6.41E-81	1.42E-41	-2.42E-66	1.42E-41
350	2.09E-82	3.18E-42	-4.33E-66	3.18E-42
360	1.10E-76	-5.32E-39	-8.26E-61	-5.32E-39
370	1.83E-48	-6.29E-24	2.11E-39	-6.29E-24
380	3.91E-44	-1.22E-19	5.12E-34	-1.22E-19
390	2.27E-48	-3.70E-24	1.87E-39	-3.70E-24
400	2.15E-71	-6.81E-37	2.87E-56	-6.81E-37
410	4.32E-76	9.41E-40	-3.42E-64	9.41E-40
420	3.12E-73	1.41E-38	7.28E-64	1.41E-38
430	1.28E-73	6.07E-39	5.16E-61	6.07E-39
440	9.91E-72	4.03E-38	-2.81E-65	4.03E-38
450	8.54E-11	2.98E-16	5.16E-11	1.37E-10
460	2.50E-10	1.32E-15	1.46E-10	3.96E-10
470	6.69E-10	5.23E-15	3.97E-10	1.07E-09
480	1.71E-09	1.96E-14	1.03E-09	2.75E-09
490	4.30E-09	7.11E-14	2.59E-09	6.89E-09
500	1.05E-08	2.48E-13	6.26E-09	1.68E-08
510	2.43E-08	8.07E-13	1.46E-08	3.89E-08
520	5.49E-08	2.54E-12	3.30E-08	8.79E-08
530	1.20E-07	7.67E-12	7.23E-08	1.93E-07
540	2.56E-07	2.23E-11	1.54E-07	4.10E-07
550	5.30E-07	6.22E-11	3.19E-07	8.49E-07
560	1.07E-06	1.68E-10	6.44E-07	1.71E-06

Figure E-1. Heat flux to non-reacting flow from 320K to 560K in new mechanism with faster deactivation

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
570	2.10E-06	4.37E-10	1.27E-06	3.37E-06
580	4.04E-06	1.10E-09	2.44E-06	6.48E-06
590	7.60E-06	2.69E-09	4.59E-06	1.22E-05
600	1.40E-05	6.38E-09	8.45E-06	2.24E-05
610	2.52E-05	1.47E-08	1.52E-05	4.05E-05
620	4.46E-05	3.31E-08	2.70E-05	7.16E-05
630	7.72E-05	7.23E-08	4.69E-05	1.24E-04
640	1.32E-04	1.54E-07	7.99E-05	2.12E-04
650	2.20E-04	3.21E-07	1.34E-04	3.55E-04
660	3.62E-04	6.53E-07	2.21E-04	5.84E-04
670	5.87E-04	1.30E-06	3.59E-04	9.47E-04
680	9.40E-04	2.53E-06	5.73E-04	1.52E-03
690	1.46E-03	4.78E-06	9.02E-04	2.37E-03
700	2.26E-03	8.90E-06	1.40E-03	3.67E-03
710	3.43E-03	1.62E-05	2.14E-03	5.59E-03
720	5.13E-03	2.89E-05	3.24E-03	8.40E-03
730	7.57E-03	5.06E-05	4.83E-03	1.25E-02
740	1.19E-02	9.43E-05	7.15E-03	1.92E-02
750	1.57E-02	1.46E-04	1.03E-02	2.62E-02
760	2.69E-02	2.93E-04	1.50E-02	4.23E-02
770	3.09E-02	3.89E-04	2.09E-02	5.22E-02
780	4.20E-02	6.13E-04	2.91E-02	7.17E-02
790	6.20E-02	1.05E-03	4.02E-02	1.03E-01
800	7.44E-02	1.44E-03	5.36E-02	1.29E-01

Figure E-2. Heat flux to non-reacting flow from 570K to 800K in new mechanism with faster deactivation

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
820	1.24E-01	3.14E-03	9.23E-02	2.19E-01
830	1.53E-01	4.43E-03	1.18E-01	2.75E-01
840	1.84E-01	6.06E-03	1.47E-01	3.37E-01
850	2.19E-01	8.18E-03	1.80E-01	4.07E-01
860	2.59E-01	1.09E-02	2.16E-01	4.86E-01
870	2.92E-01	1.39E-02	2.53E-01	5.60E-01
880	3.23E-01	1.72E-02	2.90E-01	6.31E-01
890	3.51E-01	2.08E-02	3.26E-01	6.97E-01
900	3.70E-01	2.44E-02	3.58E-01	7.53E-01
910	3.85E-01	2.81E-02	3.88E-01	8.01E-01
920	3.95E-01	3.19E-02	4.14E-01	8.41E-01
930	4.00E-01	3.54E-02	4.37E-01	8.72E-01
940	3.99E-01	3.84E-02	4.56E-01	8.93E-01
950	3.96E-01	4.16E-02	4.73E-01	9.11E-01
960	3.90E-01	4.44E-02	4.86E-01	9.20E-01
970	3.82E-01	4.69E-02	5.00E-01	9.28E-01
980	3.71E-01	4.92E-02	5.10E-01	9.30E-01
990	3.62E-01	5.16E-02	5.19E-01	9.33E-01
1000	3.47E-01	5.30E-02	5.27E-01	9.27E-01
1010	3.39E-01	5.55E-02	5.37E-01	9.32E-01
1020	3.19E-01	5.56E-02	5.43E-01	9.17E-01
1030	3.03E-01	5.63E-02	5.49E-01	9.08E-01
1040	2.85E-01	5.62E-02	5.53E-01	8.94E-01
1050	2.65E-01	5.57E-02	5.55E-01	8.76E-01
1070	2.20E-01	5.20E-02	5.56E-01	8.29E-01
1100	1.56E-04	4.29E-05	9.10E-07	2.00E-04

