UCLA
UCLA Electronic Theses and Dissertations

Title
Simulation of Reforming Reactor Tube: Quantifying Catalyst Pellet's Effectiveness Factor

Permalink
https://escholarship.org/uc/item/57q4d400

Author
Da Cruz, Flavio Eduardo

Publication Date
2016

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Simulation of Reforming Reactor Tube:

Quantifying Catalyst Pellet's Effectiveness Factor

A thesis submitted in partial satisfaction

of the requirements for de degree of Master of Science

in Chemical Engineering

by

Flavio Eduardo Da Cruz

2016

ABSTRACT OF THE THESIS

Simulation of a Reforming Reactor Tube:

Quantifying Catalyst Pellet's Effectiveness Factor

by

Flavio Eduardo Da Cruz

Master of Science in Chemical Engineering

University of California, Los Angeles, 2016

Professor Vasilios I. Manousiouthakis, Chair

In this work, a consistent mathematical model to simulate a spherical catalytic pellet and a Packed-Bed Reactor (PBR) is develop. The Dusty Gas Model (DGM) is applied to the calculation of the diffusive fluxes in the porous media. Simulations are executed considering hydrogen production from steam methane reforming. Species' diffusivities are calculated using data from literature as well as the values for tortuosity and porosity. The pellet simulation is performed considering mass, species, momentum and energy conservation equations, while the PBR model considers constant temperature along the reactor. The simulations are performed at a variety of temperatures, and results are presented in terms of species' production/consumption along the pellet radius and reactor length. The effectiveness factor for each species along the reactor length is determined and results are discussed.

The thesis of Flavio Eduardo Da Cruz is approved.

Selim M. Senkan

Dante A. Simonetti

Vasilios I. Manousiouthakis, Committee Chair

University of California, Los Angeles

2016

DEDICATION

This thesis work is dedicated to my beloved wife, Rita, who has been a constant source of encouragement during the challenges of graduate school and life, and to my mother, Piedade, whose examples have taught me to work hard for the things that I aspire to achieve.

TABLE OF CONTENTS

1. Introduction 1

1.1. Proposed work and literature review 3

2. Conceptual framework and solution approach 7

3. Packed-bed reactor model 10

3.1. Structural parameters and porous media geometry 10

3.2. Reynolds Transport Theorem - RTT 14

3.3. Total mass conservation 18

3.4. Component mass conservation 20

3.5. Diffusion models for mass transfer in the pellet and the reactor domains 23

4. Simulation I – Catalytic spherical pellet 42

4.1. Modeling approach 42

4.2. Model equations 43

4.3. Boundary conditions 49

4.4. Simulation parameters 51

4.5. Results and discussion – Simulation I – Catalytic pellet 55

5. Simulation II – Packed-bed reactor 66

5.1. Modeling approach 66

5.2. Model equations 67

5.3. Boundary conditions 72

5.4. Effectiveness Factor 75

5.5. Simulation parameters 76

5.6. Results and discussion – Simulation II – Packed-bed reactor 76

6. Conclusions 84

7. Appendix 85

7.1. Appendix A – Model Parameters 85

7.2. Appendix B – Simulation codes 92

8. References 224

LIST OF FIGURES

Figure 1 - Pellet temperature as a function of the pellet radius, for different T_bulk 56

Figure 2- Pellet pressure as a function of the pellet radius, for different T_bulk 57

Figure 3 - Species molar fractions as a function of the pellet radius (T_bulk=700K)..... 58

Figure 4 - Species molar fractions as a function of the pellet radius (T_bulk=1000K)..... 58

Figure 5 - Species molar fractions as a function of the pellet radius (T_bulk=1200K)..... 59

Figure 6 - Comparison between variable T and constant T along the pellet in terms of species' production..... 60

Figure 7 - Species concentrations as a function of the pellet radius (T_bulk=1000K)..... 61

Figure 8 - Species concentrations as a function of the pellet radius (T_bulk=1200K)..... 61

Figure 9 - Species mass fractions as a function of the pellet radius (T_bulk=1000K)..... 62

Figure 10 - Species mass fractions as a function of the pellet radius (T_bulk=1200K)..... 62

Figure 11 - Species mass flux (r-direction) as a function of the pellet radius (T_bulk=1000K).. 63

Figure 12 - Species mass flux (r-direction) as a function of the pellet radius (T_bulk=1200K).. 63

Figure 13 - Species mass flux (in the r-direction) as a function of the pellet radius (4.5mm<r_pellet < 5mm) (T_bulk=1000K)..... 64

Figure 14 - Species mass flux (in the r-direction) as a function of the pellet radius (4.5mm<r_pellet < 5mm) (T_bulk=1200K)..... 65

Figure 15 - Coupled reactor and pellet model approach..... 67

Figure 16 - Species mass fractions as a function of the reactor's length, for T=1100K 77

Figure 17 - Species mass fraction as a function of the reactor's length, for T=1100K..... 77

Figure 18 - CH4 molar fractions x reactor length (xz-plan@y=0) and CH4 molar fraction x pellet radius (yz-plan) along the reactor (x-axis) 78

Figure 19 - CO molar fractions x reactor length (xz-plan@y=0) and CO molar fraction x pellet radius (yz-plan) along the reactor (x-axis) 79

Figure 20 - CO2 molar fractions x reactor length (xz-plan@y=0) and CO2 molar fraction x pellet radius (yz-plan) along the reactor (x-axis) 79

Figure 21 - H2 molar fractions x reactor length (xz-plan@y=0) and H2 molar fraction x pellet radius (yz-plan) along the reactor (x-axis) 80

Figure 22 - H2O molar fractions x reactor length (xz-plan@y=0) and H2O molar fraction x pellet radius (yz-plan) along the reactor (x-axis) 80

Figure 23 - Effectiveness factor variation along the reactor lenght at T=700K 81

Figure 24 - Effectiveness factor variation along the reactor lenght at T=900K 82

Figure 25 - Effectiveness factor variation along the reactor lenght at T=1100K 82

LIST OF TABLES

Table 1: Pore size distribution of catalyst…….....51

Table 2: Estimated geometrical parameters for the catalyst pellet……53

Table 3: Parameters of the rate coefficients and adsorption constants for related Arrhenius or Van't Hoff Equations……..55

Table 4: Reaction equilibrium constants……55

Table 5: Parameters for the coupled PRF/Pellet simulation……..76

LIST OF SYMBOLS

Latin symbols

A	Area, cm^2
A_p	Pellet outside area, cm^2
B_O	Viscous flow parameter of the fluid, cm^2
c_i	Concentration of species i in the mixture, mol/cm^3
c_i^p	Concentration of species i per volume of pellet (solid+voids), mol/cm^3
c_i^r	Concentration of species i per volume of reactor (solid+voids), mol/cm^3
c_{Tot}	Total concentration of the mixture, mol/cm^3
$C_{P,i}$	Constant pressure heat capacity of species i, erg/K
C_V	Constant volume heat capacity, erg/K
d_{pellet}	Diameter of the catalytic pellet, cm
d_{pore}	Mean pore diameter, cm
D_{iK}	Knudsen diffusion coefficient of species i, cm^2/s
D_{ij}	Binary diffusion coefficient in the free space, cm^2/s
D_{ij}^{eff}	Effective Binary diffusion coefficient in the porous media, cm^2/s
D_i^T	Thermal diffusion coefficient for Maxwell-Stefan equation, cm^2/s

E	Energy of the system, erg
$\vec{F}_i$	Forces acting on species i per volume of pellet, N/cm^3
$\vec{g}$	Gravitational acceleration, m/s^2
$\tilde{h}_{f,i}$	Specific molar enthalpy of species i in the fluid phase, erg/g
h_{pore}	Length of a cylindrical pore in the catalyst pellet (estimate), erg/g
$\vec{j}_i$	Diffusive mass flux of species i per area of single phase material, g/(cm$^2 \cdot$s)
$\vec{j}_i^p$	Diffusive mass flux of species i per area of pellet (solid+voids), g/(cm$^2 \cdot$s)
$\vec{j}_i^r$	Diffusive mass flux of species i per area of reactor (solid+voids), g/(cm$^2 \cdot$s)
k_B	Boltzmann constant, (cm$^2 \cdot$g)/(s$^2 \cdot$K)
k_f	Thermal conductivity of the fluid phase, erg/(s$\cdot$cm$\cdot$K)
k_s	Thermal conductivity of the solid phase, erg/(s$\cdot$cm$\cdot$K)
L^{Pore}	Length of a pore in the catalyst pellet, cm
M	Intensive property of a thermodynamic system
M_i	Molecular weight of species i, g/mol
M^t_{CSys}	Extensive property of a composite thermodynamic system
M_{AB}	Reduced molecular weight of the species' pair A and B, g/mol
$\vec{n}_i$	Total mass flux of species i per area of single phase material, g/(cm$^2 \cdot$s)