Figure E-3. Heat flux to non-reacting flow from 820K to 1100K in new mechanism with faster deactivation

APPENDIX F
HEAT TRANSFER IN LOW IGNITION BRANCH IN NEW MECHANISM WITH FASTER
DISSOCIATION

Temp(K)	Heat Flux(w/m ²)	Temp(K)	Heat Flux(w/m ²)	Temp(K)	Heat Flux(w/m ²)
310	-1165.939258	410	-1562.834846	510	-1992.280801
315	-1187.352749	415	-1583.602155	515	-2014.540282
320	-1208.832788	420	-1604.443135	520	-2036.874016
325	-1230.380884	425	-1625.358084	525	-2059.281809
330	-1251.997502	430	-1646.347262	530	-2081.763454
335	-1273.682988	435	-1667.375653	535	-2104.31873
340	-1295.437579	440	-1688.510745	540	-2126.947403
345	-1317.258336	445	-1709.720624	545	-2149.649226
350	-1339.150342	450	-1731.00543	550	-2172.423942
355	-1361.111402	455	-1752.365273	555	-2195.271283
360	-1383.141395	460	-1773.800304	560	-2218.190969
365	-1405.240126	465	-1795.310435	565	-2241.182711
370	-1427.407332	470	-1816.895763	570	-2264.246212
375	-1449.642691	475	-1838.556289	575	-2287.381165
380	-1471.94578	480	-1860.291988	580	-2310.587255
385	-1467.03623	485	-1882.102816	585	-2333.864159
390	-1487.748976	490	-1903.988705	590	-2357.211547
395	-1500.971319	495	-1925.949565	595	-2380.629083
400	-1521.519832	500	-1947.985289	600	-2404.116424
405	-1542.140867	505	-1970.095749	605	-2425.484033

Figure F-1. Heat flux to non-reacting flow from 320K to 605K in low ignition branch in new mechanism

Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)
610	-2449.028653	710	-2932.374672	810	-3441.024578
615	-2472.639804	715	-2957.195024	815	-3466.878427
620	-2496.317076	720	-2982.070468	820	-3492.74078
625	-2520.060055	725	-3007.002773	825	-3518.648228
630	-2543.30831	730	-3031.991276	830	-3544.600271
635	-2567.161198	735	-3057.033567	835	-3570.596405
640	-2591.078096	740	-3082.130051	840	-3596.669956
645	-2615.058553	745	-3107.280205	845	-3622.76042
650	-2639.102115	750	-3132.483505	850	-3648.893882
655	-2663.208321	755	-3160.229326	855	-3675.75471
660	-2687.376705	760	-3185.511664	860	-3699.943906
665	-2711.606797	765	-3210.845055	865	-3726.224332
670	-2735.898121	770	-3236.229008	870	-3752.545294
675	-2760.250199	775	-3261.663034	875	-3778.90348
680	-2784.662546	780	-3287.146639	880	-3805.304602
685	-2809.134674	785	-3312.679329	885	-3831.335172
690	-2833.66609	790	-3338.260609	890	-3859.470425
695	-2858.256299	795	-3363.889981	895	-3885.699133
700	-2882.904802	800	-3389.566948	900	-3912.138569
705	-2907.611095	805	-3415.291011	905	-3939.039777

Figure F-2. Heat flux to non-reacting flow from 610K to 905K in low ignition branch in new mechanism

Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)
910	-3965.66743	1010	-4278.786741
915	-3992.723979	1015	-4266.648193
920	-4220.409772	1015.1	-403.6288151
925	-4250.793013	1015.2	-399.1762875
930	-4281.243155		
935	-4311.759011		
940	-3921.874484		
945	-3947.614279		
950	-3972.184322		
955	-3997.775504		
960	-4023.415985		
965	-4475.234838		
970	-4073.434362		
975	-4099.16741		
980	-4124.90561		
985	-4150.670629		
990	-4176.445691		
995	-4202.211554		
1000	-4227.930961		
1005	-4253.527985		