$\overrightarrow{n_i^p}$	Total mass flux of species i per area of pellet (solid+voids), $\mathrm{g/(cm^2 \cdot s)}$
$\overrightarrow{n_i^r}$	Total mass flux of species i per area of reactor (solid+voids), $\mathrm{g/(cm^2 \cdot s)}$
$\vec{n}$	Unity vector normal to the surface area, dimensionless
$\overrightarrow{N_i}$	Total molar flux of species i per area of single phase material, $\mathrm{mol/(cm^2 \cdot s)}$
$\overrightarrow{N_i^p}$	Total molar flux of species i per area of pellet (solid+voids), $\mathrm{mol/(cm^2 \cdot s)}$
N_{holes}	Number of holes in the pellet surface (estimate), dimensionless
p	Pressure, Ba
Q	Heat absorbed by the system, erg
$q_{f \to s}$	Convective heat flux from the fluid to the solid phase, $\mathrm{erg/(s \cdot m^2)}$
$\overrightarrow{q_f}$	Conductive heat flux in the fluid phase, $\mathrm{erg/(s \cdot m^2)}$
$\overrightarrow{q_s}$	Conductive heat flux in the solid phase, $\mathrm{erg/(s \cdot m^2)}$
r	Spherical pellet radial variable, cm
r_{pellet}	Radius for the spherical pellet, cm
R_i	Mass reaction rate of species i per volume of pellet, $\mathrm{g/(cm^3 \cdot s)}$
$\overline{R}_i$	Molar reaction rate of species i per mass of catalyst, $\mathrm{mol/[(g)_{cat} \cdot s]}$
$\tilde{R}$	Ideal gas constant, $\mathrm{J/mol \cdot K}$
r_{pore}	Radius of a pore in the catalyst pellet, cm
T	Temperature, K

T_{AB}^{*}	Reduced temperature of the A-B species pair, dimensionless
T_i^b	Normal boiling point of species i at 1atm, K
T_f	Temperature of the fluid phase, K
T_s	Temperature of the solid phase, K
$\vec{\vec{T}}'$, T_f'	Fluid stress tensor, Ba
U	Specific internal energy of the system, erg/g
$\vec{v}_i$	Velocity of species i in the mixture, cm/s
$\vec{v}_f$	Mass velocity of the fluid phase, cm/s
$\vec{v}_s$	Mass velocity of the solid phase, cm/s
V_i^b	Liquid molar volume of species i at the normal boiling point, cm^3/mol
V^{Pellet}	Total volume of the pellet (solid+voids), cm^3
V_{Solid}^{Pellet}	Volume of the solid phase in the catalyst pellet, cm^3
V_{Voids}^{Pellet}	Volume of the voids in the catalyst pellet, cm^3
$V^{Reactor}$	Total volume of the reactor (solid+voids), cm^3
$V_{Solid}^{Reactor}$	Volume of the solid phase in the reactor, cm^3
$V_{Voids}^{Reactor}$	Volume of the voids in the reactor, cm^3
W	Work done by the system, erg

w_i	Mass fraction rate of species i in the mixture, dimensionless
$w_{f \to s}$	Work done on the solid phase by the fluid, $\mathrm{erg/(s \cdot m^2)}$
x_i	Molar fraction rate of species i in the mixture, dimensionless

Greek symbols

$\left(\alpha'_{ij}\right)_{tr}$	Thermal diffusion coefficient, $\mathrm{mol/cm^3}$
$\left(\alpha'_{id}\right)_{tr}$	Thermal transpiration coefficient, $\mathrm{mol/cm^3}$
Δ'_{id}	Correction term of Knudsen Diffusivity, dimensionless
ε_V	Pellet's volumetric porosity, dimensionless
ε_A	Pellet's area porosity, dimensionless
ε_I	Pellet's interfacial factor, 1/cm
$\hat{\varepsilon}_A$	Lennard-Jones characteristic energy for pure species A, $(\mathrm{cm^2 \cdot g})/(\mathrm{s^2 \cdot K^2})$
λ	Convective heat transfer coefficient, $\mathrm{erg/(s \cdot cm^2 \cdot K)}$
μ_i	Dynamic viscosity of pure species i, $\mathrm{g/(cm \cdot s)}$
$\hat{\mu}_i$	Dipole moment of species i, D
μ_f	Dynamic viscosity of the fluid phase, $\mathrm{g/(cm \cdot s)}$
ρ	Mass density, $\mathrm{g/cm^3}$
ρ_i	Mass density of species i, $\mathrm{g/cm^3}$

ρ_f	Mass density of the fluid phase, g/cm^3
ρ_s	Mass density of the solid phase (catalyst), g/cm^3
σ_{AB}	Reduced collision diameter, Å
τ	Pellet's tortuosity factor, dimensionless
υ	Total of species presented in the system, dimensionless
ϕ_V	Reactor's volumetric void fraction, dimensionless
ϕ_A	Reactor's area void fraction, dimensionless
ϕ_I	Reactor's area to volume interface factor, 1/cm
Φ_{ij}	Correction function for the mixture's viscosity calculation, dimensionless
Ω_{AB}^{D}	Lennard-Jones collision integral, dimensionless

Subscripts

f	Fluid phase
i	Component i in the mixture
j	Component j in the mixture
p	Pellet domain
r	Reactor domain
s	solid phase

ACKNOWLEDGMENTS

I would like to thank the UCLA Chemical and Biomolecular Engineering Department for providing me the opportunity to be a teaching assistant. I would also like to thank my advisor, Professor Vasilios I. Manousiouthakis, for his guidance and contributions to this work. I gratefully acknowledge the funding received towards my graduate studies from the Ministry of Education in Brazil.

1. Introduction

A common route of hydrogen production is through steam methane reforming with natural gas as feedstock. Natural gas and steam flow inside a tube filled with catalyst pellets, where all reactions take place. In order to maximize reactions, catalyst pellets are engineered to have a high area per volume ratio, which is accomplished by a network of porous channels throughout the pellet's body.

The performance of the pellet is not dependent just on the existence of a high catalyst density area. Indeed, transport of species from the bulk flow into the pellet's body and the catalytic activity are both crucial in this case, since they ultimately dictate the amount of area that is actually being reached by reactants inside the pellet. For instance, if a reaction produces products in the interior of the pellet faster than diffusion can carry them away, reactions tend to be confined to the outer layer of the pellet, the interior being relatively inactive [1]. One method used to characterize the relation between catalytic activity and diffusivity in a porous media is by calculating the Thiele's module for the studied system [1].

The inherent limitation on the pellet's efficiency, caused by the competition between reaction and transport of species through the porous network, can also be define in terms of the catalyst pellet's effectiveness factor. The pellet's effectiveness factor is calculated as the actual overall rate of reaction, over the rate of reaction that would result if there were no diffusion limitation and all the catalyst surface area were available to the reactant at the same conditions as the external pellet surface [2, 3]. The introduction of the effectiveness factor parameter into the simulation and design of packed-bed reactors simplified calculations and reduced computational requirements.

Along with the popularization of the effectiveness factor approach, scientific efforts have been made to accurately predict the influence of several parameters on packed-bed reactors' simulations [4-9]. Due to the complexity of the physical-chemical processes, occurring simultaneously on the macroscopic and microscopic, the number of variables is enormous [10]. Parameters that can be considered in a simulation are the pellet's geometry, the pore distribution, the porous network's structure, the methodology for diffusion calculations in the porous media, the reactor's geometry, the distribution of pellets in the reactor, the model for momentum calculation (Navier-Stokes, Ergun, Darcy), assumptions regarding energy (isothermal, constant fluxes from the furnace, additional considerations on boundaries), model for diffusion calculation of the bulk flow in the reactor, among many others.

Despite the number of parameters that can influence the reaction-diffusion process on a packed-bed reactor, a study suggested that the effectiveness factor for hydrogen production from the steam methane reforming can be considered constant throughout the reactor length [11]. The study was performed considering Fickian diffusion model approach to calculate species transport inside the porous media, and the evaluation of the effectiveness factor considered geometrical approximations used by Froment years before [12]. Although the model in this work is well developed, no investigation confirming the claim by using different approaches or more accurate models was found in the literature review.

The application of more accurate models for species' transport in porous media, such as the dusty-gas model (DGM) [13], coupled with a methodology that accounts for the actual shape of the catalyst pellet, can lead to more precise calculations of the effectiveness factor for the steam reforming process. Such procedure can be used to either confirm applicability of a constant effectiveness factor hypothesis, or to reject it and propose a better approach.

1.1. Proposed work and literature review

In this work, a methodology to calculate the effectiveness factor of a packed-bed reactor from basic physical properties of the species, catalyst and packed-bed is presented. Diffusion in the porous media is described by the dusty-gas model (DGM) approach, while bulk flow diffusion is evaluated through the Maxwell-Stefan diffusion equations. A modeling procedure that couples the catalyst pellets with the reactor is then presented. Finally, the effectiveness factor for hydrogen production using the steam reforming process is calculated along a packed-bed reactor and the results are discussed.

In order to compare the proposed work with previous works in the same field, a careful review was conducted to identify modeling methods and their correspondent assumptions.

The transport of species in porous structured media has being heavily studied since the half of last century [1, 2], but earlier contributions were made more than a century ago [14]. Nevertheless, the proposal of an accurate predictive model and the subsequent validation through experiments is a very challenging task, since the dynamics of molecular transport occurs in a scale where the concept of continuum is usually not valid [15].

The dusty-gas model (DGM) is a theoretical approach developed to describe the transport of gases through porous media. James Clerk Maxwell [14] made early propositions for this theory and, more recently, some other authors [16, 17] reviewed the DGM formulation. There is a number of works comparing the DGM approach, which is developed from the Maxwell-Stefan equations, with models involving the application of the Fick's law.

Veldsink and co-authors compared Fick's law and DGM based models in the evaluation of transport and reaction in porous catalyst [18]. This study was performed for simple catalyst geometries, in which the transport is restricted to the transition region between Knudsen and

continuum diffusion. Under these conditions, deviations between these two models were very small, so they can be interchangeably used to calculate the fluxes. In spite of that, the authors highlighted the occurrence of considerable differences between the two approaches in the calculation of the partial pressures and the Thiele modulus in the pellet, parameters used in the effectiveness factor's prediction.