Figure F-3. Heat flux to non-reacting flow from 910K to 1015.2K in low ignition branch in new mechanism

Temp(K)	Heat Flux(w/m2)
916	-3998.060561
917	-4002.997316
918	-4210.155222
919	-4216.215106
920	-4220.409772
921	-4226.481021
922	-4232.554976
923	-4238.631633
924	-4244.710984
925	-4250.793013
926	-4256.877729
927	-4262.965105
928	-4269.055142
929	-4275.147828
930	-4281.243153
931	-4287.341107
932	-4293.44168
933	-4299.544864
934	-4305.650642
935	-4311.759011
936	-4317.869953
937	-3921.253404

Figure F-4. Heat flux to non-reacting flow from 916K to 937K in low ignition branch in new mechanism

APPENDIX G
HEAT TRANSFER IN HIGH IGNITION BRANCH IN NEW MECHANISM

Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)
616.5	-2412.702759	715	2200.02711	815	1911.540143
620	2379.793932	720	2178.424274	820	1894.787808
625	2357.160179	725	2165.932519	825	1878.726546
630	2390.921914	730	2153.453212	830	1859.94857
635	2348.85734	735	2141.013541	835	1843.824129
640	2337.962799	740	2127.163052	840	1838.700144
645	2329.042959	745	2114.260399	845	1811.240118
650	2321.873826	750	2100.478762	850	1794.95188
655	2314.582799	755	2086.471078	855	1780.119284
660	2306.043244	760	2074.103945	860	1777.732884
665	2300.077634	765	2061.288988	865	1771.942289
670	2288.34528	770	2044.640845	870	1722.428463
675	2279.000289	775	2032.106721	875	1715.578709
680	2269.280556	780	2014.705465	880	1695.701106
685	2259.191785	785	1999.874538	885	1678.535981
690	2248.92302	790	1986.648217	890	1661.176667
695	2238.663049	795	1973.098139	895	1646.959286
700	2225.709296	800	1957.091938	900	1626.483666
705	2214.313985	805	1948.468918	905	1612.231249
710	2202.626581	810	1927.211055	910	1587.656383

Figure G-1. Heat flux to non-reacting flow from 616.5K to 910K in high ignition branch in new mechanism

Temp(K)	Heat Flux(w/m2)	Temp(K)	Heat Flux(w/m2)
915	1574.150016	1015	1081.325427
920	1562.63597	1020	1067.728817
925	1538.001466	1025	1042.941011
930	1526.577299	1030	1028.355905
935	1425.027523	1035	1005.218626
940	1408.949175	1040	986.1594385
945	1387.762812	1045	976.1276954
950	1369.073284	1050	943.6638554
955	1353.302707	1055	924.741557
960	1330.074355	1060	905.7531531
965	1315.615682	1065	888.3059587
970	1314.279693	1070	866.6052324
975	1276.81722	1075	867.4950363
980	1262.677241	1080	1328.605957
985	1238.103332		
990	1216.482861		
995	1199.171953		
1000	1145.358354		
1005	1130.656676		
1010	1103.29645		

Figure G-2. Heat flux to non-reacting flow from 915K to 1080K in high ignition branch in new mechanism

APPENDIX H ENHANCEMENT OF HEAT TRANSFER

Table shows enhancement of heat transfer from surface reactions of $\text{OH(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + \text{O(s)} + \text{Pt(s)}$, $\text{H(s)} + \text{OH(s)} \Rightarrow \text{H}_2\text{O}^* + 2 \text{Pt(s)}$, and $\text{CO(s)} + \text{O(s)} \Rightarrow \text{CO}_2^* + 2 \text{Pt(s)}$

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
310	8.6691E-60	-3.0076E-29	1.4056E-43	-3.0076E-29
315	8.5543E-76	4.7207E-39	1.8708E-73	4.7207E-39
320	1.5244E-65	-1.4539E-35	-1.8106E-47	-1.4539E-35
325	5.5229E-73	8.7590E-38	4.9493E-59	8.7590E-38
330	1.6855E-74	-8.3394E-39	-7.5933E-60	-8.3394E-39
335	1.9482E-83	-4.2164E-43	-3.4839E-63	-4.2164E-43
340	1.0598E-78	7.5784E-41	1.3444E-71	7.5784E-41
345	2.0591E-73	-4.3317E-38	9.5032E-61	-4.3317E-38
350	4.2584E-75	3.9161E-39	-9.4646E-62	3.9161E-39
355	2.4845E-80	2.1555E-41	1.4187E-63	2.1555E-41
360	3.8870E-56	5.7634E-28	4.1907E-43	5.7634E-28
365	2.0958E-69	4.5969E-35	-6.6419E-54	4.5969E-35
370	8.6691E-60	-3.0076E-29	1.4056E-43	-3.0076E-29
375	8.6691E-60	-3.0076E-29	1.4056E-43	-3.0076E-29
380	1.3944E-75	-6.3838E-40	3.4630E-63	-6.3838E-40
385	9.1208E-76	4.3667E-40	-1.1040E-61	4.3667E-40
390	7.1774E-75	3.0328E-39	-2.4071E-65	3.0328E-39
395	3.8633E-82	-2.5291E-43	6.3203E-67	-2.5291E-43
400	1.7997E-21	1.0195E-29	-4.7471E-54	1.7997E-21
405	1.0542E-75	3.8940E-40	2.0561E-60	3.8940E-40
410	5.8505E-51	-1.0264E-25	-4.9228E-39	-1.0264E-25
415	2.0301E-80	1.6254E-42	7.6763E-65	1.6254E-42
420	4.5696E-82	1.8565E-43	-4.3027E-66	1.8565E-43
425	8.8436E-79	1.0513E-41	1.1933E-64	1.0513E-41
430	1.4708E-73	6.6232E-39	-2.8008E-60	6.6232E-39
435	3.4100E-73	6.6101E-39	3.2551E-64	6.6101E-39
440	5.5436E-79	4.3576E-42	-4.5524E-65	4.3576E-42
445	1.4949E-72	1.3278E-38	1.6923E-63	1.3278E-38
450	5.0033E-78	1.2193E-41	9.9405E-66	1.2193E-41
455	5.4462E-71	8.3975E-38	-6.0245E-63	8.3975E-38

Figure H-1. Enhancement of heat transfer from 310K to 455K in low ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
460	1.4537E-76	6.7210E-41	8.5180E-65	6.7210E-41
465	5.6149E-68	3.0444E-36	-2.5182E-58	3.0444E-36
470	1.0475E-73	2.0602E-39	2.5829E-64	2.0602E-39
475	1.6929E-64	2.5612E-34	-2.0964E-58	2.5612E-34
480	1.1039E-64	1.9259E-34	-1.2427E-57	1.9259E-34
485	2.4475E-67	-1.9665E-35	7.6411E-52	-1.9665E-35
490	9.4721E-48	-8.7944E-25	2.4349E-37	-8.7944E-25
495	1.0782E-63	5.1118E-34	-1.2010E-58	5.1118E-34
500	1.0000E-74	4.0714E-40	4.8141E-60	4.0714E-40
505	1.0821E-65	3.1694E-35	7.0733E-60	3.1694E-35
510	4.4857E-74	7.5685E-40	1.7350E-60	7.5685E-40
515	2.1799E-67	3.2499E-36	1.4578E-58	3.2499E-36
520	7.5880E-73	2.9698E-39	-1.9987E-60	2.9698E-39
525	4.7666E-68	1.2100E-36	-1.9576E-59	1.2100E-36
530	1.3058E-71	1.2125E-38	1.6663E-60	1.2125E-38
535	3.7877E-68	8.9077E-37	2.7403E-59	8.9077E-37
540	3.9598E-70	6.5917E-38	-1.0468E-59	6.5917E-38
545	5.6736E-68	9.3871E-37	6.3672E-59	9.3871E-37
550	9.1899E-69	3.0817E-37	5.4061E-60	3.0817E-37
555	1.7243E-67	1.4410E-36	9.9954E-60	1.4410E-36
560	1.7886E-67	1.3049E-36	-1.6889E-59	1.3049E-36
565	1.1202E-66	3.3141E-36	-3.2656E-60	3.3141E-36
570	3.3129E-66	5.4143E-36	1.5977E-58	5.4143E-36
575	1.1869E-65	9.8240E-36	-1.4396E-59	9.8240E-36
580	6.1118E-65	2.2210E-35	1.2011E-58	2.2210E-35
585	1.9680E-64	3.7051E-35	2.2160E-59	3.7051E-35
590	1.1261E-63	9.2103E-35	-5.2343E-59	9.2103E-35
595	3.2179E-63	1.4125E-34	1.5545E-58	1.4125E-34
600	1.7393E-62	3.4180E-34	-2.4160E-59	3.4180E-34
605	4.9843E-62	5.2075E-34	3.4309E-59	5.2075E-34