Later, Elnashaie and Abashar developed a mathematical model to investigate the phenomena of diffusion and reactions in porous media [19]. The rigorous dusty gas model was compared to simpler Wilke-Bosanquet model under the assumptions of steady-state, negligible viscous flow and isothermal conditions. It was found that, at low steam to methane ratios, the simplified diffusion model is adequate for the reforming process simulation. Nevertheless, at high steam to methane ratios, the implementation of DGM was found to be essential for accurate predictions.

More recently, Solsvik and Jakobsen [20] have compared several modeling formulations for the pellet equation. Minor differences between the Maxwell-Stefan and Wilke models were found on the level of fixed-bed reactor, but for a detailed calculation of a single pellet it becomes of interest.

In one of his works about hydrogen production through steam methane reforming, Froment and co-author presented a model to evaluate changes in the effectiveness factor along a reactor [12]. The pellet's reaction-diffusion mechanism is based on Fick's law and continuity equations, and the pellet's geometry is simplified from the actual form used on the experiments (catalyst rings) to an equivalent sphere and then to a planar geometry (slab) in the effectiveness factor evaluation.

In Wesenberg doctorate thesis [21], a detailed procedure for modeling a steam methane reformer is shown, including equations to model diffusion in the porous media. The author states that the dusty-gas model approach is more accurate than Fick's law to describe transport of species inside the catalyst pellet. Nevertheless, the author decided to use a Fickian model, following the study from Elnashaie and Abashar [22], which shows that, for steam reformer models with low steam-to-methane ratios, Fick's law and DGM were in good agreement. For the general case, however, the DGM is considered more precise than Fickian models in all aforementioned works. Additionally, in Wesenberg's work the effectiveness factor calculation considered the same geometrical approximations used by Froment [12]. Therefore, the actual pellet's geometry is not used to evaluate the effectiveness factor.

In terms of geometrical parameters, several approaches are possible. For the porous geometry and distribution, the most basic assumption is the approximation of porous' geometry as straight cylinders, but some works have been done in order to consider three-dimensional random networks for the catalyst support structure [8].

In another work, Behnam and co-authors focused on the actual pellet shape and compared results with experimental data for a small size reactor [23]. In their model, straight-pore Knudsen diffusion was applied. Although, it was not possible to identify the use of the DGM approach on that model.

The pellet's actual shape, and also the packing structure influence over the bulk flow, were studied by Troupel [24] in a CFD model. The author tried to simulate the diffusion on the porous media using the dusty-gas model approach, but no results were obtained due to convergence issues.

Some other modeling approaches for reaction-diffusion in porous media are presented on the literature. Whitaker proposed a multi-scale process [25,26], presenting a detailed discussion regarding the physics behind from the molecular to the continuum level in a packed-bed reactor, but the concept of effectiveness factor is not discussed in his work.

In this work, the development of a general model for catalyst pellets and packed-bed reactors that includes an accurate diffusion model, and the evaluation of the local effectiveness factor for the steam reforming reaction are pursued. The objective is to complement the current set of methodologies presented in this literature review.

2. Conceptual framework and solution approach

As described by many authors [22, 23, 26], this work also recognizes that the physical processes are occurring in the packed-bed reactor have different length scales. Transport of species occurring inside the pellet, where the pore size has the same order of magnitude as the mean free path of the molecules involved, is subject to different effects when compared to the transport of species in the reactor flow.

Consequently, the model proposed consists of solving the transport equations in two different scales: the microscale, which is called pellet domain, and the macroscale, called reactor domain. Each domain has its own characteristics:

I. Pellet domain – spatial region of the PBR suited to evaluate the pellet's features, such as porous size, porous distribution, catalytic properties, pellet shape and pellet material.

II. Reactor domain – spatial region of the PBR suited to evaluate the reactor features, such as bulk flow velocity, catalyst packing distribution, etc.

During the simulation, these two domains are coupled through the flow properties at their boundaries, such as species' concentration, species fluxes from the pellet to the bulk, pressure and temperature.

Initially, definitions regarding the geometry of the pellet and the packed-bed, such as porosity and void fractions, are presented. The objective is to clarify the mathematical integration process for a volume containing both solid and fluid phase. Then, the Reynolds Transport Theorem (RTT) is used to derive the set of transport equations needed in this work, that is, mass balance, component balance, momentum balance and energy balance. These balances are derived for the solid and fluid systems separately, for each of the pre-defined

domains (pellet and reactor). The general equations are then complemented by a set of boundary conditions, forming a system of differential equations that need to be solved simultaneously in the two domains of interest.

The main difference between the pellet and the reactor domains lies on how molecular diffusion is treated. In the pellet domain, as it is presented by several authors [17, 27, 28], transport in porous media is mainly performed by four independent mechanisms:

1) Knudsen diffusion, which takes place in very small porous where collisions between molecules can be ignored compared with molecule-wall collisions.
2) Continuum diffusion, which is the movement of species due to concentration gradient.
3) Viscous (or convective) diffusion, which is the movement of species due to pressure gradients.
4) Surface diffusion, which molecules move along a solid surface.

Additionally, capillarity condensation, molecular sieving and diffusion solubility mechanisms [15] can be important in the molecular transport involving small pores. Nevertheless, in this study, the effects of these mechanisms are neglected.

The dusty-gas model takes in account the three main aforementioned mechanisms, where the continuum diffusion is derived using the Stefan-Maxwell equations [14]. Surface diffusion is not considered in this work, although the DGM derivation allows the consideration of this mechanism if needed.

For the reactor domain, due to the size of the voids, Knudsen effects are inexistent and diffusive fluxes can be calculated using a continuum diffusion model. In this work, diffusion in the reactor domain is calculated through the Stefan-Maxwell equation.

To estimate the diffusion coefficients for the respective pairs of species in the mixture, the theory developed by Chapman and Enskog [30, 31] is applied in this work.

Lastly, the system of differential equations is then solved in COMSOL® , considering two different simulation cases: 1) a steady-state, one dimensional catalyst pellet; 2) a steam methane reformer reactor, in which the effectiveness factor is calculated throughout the reactor length.

3. Packed-bed reactor model

In this section, structural parameters are discussed, and then transport equations are derived for the two domains previously defined for the packed bed reactor.

3.1. Structural parameters and porous media geometry

Intensive properties are independent of the sample size. Thus, by definition, intensive properties are given either per unit of volume, e.g. mass concentration (density) and molar concentration, or per unit of surface area, such as diffusive mass flux. For a given intensive property, the usual approach is to consider the volume or surface area completely filled by the extensive property of interest.

Nevertheless, some media of interest in engineering are in fact a composition of different physical states, such as porous media. This fact does not change the intensive properties per se, since the intensive properties of any isolated phase, even inside a porous media, will be the same when compared to the same phase in a continuous media. But in practice, the integration of any property over its single phase volume is not feasible, considering the random distribution of the solid and voids inside the porous media. Moreover, even if the solid and voids spatial positions are determined for one pellet, it is unlikely to find the same distribution in different pellets, which reduces the generality of a methodology.

Thus, to simplify integrations and improve generality when solving the Reynolds Transport Theorem (RTT), functions mapping the spatial position of a homogenous phase onto the overall spatial domain are defined.

In this document, when the terms pellet volume, pellet area, reactor volume or reactor area are used, the reference is made to describe that, for a given intensive variable or quantity, the volume or area of reference contains all applicable phases, as showed in the example below:

$$\begin{Bmatrix} V^{Pellet} = V_{Solid}^{Pellet} + V_{Voids}^{Pellet} \\ A^{Pellet} = A_{Solid}^{Pellet} + A_{Voids}^{Pellet} \\ V^{Reactor} = V_{Solid}^{Reactor} + V_{Voids}^{Reactor} \\ A^{Reactor} = A_{Solid}^{Reactor} + A_{Voids}^{Reactor} \end{Bmatrix} \tag{1}$$

This definition is important to avoid misinterpretation in the terms used in this work.

3.1.1. Pellet geometric parameters

Considering the notation presented in the last section, the pellet's volumetric porosity ε_V is defined as the volume of voids inside a differential volume of the pellet divided by the differential volume of the pellet:

$$\varepsilon_V \triangleq \frac{dV_{Voids}^{Pellet}}{dV^{Pellet}} \tag{2}$$

Then, the ratio between the solid portion of a differential volume and the whole differential volume of the pellet must satisfy the following equation:

$$\frac{dV_{Solid}^{Pellet}}{dV^{Pellet}} = \frac{dV^{Pellet} - dV_{Voids}^{Pellet}}{dV^{Pellet}} = 1 - \frac{dV_{Voids}^{Pellet}}{dV^{Pellet}} = \left(1 - \varepsilon_V\right) \tag{3}$$

Using a similar approach for the pellet area, the area porosity ε_A is defined as the void portion of the pellet's differential area divided by the pellet's differential area.

$$\varepsilon_A \triangleq \frac{dA_{Void}^{Pellet}}{dA^{Pellet}} \tag{4}$$

The area to volume interfacial factor ε_I quantifies the area of the solid-void interface within a differential volume of the pellet:

$$\varepsilon_I \triangleq \frac{dA^{Pellet}_{Solid-Voids\ interface}}{dV^{Pellet}} \tag{5}$$

An important parameter in porous media is the tortuosity factor τ, which is defined as the average distance traveled by a molecule in a porous media divided by the differential distance of the respective straight path:

$$\tau \triangleq \frac{dL^{Porous}_{Actual}}{dL^{Porous}_{Straight\ path}} \tag{6}$$

In a real porous media, such as catalyst pellets, parameters like porosity and tortuosity varies throughout the pellet's volume. Despite the efforts that have been done to characterize the network distribution inside the pellet [8], a descriptive model was not found. For the purpose of this work, all these parameters are considered to be constant throughout the pellet shape.