Figure H-2. Enhancement of heat transfer from 460K to 605K in low ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
610	2.6835E-61	1.2793E-33	7.1072E-58	1.2793E-33
615	7.4929E-61	1.9298E-33	-5.0138E-58	1.9298E-33
620	3.8382E-60	4.6101E-33	2.3633E-57	4.6101E-33
625	1.0530E-59	6.8496E-33	-1.6168E-58	6.8496E-33
630	8.2198E-59	1.9643E-32	-1.0383E-58	1.9643E-32
635	2.1954E-58	2.9092E-32	-1.2684E-58	2.9092E-32
640	7.6834E-58	5.2949E-32	-4.3522E-59	5.2949E-32
645	2.5021E-57	9.1492E-32	-4.9515E-59	9.1492E-32
650	8.7518E-57	1.6959E-31	-5.9299E-59	1.6959E-31
655	2.6766E-56	2.8359E-31	-8.5574E-59	2.8359E-31
660	9.1562E-56	5.1653E-31	1.2562E-58	5.1653E-31
665	2.6897E-55	8.4722E-31	4.5574E-58	8.4722E-31
670	9.3116E-55	1.5762E-30	-5.3549E-58	1.5762E-30
675	1.6435E-52	3.7691E-29	-8.5451E-55	3.7691E-29
680	8.6752E-54	4.5408E-30	-3.2327E-57	4.5408E-30
685	1.3690E-52	2.3885E-29	-1.7006E-56	2.3885E-29
690	7.6056E-53	1.2885E-29	-4.2016E-57	1.2885E-29
695	4.7360E-52	3.6455E-29	-6.2389E-56	3.6455E-29
700	6.4437E-52	3.5964E-29	-1.4358E-56	3.5964E-29
705	2.5938E-51	7.5426E-29	-1.4940E-55	7.5426E-29
710	5.0680E-51	9.6029E-29	-1.1120E-55	9.6029E-29
715	1.6047E-50	1.7211E-28	-3.4166E-55	1.7211E-28
720	3.5602E-50	2.2377E-28	-4.2411E-55	2.2377E-28
725	1.1785E-49	4.1445E-28	-1.5126E-54	4.1445E-28
730	2.8171E-49	6.4106E-28	-3.4886E-54	6.4106E-28
735	6.4611E-49	9.3121E-28	-6.3540E-54	9.3121E-28
740	2.7350E-48	2.0486E-27	-3.7401E-53	2.0486E-27
745	4.8051E-48	2.4841E-27	-4.5703E-53	2.4841E-27
750	3.3993E-47	7.6275E-27	-6.4809E-52	7.6275E-27
755	4.2359E-47	7.4242E-27	-4.7228E-52	7.4242E-27

Figure H-3. Enhancement of heat transfer from 610K to 755K in low ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
760	6.3704E-46	3.6845E-26	-2.2302E-50	3.6845E-26
765	4.9404E-46	2.7091E-26	-7.8871E-51	2.7091E-26
770	1.4877E-44	1.9988E-25	-1.1072E-48	1.9988E-25
775	8.3137E-45	1.2263E-25	-2.2990E-49	1.2263E-25
780	3.8242E-43	1.1540E-24	-6.8114E-47	1.1540E-24
785	1.8317E-43	6.4734E-25	-1.0064E-47	6.4734E-25
790	9.3760E-42	6.5381E-24	-4.2861E-45	6.5381E-24
795	4.2487E-42	3.5423E-24	-5.0500E-46	3.5423E-24
800	2.0320E-40	3.4999E-23	-2.3581E-43	3.4999E-23
805	9.1105E-41	1.8572E-23	-2.3950E-44	1.8572E-23
810	1.8362E-38	2.4777E-22	-3.2110E-39	2.4777E-22
815	1.6596E-39	8.9949E-23	-9.6948E-43	8.9949E-23
820	6.6632E-39	1.9132E-22	-5.6747E-42	1.9132E-22
825	2.4518E-23	-5.8383E-17	1.6229E-38	-5.8383E-17
830	9.2221E-38	7.9809E-22	-1.5623E-40	7.9809E-22
835	8.4617E-25	-1.4364E-17	1.8884E-40	-1.4364E-17
840	1.6032E-36	3.2460E-21	-3.4756E-39	3.2460E-21
845	1.4842E-23	-7.7638E-17	1.9905E-38	-7.7638E-17
850	1.2084E-35	9.2823E-21	-3.2483E-38	9.2823E-21
855	4.7028E-24	-5.6212E-17	2.6124E-38	-5.6212E-17
860	2.6350E-34	4.4293E-20	-4.8996E-37	4.4293E-20
865	5.6809E-22	-7.8387E-16	2.9319E-35	-7.8387E-16
870	1.2716E-33	9.5021E-20	-2.1609E-36	9.5021E-20
875	3.4978E-34	3.1571E-20	-1.3450E-37	3.1571E-20
880	5.7256E-33	1.6444E-19	-5.6627E-36	1.6444E-19
885	5.9495E-33	1.2649E-19	-3.0345E-36	1.2649E-19
890	2.0297E-32	2.1244E-19	-9.9748E-36	2.1244E-19
895	1.3425E-32	1.6750E-19	-5.9245E-36	1.6750E-19
900	1.5763E-32	1.6911E-19	-6.4559E-36	1.6911E-19
905	1.9268E-32	1.7398E-19	-7.7050E-36	1.7398E-19