3.1.2. Reactor geometric parameters

In the reactor domain, a similar procedure is executed to define the reactor's volumetric void fraction ϕ_V, which is defined as the portion of voids inside a differential volume of the reactor divided by the respective differential volume of the reactor:

$$\phi_V \triangleq \frac{dV^{Reactor}_{Voids}}{dV^{Reactor}} \tag{7}$$

Then, the ratio of the volume occupied by solids in a differential volume of the reactor is:

$$\frac{dV_{Solid}^{Reactor}}{dV^{Reactor}} = \frac{dV^{Reactor} - dV_{Solid}^{Reactor}}{dV^{Reactor}} = 1 - \frac{dV_{Fluid}^{Reactor}}{dV_{Total}^{Reactor}} = \left(1 - \phi_V\right) \tag{8}$$

Considering a differential area of the reactor, the area void fraction ϕ_A represents the ratio between the area of voids and the total differential area of the reactor.

$$\phi_A \triangleq \frac{dA_{Voids}^{Reactor}}{dA^{Reactor}} \tag{9}$$

The area to volume interface factor for the reactor ϕ_I takes in account the interfacial area between solids and voids inside the reactor:

$$\phi_I \triangleq \frac{dA_{Solid-Voids\ interface}^{Reactor}}{dV^{Reactor}} \tag{10}$$

It is important to clarify that Eq.(7) to Eq.(10) are defined for the macroscale, i.e., the reactor domain. Therefore, in these equations the pellet is considered a solid, nonporous entity.

3.1.3. Notation definitions for reactor, pellet and single phase properties

As stated before, it is important to correctly identify the volume used on intensive properties. Even though the definition of the terms “pellet volume” and “reactor volume” were previously stated, the correspondent notation in the equations needs to be specified, especially to avoid confusion when treat the systems’ boundaries.

All intensive properties based on the volume or area of a homogeneous phase, such as a fluid in a void volume, or a property of the solid, will be written without any superscript. For exemple:

$$c_i = \frac{mol\ of\ i}{m^3\ of\ \ fluid} \quad ; \quad \rho_{cat} = \frac{kg\ of\ catalyst}{m^3\ of\ solid\ catalyst} \quad ; \quad n_i = \frac{kg\ of\ i}{m^2\ of\ \ fluid \cdot s} \tag{11}$$

Conversely, when an intensive property is then written in terms of the pellet or the reactor's spatial characteristics, meaning that one of the porosity parameters defined previously was used, a superscript is used to identify which of the domains are being. For example:

$$\begin{aligned} c_i^r = \phi_V c_i \quad &= \quad \frac{m^3\ of\ \ fluid}{m^3\ of\ reactor} \cdot \frac{mol\ of\ i}{m^3\ of\ \ fluid} = \frac{mol\ of\ i}{m^3\ of\ reactor} \\ \rho_s^p = \left(1-\varepsilon_V\right)\rho_s \quad &= \quad \frac{m^3\ of\ solid}{m^3\ of\ \ pellet} \cdot \frac{kg\ of\ solid}{m^3\ of\ solid} = \frac{kg\ of\ solid}{m^3\ of\ \ pellet} \\ n_i^p = \varepsilon_A n_i \quad &= \quad \frac{m^2\ of\ \ fluid}{m^2\ of\ \ pellet} \cdot \frac{kg\ of\ i}{m^2\ of\ \ fluid} = \frac{kg\ of\ i}{m^2\ of\ \ pellet} \end{aligned} \tag{12}$$

This notation will be used throughout the derivation of the conservation laws using the Reynolds Transport Theorem (RTT).

3.2. Reynolds Transport Theorem - RTT

The Reynolds' Transport Theorem (RTT) is the basic concept behind the derivation of the modeling equations for any physical system. The RTT relates the rate of change of a given extensive property of the system, with the accumulation and the influx/outflow of the property within a defined control volume.

$$\frac{dM_{CSys}^{t}(t)}{dt} = \frac{\partial}{\partial t}\left(\iiint_{CV} M\,\rho dV^{t}\right) + \iint_{CS} M\,\rho\vec{v}\cdot\vec{n}dA^{t} \tag{13}$$

The left hand side term in Eq.(13) represents the rate of change of any extensive property M_{CSys}^{t} of the considered system. The subscript $CSys$ refers to a composite thermodynamic system, which is a large collection of simple thermodynamic systems (Sys) of infinitesimal size and mass m, and the superscript t represents the time dependence of the RTT variables.

The first term of the right hand side of the RTT represents the variation of the considered extensive property contained in the control volume CV with respect to time. The second term of the right hand side represents the variation inside the control volume CV due to the transport of the extensive variable from or to the control volume through the control surface CS. In both terms presented in the RHS, the extensive property is evaluated as the product of the respective intensive property M and the local mass density ρ of a differential volume dV^{t}.

For the derivation of the PBR equations, the RTT is applied separately to the pellet domain and the reactor domain. In addition, two subsystems are considered in each domain: the fluid subsystem, that occupies the voids in the system, and the solid subsystem.

In the RTT equation, ρ is the system density, i.e., no differentiation in terms of subsystems is considered on it. In a system where the geometry of interest contains solids and voids (ultimately a fluid) with different properties, the proportion between the two phases in the volumetric and/or surface integration of the system is defined as:

$$\rho = \varepsilon_V \rho_f + (1-\varepsilon_V)\rho_s \quad ; \quad \textit{in the pellet volume integral} \tag{14}$$

$$\rho = \varepsilon_A \rho_f + (1-\varepsilon_A)\rho_s \quad ; \quad \textit{in the pellet surface integral} \tag{15}$$

$$\rho = \phi_V \rho_f + (1-\phi_V)\rho_s \quad ; \quad \textit{in the reactor volume integral} \tag{16}$$

$$\rho = \phi_A \rho_f + \left(1-\phi_A\right)\rho_s \quad ; \quad in\ the\ reactor\ surface\ integral \tag{17}$$

Then, the application of the RTT for the pellet composite system leads to:

$$\begin{aligned}\frac{dM^t_{CSys}(t)}{dt} &= \frac{dM^t_{Fluid\,sys}(t)}{dt} + \frac{dM^t_{Solid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left(\iiint_{CV} M\,\rho\,dV^t\right) + \iint_{CS} M\,\rho\,\vec{v}\cdot\vec{n}dA^t = \\ &= \frac{\partial}{\partial t}\left(\iiint_{CV}\left[M_f\varepsilon_V\rho_f + M_s\left(1-\varepsilon_V\right)\rho_s\right]dV^t\right) + \iint_{CS}\left(M_f\varepsilon_A\rho_f + M_s\left(1-\varepsilon_A\right)\rho_s\right)\vec{v}\cdot\vec{n}dA^t\end{aligned} \tag{18}$$

Regrouping the multiphase equation in terms of the individual subsystems:

$$\begin{aligned}\frac{dM^t_{Fluid\,sys}(t)}{dt} + \frac{dM^t_{Solid\,sys}(t)}{dt} &= \left[\frac{\partial}{\partial t}\left(\iiint_{CV} M_f\rho_f\varepsilon_V\,dV^t\right) + \iint_{CS} M_f\rho_f\overrightarrow{v_f}\cdot\vec{n}\,\varepsilon_A dA^t\right] + \\ &+ \left[\frac{\partial}{\partial t}\left(\iiint_{CV} M_s\,\rho_s\left(1-\varepsilon_V\right)dV^t\right) + \iint_{CS} M_s\,\rho_s\overrightarrow{v_s}\cdot\vec{n}\left(1-\varepsilon_A\right)dA^t\right]\end{aligned} \tag{19}$$

Similarly, for the reactor composite system:

$$\begin{aligned}\frac{dM^t_{Fluid\,sys}(t)}{dt} + \frac{dM^t_{Solid\,sys}(t)}{dt} &= \left[\frac{\partial}{\partial t}\left(\iiint_{CV} M_f\rho_f\phi_V\,dV^t\right) + \iint_{CS} M_f\rho_f\overrightarrow{v_f}\cdot\vec{n}\,\phi_A dA^t\right] + \\ &+ \left[\frac{\partial}{\partial t}\left(\iiint_{CV} M_s\,\rho_s\left(1-\phi_V\right)dV^t\right) + \iint_{CS} M_s\,\rho_s\overrightarrow{v_s}\cdot\vec{n}\left(1-\phi_A\right)dA^t\right]\end{aligned} \tag{20}$$

Equations (19) and (20) show that the total change for the composite system is equal to the sum of the individual subsystems.