Figure H-4. Enhancement of heat transfer from 610K to 905K in low ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
910	2.4718E-32	1.6838E-19	-8.1298E-36	1.6838E-19
915	4.4618E-36	1.0078E-21	-3.4522E-40	1.0078E-21
920	1.7005E-32	1.3726E-19	-6.5007E-36	1.3726E-19
925	1.0326E-30	1.3801E-18	-7.6407E-34	1.3801E-18
930	4.7266E-20	-3.0002E-14	6.2667E-33	-3.0002E-14
935	3.3823E-33	4.2613E-20	-9.4707E-37	4.2613E-20
940	3.4520E-31	3.3674E-19	-6.1670E-35	3.3674E-19
945	1.4673E-19	-7.1680E-14	-8.2692E-32	-7.1680E-14
950	5.8825E-30	2.1264E-18	-2.9080E-33	2.1264E-18
955	9.1481E-29	9.8545E-18	-6.9033E-32	9.8545E-18
960	3.6543E-29	4.4120E-18	-1.7466E-32	4.4120E-18
965	3.0945E-31	2.8194E-19	-1.0149E-34	2.8194E-19
970	7.6023E-29	4.3956E-18	-2.8857E-32	4.3956E-18
975	6.7236E-25	4.1483E-16	-2.4810E-28	4.1483E-16
980	3.3903E-29	2.0551E-18	-1.2002E-32	2.0551E-18
985	3.3516E-27	3.9708E-17	-2.6955E-30	3.9708E-17
990	4.3130E-31	1.9936E-19	-1.7951E-34	1.9936E-19
995	1.2015E-19	-3.8757E-13	5.3715E-28	-3.8757E-13
1000	1.4228E-29	-1.2532E-18	-5.9999E-33	-1.2532E-18
1005	2.7439E-31	2.1045E-19	-5.3885E-34	2.1045E-19
1010	1.9568E-36	-5.1610E-23	-7.3668E-40	-5.1610E-23
1015	2.6303E-39	9.6273E-24	-3.3632E-42	9.6273E-24
1015.1	1.9199E-41	1.7891E-25	-2.6000E-47	1.7891E-25
1015.2	2.1671E-42	-2.5621E-25	-7.2467E-51	-2.5621E-25

Figure H-5. Enhancement of heat transfer from 910K to 1015K in low ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
616.5	2.1559E-10	5.0332E-08	1.0268E-05	1.0319E-05
620	2.5551E-01	1.2558E-02	8.0199E+00	8.2880E+00
625	2.5434E-01	1.0884E-02	7.8409E+00	8.1061E+00
630	2.7326E-01	1.1014E-02	7.8874E+00	8.1717E+00
635	2.6683E-01	1.0174E-02	7.8631E+00	8.1401E+00
640	2.6826E-01	9.8256E-03	7.8061E+00	8.0842E+00
645	2.6609E-01	9.4042E-03	7.6222E+00	7.8977E+00
650	2.7209E-01	9.4194E-03	7.6963E+00	7.9778E+00
655	2.7008E-01	9.1233E-03	7.5141E+00	7.7933E+00
660	2.7553E-01	9.1800E-03	7.5866E+00	7.8713E+00
665	2.7786E-01	9.1264E-03	7.5345E+00	7.8215E+00
670	2.7899E-01	9.0702E-03	7.5073E+00	7.7954E+00
675	2.7593E-01	8.8218E-03	7.3092E+00	7.5939E+00
680	2.8110E-01	8.9268E-03	7.3755E+00	7.6656E+00
685	2.7832E-01	8.7429E-03	7.2063E+00	7.4933E+00
690	2.8334E-01	8.8600E-03	7.2700E+00	7.5622E+00
695	2.8452E-01	8.8424E-03	7.2157E+00	7.5091E+00
700	2.8527E-01	8.8187E-03	7.1632E+00	7.4573E+00
705	2.8229E-01	8.6708E-03	7.0014E+00	7.2923E+00
710	2.8711E-01	8.8093E-03	7.0597E+00	7.3557E+00
715	2.9004E-01	8.8794E-03	6.9931E+00	7.2921E+00
720	2.8880E-01	8.8278E-03	6.9571E+00	7.2547E+00
725	2.8570E-01	8.7130E-03	6.8019E+00	7.0963E+00
730	2.9036E-01	8.8781E-03	6.8551E+00	7.1543E+00
735	2.9123E-01	8.9217E-03	6.8034E+00	7.1035E+00
740	2.9177E-01	8.9657E-03	6.7534E+00	7.0541E+00
745	2.9261E-01	9.0323E-03	6.7021E+00	7.0038E+00
750	2.9313E-01	9.1020E-03	6.6511E+00	6.9533E+00
755	2.8980E-01	9.0535E-03	6.5048E+00	6.8037E+00
760	2.9480E-01	9.3094E-03	6.5486E+00	6.8527E+00