In the reverse operation, one can apply the RTT for the solid and the fluid subsystems individually, generating two independent equations:

$$\frac{dM^t_{Fluid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left(\iiint_{CV} M_f\rho_f dV^t_f\right) + \iint_{CS} M_f\rho_f\overrightarrow{v_f}\cdot\vec{n}dA^t_f \tag{21}$$

$$\frac{dM^{t}_{Solid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left(\iiint_{CV} M_s \rho_s dV_s^t\right) + \iint_{CS} M_s \rho_s \overrightarrow{v_s} \cdot \overrightarrow{n} dA_s^t \tag{22}$$

If the individual subsystem, with their respective solid or liquid volumes, are summed up, the proportion between solid and liquid volumes with respect to the total volume can be calculated, and the relations presented for the pellet, in Eq.(2) to Eq.(4), and for the reactor, in Eq.(7) to Eq.(9), can be defined. Therefore, using the notation previously defined for the volumetric proportions, the RTT for the individual subsystems can be written in terms of the sum of volumes as follows:

$$\frac{dM^{t}_{Fluid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left(\iiint_{CV} M_f \rho_f \varepsilon_V dV^t\right) + \iint_{CS} M_f \rho_f \overrightarrow{v_f} \cdot \overrightarrow{n}\, \varepsilon_A\, dA^t \tag{23}$$

$$\frac{dM^{t}_{Solid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left[\iiint_{CV} M_s \rho_s \left(1-\varepsilon_V\right) dV^t\right] + \iint_{CS} M_s \rho_s \overrightarrow{v_s} \cdot \overrightarrow{n} \left(1-\varepsilon_A\right) dA^t \tag{24}$$

$$\frac{dM^{t}_{Fluid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left(\iiint_{CV} M_f \rho_f \phi_V dV^t\right) + \iint_{CS} M_f \rho_f \overrightarrow{v_f} \cdot \overrightarrow{n}\, \phi_A\, dA^t \tag{25}$$

$$\frac{dM^{t}_{Solid\,sys}(t)}{dt} = \frac{\partial}{\partial t}\left[\iiint_{CV} M_s \rho_s \left(1-\phi_V\right) dV^t\right] + \iint_{CS} M_s \rho_s \overrightarrow{v_s} \cdot \overrightarrow{n} \left(1-\phi_A\right) dA^t \tag{26}$$

where Eq.(23) and Eq.(24) are valid for the pellet domain and Eq.(25) and (26) for the reactor domain. It is easy to verify that these last equations, derived for the individual homogeneous phase in each domain, are the same expressions forming in Eq.(19) and Eq.(20), derived for the multiphase system. Thus, the application of the RTT in a single homogeneous phase system, the subsequent approach to describe an individual phase in a composite volume, and the relation of the last with the composite system are central in the derivation of the conservation laws for the studied systems.

3.3. Total mass conservation

In the derivation of the total mass conservation law for the liquid and the solid subsystems using the RTT, the extensive property chosen for the fluid and the solid system are the mass of fluid m_f and the mass of solid m_s, respectively. Thus, the intensive property M_f and M_s for each of the considered system is described as:

$$\left\{ M_f = 1 \wedge \frac{dM^t_{Fluid\,sys}(t)}{dt} = \frac{d(m)_{Fluid\,sys}(t)}{dt} \right\} \tag{27}$$

$$\left\{ M_s = 1 \wedge \frac{dM^t_{Solid\,sys}(t)}{dt} = \frac{d(m)_{Solid\,sys}(t)}{dt} = \frac{-d(m)_{Fluid\,sys}(t)}{dt} \right\} \tag{28}$$

Considering that the mass of the solid system is not augmented through incorporation of mass from the fluid system, then

$$\frac{d(m)_{fluid\,sys}(t)}{dt} = \frac{-d(m)_{solid\,sys}(t)}{dt} = 0 \tag{29}$$

Thus, the RTT for the fluid subsystem in the pellet becomes:

$$\frac{d(m)_{fluid\,sys}(t)}{dt} = 0 = \frac{\partial}{\partial t}\left(\iiint_{CV} \rho_f \varepsilon_V dV^t \right) + \iint_{CS} \rho_f \overrightarrow{v_f} \cdot \overrightarrow{n}\, \varepsilon_A dA^t \tag{30}$$

Applying of the Gauss' divergence theorem on Eq.(30), the double integral over the control surface becomes a triple integral over the control volume:

$$\frac{d(m)_{fluid\,sys}(t)}{dt} = 0 = \frac{\partial}{\partial t}\left(\iiint_{CV} \rho_f \varepsilon_V dV^t \right) + \iiint_{CV} \vec{\nabla} \cdot \left(\varepsilon_A \rho_f \overrightarrow{v_f} \right) dV^t \tag{31}$$

Using the Leibnitz's theorem, the partial derivative with respect to the time in the first term of the right hand side of Eq.(31) can be included in the integrand of the volume integral as follows:

$$\frac{d\left(m\right)_{fluid\ sys}\left(t\right)}{dt}=0=\iiint_{CV}\frac{\partial}{\partial t}\left(\varepsilon_V\rho_f\right)dV^t+\iiint_{CV}\vec{\nabla}\cdot\left(\varepsilon_A\rho_f\overrightarrow{v_f}\right)dV^t \tag{32}$$

Finally, considering for the fluid system an infinitesimally small control volume CV with volume $\delta V^t=\iiint_{CV}dV^t$, then for the fluid subsystem in the pellet domain:

$$0=\frac{\partial}{\partial t}\left(\varepsilon_V\rho_f\right)+\vec{\nabla}\cdot\left(\varepsilon_A\rho_f\overrightarrow{v_f}\right) \tag{33}$$

For the solid subsystem in the pellet domain, the same procedure is valid resulting in:

$$0=\frac{\partial}{\partial t}\left(\left(1-\varepsilon_V\right)\rho_s\right)+\vec{\nabla}\cdot\left(\left(1-\varepsilon_A\right)\rho_s\overrightarrow{v_s}\right) \tag{34}$$

Similarly, for the fluid and solid subsystems in the reactor domain, respectively:

$$0=\frac{\partial}{\partial t}\left(\phi_V\rho_f\right)+\vec{\nabla}\cdot\left(\phi_A\rho_f\overrightarrow{v_f}\right) \tag{35}$$

$$0=\frac{\partial}{\partial t}\left(\left(1-\phi_V\right)\rho_s\right)+\vec{\nabla}\cdot\left(\left(1-\phi_A\right)\rho_s\overrightarrow{v_s}\right) \tag{36}$$

The result obtained for the solid system, Eq.(34) and Eq.(36), are important for the analysis of systems fluidized bed systems, where the second phase present in the system is not stagnant, or systems where the generation or consumption of the solid phase with time is not negligible. In this study, however, the solid phase, represented by the catalyst and its support material, is considered stagnant and time-invariant.

3.4. Component mass conservation

Assuming no changes in the solid phase composition, the component mass conservation law is derived for the fluid system only. In this case, the extensive property of the system is the mass m_i of the species i in the mixture, and the respective intensive property for the fluid system M_f is the mass fraction w_i of the component i in the fluid phase. Thus, the RTT for the pellet domain takes the following form:

$$M_f = w_i \wedge \frac{dM^t_{fluid\,sys}(t)}{dt} = \iiint_{CV} R_i \, dV^t - \iint_{CS} \overrightarrow{j_i} \cdot \overrightarrow{n} \varepsilon_A dA^t \quad ; \quad i = 1, \upsilon \tag{37}$$

In Eq. (37), the accumulation of species i mass over time is equal to the mass of i generated inside the control volume due to reaction, represented by the volumetric reaction rate R_i, minus the diffusive mass flux $\overrightarrow{j_i}$ of component i crossing the permeable area of the control surface.

For the pellet domain, using Eq.(37) as the starting point, the same procedure previously applied in the total mass balance can be followed in this case:

$$\begin{aligned} &\iiint_{CV} R_i dV^t - \iint_{CS} \overrightarrow{j_i} \cdot \overrightarrow{n} \varepsilon_A dA^t = \frac{\partial}{\partial t}\left(\iiint_{CV} w_i \rho_f \varepsilon_V dV^t \right) + \iint_{CS} w_i \rho_f \overrightarrow{v_f} \cdot \overrightarrow{n} \, \varepsilon_A dA^t \underset{Divergence}{\overset{Gauss'}{\Rightarrow}} \\ &\iiint_{CV} R_i dV^t - \iiint_{CV} \vec{\nabla} \cdot \left(\varepsilon_A \overrightarrow{j_i} \right) dV^t = \frac{\partial}{\partial t}\left(\iiint_{CV} \varepsilon_V w_i \rho_f dV^t \right) + \iiint_{CV} \vec{\nabla} \cdot \left(\varepsilon_A w_i \rho_f \overrightarrow{v_f} \right) dV^t \end{aligned} \tag{38}$$

Applying the Leibnitz's theorem on Eq.(38):

$$\iiint_{CV} R_i dV^t - \iiint_{CV} \vec{\nabla} \cdot \left(\varepsilon_A \overrightarrow{j_i} \right) dV^t = \iiint_{CV} \frac{\partial}{\partial t}\left(\varepsilon_V w_i \rho_f \right) dV^t + \iiint_{CV} \vec{\nabla} \cdot \left(\varepsilon_A w_i \rho_f \overrightarrow{v_f} \right) dV^t \tag{39}$$

Then, for the pellet domain, considering in Eq.(39) an infinitesimally small control volume CV with volume $\delta V^t = \iiint_{CV} dV^t$, the following expression is obtained:

$$R_i - \vec{\nabla} \cdot \left(\varepsilon_A \vec{j_i} \right) = \frac{\partial}{\partial t} \left(\varepsilon_V w_i \rho_f \right) + \vec{\nabla} \cdot \left(\varepsilon_A w_i \rho_f \vec{v_f} \right) \quad ; \quad i = 1, \upsilon \tag{40}$$

The following expressions from the mass transfer theory are used in this section to rewrite Eq.(40) in terms of the total mass flux:

$$\vec{n_i} = \rho_i \vec{v_i} \tag{41}$$

$$\vec{j_i} = \rho_i \left(\vec{v_i} + \vec{v_f} \right) \tag{42}$$

$$\vec{v_f} = \sum_{i=1}^{\upsilon} w_i \vec{v_i} \tag{43}$$

$$w_i = \frac{\rho_i}{\sum_{i=1}^{\upsilon} \rho_i} \tag{44}$$

In Eq.(41), $\vec{n_i}$ is the mass flux of i with respect to stationary axes, or the total mass flux of species i. Equation (42) is the diffusive mass flux of i, or the mass flux of species i when the origin of the reference system is considered to move with the same velocity as the mass average velocity $\vec{v_f}$, described by Eq.(43). Finally, Eq. (44) is the definition of mass fraction of species i in terms of the species' mass concentrations in the pellet system.