Figure H-6. Enhancement of heat transfer from 616.5K to 760K in high ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
765	2.9579E-01	9.4465E-03	6.4986E+00	6.8038E+00
770	2.9568E-01	9.5627E-03	6.4474E+00	6.7527E+00
775	2.9666E-01	9.7399E-03	6.3953E+00	6.7017E+00
780	2.9650E-01	9.8992E-03	6.3452E+00	6.6516E+00
785	2.9318E-01	9.9675E-03	6.2050E+00	6.5081E+00
790	2.9786E-01	1.0354E-02	6.2404E+00	6.5486E+00
795	2.9896E-01	1.0638E-02	6.1861E+00	6.4957E+00
800	2.9893E-01	1.0908E-02	6.1324E+00	6.4423E+00
805	3.0129E-01	1.1305E-02	6.0768E+00	6.3894E+00
810	2.9989E-01	1.1580E-02	6.0285E+00	6.3399E+00
815	2.9983E-01	1.1940E-02	5.9744E+00	6.2862E+00
820	2.9976E-01	1.2331E-02	5.9208E+00	6.2329E+00
825	2.9967E-01	1.2753E-02	5.8665E+00	6.1790E+00
830	2.9866E-01	1.3163E-02	5.8140E+00	6.1258E+00
835	2.9479E-01	1.3470E-02	5.6822E+00	5.9904E+00
840	3.0117E-01	1.4302E-02	5.6962E+00	6.0117E+00
845	2.9401E-01	1.4491E-02	5.5728E+00	5.8813E+00
850	2.9705E-01	1.5226E-02	5.5903E+00	5.9026E+00
855	2.9637E-01	1.5803E-02	5.5350E+00	5.8472E+00
860	3.0130E-01	1.6741E-02	5.5007E+00	5.8188E+00
865	3.0161E-01	1.7456E-02	5.3982E+00	5.7173E+00
870	2.9186E-01	1.7548E-02	5.3635E+00	5.6729E+00
875	2.9296E-01	1.8354E-02	5.2976E+00	5.6089E+00
880	2.9089E-01	1.8974E-02	5.2406E+00	5.5505E+00
885	2.8950E-01	1.9663E-02	5.1817E+00	5.4909E+00
890	2.8832E-01	2.0390E-02	5.1188E+00	5.4276E+00
895	2.8702E-01	2.1149E-02	5.0487E+00	5.3569E+00
900	2.8410E-01	2.1764E-02	4.9939E+00	5.2997E+00
905	2.8348E-01	2.2602E-02	4.9276E+00	5.2337E+00
910	2.7979E-01	2.3192E-02	4.8702E+00	5.1732E+00

Figure H-7. Enhancement of heat transfer from 765K to 910K in high ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
915	2.7834E-01	2.3993E-02	4.8052E+00	5.1075E+00
920	2.7779E-01	2.4899E-02	4.7369E+00	5.0396E+00
925	2.7375E-01	2.5483E-02	4.6770E+00	4.9762E+00
930	2.7309E-01	2.6414E-02	4.6063E+00	4.9058E+00
935	2.7070E-01	2.7188E-02	4.5643E+00	4.8622E+00
940	2.6887E-01	2.8027E-02	4.4953E+00	4.7922E+00
945	2.6519E-01	2.8668E-02	4.4324E+00	4.7263E+00
950	2.6245E-01	2.9418E-02	4.3648E+00	4.6566E+00
955	2.6052E-01	3.0274E-02	4.2915E+00	4.5823E+00
960	2.5684E-01	3.0913E-02	4.2280E+00	4.5157E+00
965	2.5463E-01	3.1745E-02	4.1542E+00	4.4406E+00
970	2.5526E-01	3.2975E-02	4.0663E+00	4.3545E+00
975	2.4812E-01	3.3129E-02	4.0146E+00	4.2958E+00
980	2.4598E-01	3.3979E-02	3.9396E+00	4.2195E+00
985	2.3886E-01	3.4102E-02	3.8259E+00	4.0988E+00

Figure H-8. Enhancement of heat transfer from 915K to 985K in high ignition branch in new mechanism