Using the aforementioned equations, Eq.(40) can be rewritten as follows:

$$
\begin{aligned}
& R_i - \vec{\nabla}\cdot\left(\varepsilon_A \vec{j_i}\right) = \frac{\partial}{\partial t}\left(\varepsilon_V w_i \rho_f\right) + \vec{\nabla}\cdot\left(\varepsilon_A w_i \rho_f \vec{v_f}\right) \qquad \Leftrightarrow \\
& R_i = \frac{\partial}{\partial t}\left(\varepsilon_V w_i \rho_f\right) + \vec{\nabla}\cdot\left(\varepsilon_A w_i \rho_f \vec{v_f}\right) + \vec{\nabla}\cdot\left(\varepsilon_A \vec{j_i}\right) \quad \overset{\vec{\nabla}\cdot(A)+\vec{\nabla}\cdot(B)=\vec{\nabla}\cdot(A+B)}{\Leftrightarrow} \\
& R_i = \frac{\partial}{\partial t}\left(\varepsilon_V w_i \rho_f\right) + \vec{\nabla}\cdot\left(\varepsilon_A w_i \rho_f \vec{v_f} + \varepsilon_A \vec{j_i}\right)
\end{aligned}
\tag{45}
$$

The last part of Eq.(45) is simplified by using Eq.(41) to Eq.(44), resulting in the component mass balance equation in terms of the total mass flux $\vec{n_i}$:

$$R_i = \frac{\partial}{\partial t}\left(\varepsilon_V w_i \rho_f\right) + \vec{\nabla}\cdot\left(\varepsilon_A \vec{n_i}\right) \quad ; \quad i = 1, \upsilon \tag{46}$$

Similarly, for the reactor domain, the component balance equation for species i in terms of the diffusive mass flux and the total mass flux are respectively:

$$R_i - \vec{\nabla}\cdot\left(\phi_A \vec{j_i}\right) = \frac{\partial}{\partial t}\left(\phi_V w_i \rho_f\right) + \vec{\nabla}\cdot\left(\phi_A w_i \rho_f \vec{v_f}\right) \quad ; \quad i = 1, \upsilon \tag{47}$$

$$R_i = \frac{\partial}{\partial t}\left(\phi_V w_i \rho_f\right) + \vec{\nabla}\cdot\left(\phi_A \vec{n_i}\right) \quad ; \quad i = 1, \upsilon \tag{48}$$

The volumetric reaction rate is defined in terms of the catalyst density as follows:

$$R_i = M_i \bar{R}_i \left(1 - \varepsilon_V\right) \rho_s \tag{49}$$

In this equation, M_i is the molecular weight of species i, the molar reaction rate $\bar{R}_i$ is defined in mol of species i per catalyst weight, and ρ_s is the density of the solid catalyst.

Diffusion models that are able to describe the mass flux in a porous media for the pellet domain, and the mass flux in a packed bed for the reactor domain, are necessary in order to solve the component mass balance equations presented in this section.

For the pellet domain, the chosen model was the Dusty Gas Model (DGM) [17], which ultimately leads to a momentum balance equation inside the porous media considering some assumptions. For the reactor domain, the diffusion model of choice was the Maxwell-Stefan diffusion equation [14, 27, 33], which describes the diffusive mass flow. These models will be discussed in the next section.

3.5. Diffusion models for mass transfer in the pellet and the reactor domains

The solution of the component mass balance equations depends on the definition of a mass transfer model that describes the diffusive mass flux $\vec{j_i}$ or the total mass flux $\vec{n_i}$. Several models can be found on the literature to describe the molecular motion subject to different driving forces and occurring in a dimensional range that goes from molecular scale to continuum, [27, 33].

In this work, as previously mentioned, two dimensional levels are considered. The first is the pellet domain, which describes the molecular level, where the diffusion process occurs in small channels (porous) that have the same order of magnitude of the mean free path of the species involved. The second is the reactor domain, which describes the macroscale level, where free diffusion can occur without interference of the solid elements spread throughout the reactor. The model corresponding to each of these dimensional levels are presented in the following sections.

3.5.1. Pellet domain - Dusty gas model

The model chosen for the pellet is the dusty gas model (DGM) [17], which assumes that diffusion occurs in the voids formed among giant stagnant molecules (called dust). In addition to the assumption of one stagnant solid species, the DGM derivation assumes steady state and neglect inertial (viscous) terms of the fluid motion. The full set of assumptions as well as the derivation procedure can be found in Mason and Malinauskas work [17], although several other authors also describe the same derivation procedure, sometimes with additional assumptions such as constant pressure [36].

The DGM equation for total molecular flow transport is:

$$\sum_{j=1}^{\upsilon} \frac{x_j}{D_{ij}^{eff}} \left(\frac{\overrightarrow{N_i^p}}{c_i} - \frac{\overrightarrow{N_j^p}}{c_j} \right) + \frac{1-\Delta'_{id}}{\left[D_{iK}\right]_1} \left[\frac{\overrightarrow{N_i^p}}{c_i} + \frac{B_O}{\mu_f} \left(\vec{\nabla} p - \sum_{j=1}^{\upsilon} c_j \overrightarrow{F_j} \right) \right] =$$

$$= -\frac{1}{x_i} \vec{\nabla} x_i - \frac{1}{p} \vec{\nabla} p + \frac{\overrightarrow{F_i}}{\tilde{R}T} - \left(\frac{1}{c'_{Tot}} \right) \left[\sum_{j=1}^{\upsilon} c_j \left(\alpha'_{ij} \right)_{tr} + c_d \left(\alpha'_{id} \right)_{tr} \right] \vec{\nabla} \ln T \qquad (50)$$

$$for \;\; i, j \neq d$$

The term D_{ij}^{eff} in Eq.(50) is the binary effective diffusion coefficient of the $i-j$ species pair, which considers the resistance imposed by the porous media to the molecular transport that would happen in the free space. The effective diffusivity is calculated as:

$$D_{ij}^{eff} = \left(\frac{\varepsilon_V}{\tau} \right) D_{ij} \qquad (51)$$

In Eq.(50) the subscript d stands for dust, which accounts for solid part of the porous media, $\overrightarrow{F_i}$ is the forces acting on species i per mole, such as gravity or forces due to electrical fields. Any primed quantity in this equation considers the dust as one of the species in the mixture, or as one of the elements in binary interactions between the species. All other terms are

listed on the notation section. In Eq.(51), D_{ij} is the binary diffusion coefficient in the free space, as shown in chapter 3.5.3 of this work.

The notation used in Eq.(50) respects the volumetric considerations previously discussed in this work. For example, concentrations are per volume of fluid phase and not per volume of pellet. This comes from the DGM derivation, where the concentration of the 'dust' species was once part of the system of equations and then eliminated due to a linear dependence with the others. Conversely, the molar fluxes $\overrightarrow{N_i^p}$ are per area of pellet and not per permeable area of the control surface.

The correction term Δ'_{id} which appears in the Knudsen diffusion term is usually very small and is considered to be equal zero. The first order approximation of the Knudsen diffusivity D_{iK} can be described by the equation [17, 29]:

$$\left[D_{iK}\right]_1 = D_{iK} = \left(\frac{\varepsilon_V}{\tau}\right)\frac{d_{pore}}{3}\sqrt{\frac{8k_B T}{\pi m_i}} = \left(\frac{\varepsilon_V}{\tau}\right)\frac{4d_{pore}}{3}\sqrt{\frac{\tilde{R}T}{2\pi M_i}} \tag{52}$$

Neglecting the effect of field forces, Eq.(50) can be further simplified and written in terms of the species molar fractions:

$$\begin{cases} \displaystyle\sum_{j=1}^{\upsilon}\frac{x_j}{D_{ij}^{eff}}\left(\frac{\overrightarrow{N_i^p}}{c_i}-\frac{\overrightarrow{N_j^p}}{c_j}\right)+\frac{1}{D_{iK}}\left[\frac{\overrightarrow{N_i^p}}{c_i}+\frac{B_O}{\mu_f}\left(\vec{\nabla}p-\sum_{j=1}^{\upsilon}c_j\overrightarrow{F_j}\right)\right]= \\ \displaystyle\qquad = -\frac{1}{x_i}\vec{\nabla}x_i-\frac{1}{p}\vec{\nabla}p+\frac{\overrightarrow{F_i}}{RT}-\left(\frac{1}{c'_{Tot}}\right)\left[\sum_{j=1}^{\upsilon}c_j\left(\alpha'_{ij}\right)_{tr}+c_d\left(\alpha'_{id}\right)_{tr}\right]\vec{\nabla}\ln T \end{cases} \overset{\overrightarrow{F_i}=\overrightarrow{F_j}=0}{\underset{c_i=x_i c_{Tot}}{\Leftrightarrow}}$$

$$\begin{cases} \displaystyle\sum_{j=1}^{\upsilon}\frac{x_j}{c_{Tot}D_{ij}^{eff}}\left(\frac{\overrightarrow{N_i^p}}{x_i}-\frac{\overrightarrow{N_j^p}}{x_j}\right)+\frac{1}{c_{Tot}D_{iK}}\left[\frac{\overrightarrow{N_i^p}}{x_i}+c_{Tot}\frac{B_O}{\mu_f}\left(\vec{\nabla}p\right)\right]= \\ \displaystyle\qquad = -\frac{1}{x_i}\vec{\nabla}x_i-\frac{1}{p}\vec{\nabla}p-\left[\sum_{j=1}^{\upsilon}x'_j\left(\alpha'_{ij}\right)_{tr}+x'_d\left(\alpha'_{id}\right)_{tr}\right]\vec{\nabla}\ln T \end{cases} \tag{53}$$

Rearranging Eq.(53) and multiplying by the molar fraction of species i:

$$\left\{\begin{aligned}&\sum_{j=1}^{\upsilon}\frac{1}{c_{Tot}D_{ij}^{eff}}\left(x_j\overrightarrow{N_i^p}-x_i\overrightarrow{N_j^p}\right)+\frac{1}{c_{Tot}D_{iK}}\overrightarrow{N_i^p}=\\&\quad=-\vec{\nabla}x_i-\frac{x_iB_O}{\mu_fD_{iK}}\vec{\nabla}p-\frac{x_i}{p}\vec{\nabla}p-x_i\left[\sum_{j=1}^{\upsilon}x_j'\left(\alpha_{ij}'\right)_{tr}+x_d'\left(\alpha_{id}'\right)_{tr}\right]\vec{\nabla}\ln T\end{aligned}\right\}\tag{54}$$

Now, multiplying Eq.(54) by the total concentration c_{Tot}, and writing in terms of the total mass fluxes, the DGM final equation has the form:

$$\begin{aligned}&\sum_{j=1}^{\upsilon}\frac{1}{D_{ij}^{eff}}\left(\frac{x_j}{M_i}\overrightarrow{n_i^p}-\frac{x_i}{M_j}\overrightarrow{n_j^p}\right)+\left(\frac{1}{M_iD_{iK}}\right)\overrightarrow{n_i^p}=\\&\quad=-c_{Tot}\vec{\nabla}x_i-\left[1+\frac{p}{D_{iK}}\frac{B_O}{\mu_f}\right]\left(\frac{x_i}{\tilde{R}T}\vec{\nabla}p\right)-x_i\left[\sum_{j=1}^{\upsilon}x_j'\left(\alpha_{ij}'\right)_{tr}+x_d'\left(\alpha_{id}'\right)_{tr}\right]\vec{\nabla}\ln T\end{aligned}\tag{55}$$

In the right hand side of Eq.(55), the first term is the diffusive driving force, the second is the convective driving force and the third is the thermal driving force, which has two components: the first term inside the brackets is related to the thermal diffusion effect and the second is related to the thermal transpiration effect.

The viscous flow parameter B_O for the porous media is calculated considering a cylindrical pore subjected to the effects of the porosity-tortuosity factor [17]:

$$B_O=\frac{\varepsilon_V}{\left(\frac{L_{Actual}^{Pore}}{L_{Straight\ path}^{Pore}}\right)^2}\left(\frac{r_{pore}{}^2}{8}\right)=\frac{\varepsilon_V}{\tau^2}\left(\frac{r_{pore}{}^2}{8}\right)\tag{56}$$

The viscosity of the fluid mixture μ_f is calculated using the following model [27]:

$$\mu_f = \sum_{i=1}^{\nu} \left(\frac{x_i \mu_i}{\sum_{j=1}^{\nu} x_j \Phi_{ij}} \right) \tag{57}$$

$$\Phi_{ij} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_i}{M_j} \right)^{-1/2} \left[1 + \left(\frac{\mu_i}{\mu_j} \right)^{1/2} \left(\frac{M_j}{M_i} \right)^{1/4} \right]^2 \tag{58}$$

The calculation of the thermal diffusion coefficients $\left(\alpha'_{ij}\right)_{tr}$ and the thermal transpiration coefficient $\left(\alpha'_{id}\right)_{tr}$ are not included in the scope of this work. However, they can be estimated by using the Chapman-Enskog theory [17], if necessary in future works. The thermal diffusion influence on the overall transport of species in a porous media depends on the pellet's temperature gradient, which is evaluated in this work and can function as a starting point for future works.

3.5.2. Reactor domain – Maxwell-Stefan equation

In the reactor domain, even though the reactor is filled with solid material (catalyst pellets), the space available for molecular diffusion is not restrictive enough to create Knudsen effects on the transport. Thus, diffusion models for free molecular transport are assumed in the rector domain. In this work, the Maxwell-Stefan model [33] was chosen to describe the multicomponent molecular transport in the reactor:

$$\sum_{j=1}^{\upsilon} \frac{x_i \; x_j}{D_{ij}} \left(\vec{v}_i - \vec{v}_j \right) = -\vec{\nabla} x_i - \left(x_i - w_i \right) \left(\frac{\vec{\nabla} p}{p} \right) - \sum_{j=1}^{\upsilon} \frac{x_i \; x_j}{\rho_f D_{ij}} \left(\frac{D_i^T}{w_i} - \frac{D_j^T}{w_j} \right) \left(\frac{\vec{\nabla} T}{T} \right) \tag{59}$$

In the Maxwell-Stefan equation, molecular velocity of species i in the mixture is proportional to a driving force composed by the gradient in concentration, pressure and temperature. Using Eq.(41) to change velocities to diffusive mass fluxes, Eq. (59) takes the following form:

$$\vec{\nabla} x_i = \sum_{j=1}^{\upsilon} \frac{x_i\, x_j}{D_{ij}} \left(\frac{1}{\rho_j} \vec{j}_j - \frac{1}{\rho_i} \vec{j}_i \right) + \left(w_i - x_i \right) \left(\frac{\vec{\nabla} p}{p} \right) + \sum_{j=1}^{\upsilon} \frac{x_i\, x_j}{\rho_f D_{ij}} \left(\frac{D_j^T}{w_j} - \frac{D_i^T}{w_i} \right) \left(\frac{\vec{\nabla} T}{T} \right) \tag{60}$$

3.5.3. Estimation of diffusion coefficients

At moderate pressures, the diffusion coefficient of a binary gas mixture consisting of molecules A and B, i.e. the binary diffusion coefficient D_{AB}, is well described by the theory developed from solving the Boltzmann equation, usually credited to both Chapman and Enskog [30, 31]. In this theory, the diffusion coefficients are presented in terms of the gas mixture pressure, temperature and their molecular properties:

$$D_{AB} = \frac{3}{16} \frac{\sqrt{\dfrac{4\pi k_B T}{M_{AB}}}}{\hat{n}\, \pi\, \sigma_{AB}{}^2 \Omega_{AB}^D} f_D \tag{61}$$

In Eq.(61), M_{AB} is the reduced molecular weights, σ_{AB} is the reduced collision diameter, Ω_{AB}^D is the diffusion collision integral for the $A-B$ species, k_B is the Boltzmann constant, $\hat{n}$ is the number density of molecules and T is the absolute temperature in the mixture gas respectively. The correction term f_D, which accounts for large differences in the molecular weights of the two components, lies between 1.0 and 1.02 regardless of composition of intermolecular forces. The value of f_D is significantly different from unity only if the molecular

masses are very unequal and the light component is present in trace amounts. Even in such cases, f_D is usually between 1.0 and 1.1 [32].

3.5.3.1. Nonpolar gas mixtures

Considering ideal-gas law and the correction term f_D equal a unity, Eq.(61) is simplified to calculate the binary diffusion coefficient for a nonpolar gas mixture [30]:

$$D_{AB} = \left(\frac{0.00266}{p\,\sigma_{AB}{}^{2}\,\Omega_{AB}^{\mathrm{D}}} \right) \sqrt{\frac{T^3}{M_{AB}}} \tag{62}$$

In Eq.(62), p is the pressure in bar, T is the temperature in K, σ_{AB} is the Lennard-Jones characteristic length in $\mathring{A}$, Ω_{AB}^{D} is the dimensionless collision integral for diffusion, M_{AB} is the reduced molar mass in g/mol and D_{AB} is the binary diffusion coefficient in cm^2/s.

Coefficients of Eq.(62), for a nonpolar gas mixtures containing the $A-B$ species pair, are calculated from the values of the individual species' parameters, which can be found in gas properties tables in the literature [15, 30]. The values for a binary gas mixture are calculated using Eq.(63) to Eq.(67).