Temp(K)	OH(s)+OH(s)	H(s)+OH(s)	CO(s)+O(s)	Sum(KW/m2)
990	2.3741E-01	3.5024E-02	3.8004E+00	4.0729E+00
995	2.3183E-01	3.5336E-02	3.6815E+00	3.9486E+00
1000	2.3258E-01	3.6611E-02	3.6573E+00	3.9264E+00
1005	2.3107E-01	3.7559E-02	3.5880E+00	3.8566E+00
1010	2.2487E-01	3.7696E-02	3.5108E+00	3.7733E+00
1015	2.1961E-01	3.7969E-02	3.4302E+00	3.6878E+00
1020	2.1741E-01	3.8771E-02	3.3526E+00	3.6088E+00
1025	2.1022E-01	3.8633E-02	3.2416E+00	3.4904E+00
1030	2.0903E-01	3.9593E-02	3.1958E+00	3.4444E+00
1035	2.0204E-01	3.9412E-02	3.0835E+00	3.3249E+00
1040	1.9969E-01	4.0115E-02	3.0361E+00	3.2759E+00
1045	1.9690E-01	4.0740E-02	2.9463E+00	3.1839E+00
1050	1.9077E-01	4.0604E-02	2.8672E+00	3.0986E+00
1055	1.8434E-01	4.0367E-02	2.7496E+00	2.9743E+00
1060	1.8144E-01	4.0867E-02	2.6938E+00	2.9161E+00
1065	1.7649E-01	4.0877E-02	2.5970E+00	2.8143E+00
1070	1.7162E-01	4.0866E-02	2.5007E+00	2.7132E+00
1075	1.6938E-01	4.1491E-02	2.3865E+00	2.5973E+00
1080	6.3674E-04	1.5687E-04	7.2930E-05	8.6654E-04

Figure H-9. Enhancement of heat transfer from 990K to 1080K in high ignition branch in new mechanism

LIST OF REFERENCES

- [1] Deutschmann, O., Behrendt, F., and Warnatz, J., "Formal treatment of catalytic combustion and catalytic conversion of methane", *Catalysis Today*, Vol 46, 1998, pp.155-163
- [2] Deutschmann, O., Behrendt, F., and Warnatz J., "Modelling and simulation of heterogeneous oxidation of methane on a platinum foil", *Catalysis Today*, Vol 21, 1994, pp.461-470
- [3] Deutschmann, O., Maier, L.I., Riedel, U., Stroemman, A.H., and Dibble, R.W, "Hydrogen assisted catalytic combustion of methane on platinum", *Catalysis Today*, Vol 59, 2000, pp.141-150
- [4] Moallemi, F., Batley, G., Dupont, V., Foster, T.J., Pourkashanian, M., and Williams, A., "Chemical modelling and measurements of the catalytic combustion of CH₄/air mixture on platinum and palladium catalysts", *Catalysis Today*, Vol 47, 1999, pp.235-244
- [5] Henry, C.A., Mikolaitis, W., Szedlacsek, P., and Hahn, D.W., "Investigation of Heat Transfer under Catalytic Combustion of Methane: Effect of Heterogeneous Chemistry on Heat Transfer Enhancement"
- [6] Williams, W.R., Stenzel, M.T., sing, X. and Schmidt, L.D., "Bifurcation behavior in homogeneous-heterogeneous combustion: I. Experimental results over platinum" *Combust and Flame*, Vol 84, 1991, pp.277-291
- [7] Kuo, K.K.(2005), *Principles of Combustion, Second Edition*, Hoboken, New Jersey, John Wiley & Sons
- [8] McCash, E.M.(2001), *Surface Chemistry*, Oxford, New York, Oxford University Press
- [9] Hayes, R.E. and Kolaczkowski, S.T.(1997), *Introduction to Catalytic Combustion*, Amsterdam, Netherlands, Gordon and Breach Science
- [10] Frenklach, M., Bowman, T., Smith, G., and Gardiner, B., "GRI Mech", 2000, Gas Research Institute, <http://www.me.berkeley.edu/gri-mech/>, July 2011
- [11] Deutschmann, O., "Surface Mechanism of the CH₄-O₂ Reactions on Platinum", 1995, Heidelberg University, <http://www.detchem.com/mechanisms.html>, July 2011
- [12] Wikipedia, "Methane", Wikipedia, <http://en.wikipedia.org/wiki/Methane>, July 2011
- [13] Wikipedia, "Catalysis", Wikipedia, <http://en.wikipedia.org/wiki/Catalysis>, July 2011

[14] Martin, K.K., and Rachul, L., "NASA Glenn Tests Alternative Green Rocket Engine", 2010, http://www.nasa.gov/centers/glenn/news/pressrel/2010/10-049_green.html, July 2011

[15] Goodwin, D.G., "Defining Phases and Interfaces, Cantera 1.5", Division of Engineering and Applied Science, California Institute of Technology, August 14, 2003

[16] Goodwin, D.G., "Cantera User's Guide Fortran Version Release 1.2", Division of Engineering and Applied Science, California Institute of Technology, November, 2001

[17] Goodwin, D.G., "CANTERA", Division of Engineering and Applied Science, California Institute of Technology

BIOGRAPHICAL SKETCH

Sungsik Kim was born in Seoul, Korea in 1983. He majored in nuclear engineering Kyung Hee University and received a Bachelor of Engineering degree in nuclear engineering in 2008. He switched to mechanical and aerospace engineering in 2009 when he attended University of Florida, Gainesville. He received a Master of Science degree in mechanical and aerospace engineering in 2011. His research interests are catalytic combustion, combustion process and emission control, and alternative fuel engines.