Reduced molecular mass:

$$M_{AB} = 2\left[\frac{1}{M_A} + \frac{1}{M_B} \right]^{-1} \tag{63}$$

Lennard-Jones characteristic dimension:

$$\sigma_{AB} = \frac{\sigma_A + \sigma_B}{2} \tag{64}$$

Collision integral for diffusion coefficients involving molecules A and B:

$$\left\{ \begin{aligned} \Omega_{AB}^{\mathrm{D}} = & \frac{1.06036}{\left(T_{AB}^{*}\right)^{0.1561}} + \frac{0.193}{\exp\left(0.47635 T_{AB}^{*}\right)} \\ & + \frac{1.03587}{\exp\left(1.52996 T_{AB}^{*}\right)} + \frac{1.76474}{\exp\left(3.89411 T_{AB}^{*}\right)} \end{aligned} \right\} \tag{65}$$

Reduced Temperature:

$$T_{AB}^{*} = \frac{k_B T}{\hat{\varepsilon}_{AB}} \tag{66}$$

Lennard-Jones characteristic energy:

$$\hat{\varepsilon}_{AB} = \left(\hat{\varepsilon}_A \hat{\varepsilon}_B\right)^{1/2} \tag{67}$$

3.5.3.2. Polar gas mixtures

For a gas mixture in which one or both components are polar, a modified Lennard-Jones relation such as the Stockmayer potential is often used [33]. An alternative method for diffusion coefficients estimation in binary mixtures containing polar components was suggested by Brokaw [34], that proposed the application of Eq.(62) with modified parameters to account for the polarity effects:

$$D_{AB}^{polar} = \left(\frac{0.00266}{p\left(\sigma_{AB}^{polar}\right)^2 \Omega_{AB}^{\mathrm{D},polar}} \right) \sqrt{\frac{T^3}{M_{AB}}} \tag{68}$$

Variables in Eq.(68) have the same units as presented in Eq.(62). Corrections on these variables to account for polar gas effects in the mixture are presented in Eq.(69) to Eq.(76):

Lennard-Jones characteristic dimension for a polar gas mixture:

$$\sigma_{AB}^{polar} = \left(\sigma_A^{polar}\sigma_B^{polar}\right)^{1/2} \tag{69}$$

Collision integral for diffusion coefficients involving polar molecules A and/or B:

$$\Omega_{AB,polar}^{\mathrm{D}} = \Omega_{AB}^{\mathrm{D}} + \frac{0.19\left(\delta_{AB}^{*}\right)^2}{T_{AB}^{*}} \tag{70}$$

Stockmayer potential for a polar gas mixture:

$$\delta_{AB}^{*} = \left(\delta_A^{polar}\delta_B^{polar}\right)^{1/2} \tag{71}$$

Reduced Temperature for a polar gas mixture:

$$T_{AB}^{*} = \frac{k_B T}{\hat{\varepsilon}_{AB}^{polar}} \tag{72}$$

Lennard-Jones characteristic energy:

$$\frac{\hat{\varepsilon}_{AB}^{polar}}{k_B} = \left(\frac{\hat{\varepsilon}_A^{polar}}{k_B}\frac{\hat{\varepsilon}_B^{polar}}{k_B}\right)^{1/2} \tag{73}$$

The parameters used in Eq.(68) to Eq.(73) for individual species must consider the polar effects. Thus, for individual species in a polar mixture, Stockmayer potential, Lennard-Jones characteristic dimension, and the Lennard-Jones characteristic energy for each individual species are calculated respectively from the equations below:

$$\delta_i^{polar} = \frac{1.94\times10^3\left(\hat{\mu}_i\right)^2}{V_i^b\,T_i^b} \quad ; \quad i = specie\,A, specie\,B \tag{74}$$

$$\sigma_i^{polar} = \left(\frac{1.585 V_i^b}{1+1.3\left(\delta_i^{polar}\right)^2} \right)^{1/3} \quad ; \quad i = specie\,A, specie\,B \tag{75}$$

$$\frac{\hat{\varepsilon}_i^{polar}}{k_B} = 1.18\left[1+1.3\left(\delta_i^{polar}\right)^2\right] T_i^b \quad ; \quad i = specie\,A, specie\,B \tag{76}$$

In Eq.(74), Eq.(75) and Eq.(76), $\hat{\mu}_i$ is the dipole moment of species i in $debyes$, V_i^b is the liquid molar volume at the normal boiling point in cm^3/mol and T_i^b is the normal boiling point (at 1 atm) in K.

3.5.4. Momentum conservation

In the dusty-gas model derivation, the momentum balance equation is applied on the species, including the "dust" species. The balance of forces has to be zero to keep the hypothesis of motionless dust valid. For the convective motion in the porous media, creeping flow is assumed. The inertial terms in the equation of motion for the DGM, e.g., turbulence, are ignored, and the flow is characterized by the fluid's viscosity coefficient. As a result of the DGM derivation, a complete set of equations for steady-state mass transport is generated.

Consequently, in the pellet domain, the relevant equation of motion can be obtained by multiplying the DGM equation, Eq.(55) , by x_i and summing over all species. With this procedure, the terms involving D_{ij} sum to zero by symmetry, and the terms involving $\left(\alpha'_{ij}\right)_{tr}$ vanish because $\left(\alpha'_{ij}\right)_{tr} = -\left(\alpha'_{ji}\right)_{tr}$. Hence, the resulting momentum equation for the pellet is:

$$\sum_{i=1}^{\upsilon} \frac{\overrightarrow{N}_i}{D_{iK}} = \frac{1}{k_B T}\left(c\overrightarrow{F} - \overrightarrow{\nabla} p\right)\left(1 + \frac{pB_O}{\mu_f}\sum_{j=1}^{\upsilon}\frac{x_i}{D_{iK}}\right) - \left(\frac{c_d}{c'}\right)\left[\sum_{j=1}^{\upsilon} c_i\left(\alpha'_{id}\right)_{tr}\right]\overrightarrow{\nabla}\ln T \tag{77}$$

The momentum equation for the pellet domain, Eq.(77), contains the assumptions made for the DGM derivation, including invariability with respect to time. Considering that diffusion is usually a very slow process, i.e., diffusion velocities considerably smaller than the bulk flow velocity in a reactor, the application of Eq.(77) in time-dependent problems generates negligible deviations. In this work, steady-state assumption is considered. Thus, the effects of the application of Eq.(77) in time-dependent problems will not be investigated.

For the reactor domain, the momentum conservation equations are derived from the RTT. In this case, the intensive property for the fluid system M_f is the velocity of the fluid, therefore the extensive property of the system is the momentum of the fluid. From the Newton's law of motion, variation in the momentum quantity is equal to the sum of external forces in the system, as shown in Eq.(78):

$$M_f = \vec{v} \wedge \frac{dM^t_{CSys}(t)}{dt} = \frac{d\left(\delta m_f \overrightarrow{v_f}\right)(t)}{dt} = \sum \overrightarrow{F_{external}} \tag{78}$$

Considering only gravity, pressure and friction as external forces acting on the fluid phase in the reactor domain, the RTT application proceeds as follows:

$$\begin{aligned}
&\iiint_{CV} \rho_f \vec{g}\, dV^t - \iint_{CS} \phi_A \left(p - \vec{\vec{T}}'\right) \cdot \vec{n} dA^t = \frac{\partial}{\partial t}\left(\iiint_{CV} \overrightarrow{v_f} \rho_f \phi_V dV^t\right) + \iint_{CS} \rho_f \overrightarrow{v_f}\,\overrightarrow{v_f} \cdot \vec{n}\, \phi_A dA^t \underset{Divergence}{\overset{Gauss'}{\Rightarrow}} \\
&\iiint_{CV} \rho_f \vec{g}\, dV^t - \iiint_{CV} \vec{\nabla} \cdot \left[\phi_A \left(p - \vec{\vec{T}}'\right)\right] dV^t = \frac{\partial}{\partial t}\left(\iiint_{CV} \phi_V \overrightarrow{v_f} \rho_f dV^t\right) + \iiint_{CV} \vec{\nabla} \cdot \left(\phi_A \rho_f \overrightarrow{v_f}\,\overrightarrow{v_f}\right) dV^t \underset{Theorem}{\overset{Leibnitz's}{\Rightarrow}} \\
&\iiint_{CV} \rho_f \vec{g}\, dV^t - \iiint_{CV} \vec{\nabla} \cdot \left[\phi_A \left(p - \vec{\vec{T}}'\right)\right] dV^t = \iiint_{CV} \frac{\partial}{\partial t}\left(\phi_V \overrightarrow{v_f} \rho_f\right) dV^t + \iiint_{CV} \vec{\nabla} \cdot \left(\phi_A \rho_f \overrightarrow{v_f}\,\overrightarrow{v_f}\right) dV^t
\end{aligned} \tag{79}$$

In the left hand side of the resulting equation (last line of Eq.(79)), the first term is the field force due to gravity. The second term accounts for the variation of the linear momentum on the control volume; caused by the pressure on the fluid and by the fluid's viscous resistance to

the movement. Consequently, for the reactor domain, considering an infinitesimally small control volume CV with volume $\delta V^t = \iiint_{CV} dV^t$, one can have the expression as following:

$$\frac{\sum \overrightarrow{F_{external}}}{\delta V^t} = \rho_f \vec{g} - \vec{\nabla}\cdot\left(\phi_A p\right) + \vec{\nabla}\cdot\left(\phi_A \vec{\vec{T}}'\right) = \frac{\partial}{\partial t}\left(\phi_V \rho_f \overrightarrow{v_f}\right) + \vec{\nabla}\cdot\left(\phi_A \rho_f \overrightarrow{v_f}\,\overrightarrow{v_f}\right) \overset{\frac{\partial}{\partial t}\left(\phi_V \rho_f\right)+\vec{\nabla}\cdot\left(\phi_A \rho_f \overrightarrow{v_f}\right)=0}{\underset{Eq.of\ continuity}{\Leftrightarrow}}$$

$$\Leftrightarrow \quad \rho_f \vec{g} - \vec{\nabla}\cdot\left(\phi_A p\right) + \vec{\nabla}\cdot\left(\phi_A \vec{\vec{T}}'\right) = \rho_f \left[\frac{\partial}{\partial t}\left(\phi_V \overrightarrow{v_f}\right) + \vec{\nabla}\cdot\left(\phi_A \overrightarrow{v_f}\,\overrightarrow{v_f}\right)\right] \tag{80}$$

Equation (80), which has embedded the continuity equation, is the resulting equation of motion in its vectorial-tensorial form. It can be further developed in terms of the coordinate system of the problem. For example, the three components of Eq.(80) in Cartesian coordinates (x, y and z respectively) are:

$$\begin{aligned} &\rho_f g_x - \frac{\partial}{\partial x}\left(\phi_A p\right) + \left[\frac{\partial}{\partial x}\left(\phi_A T_{xx}\right) + \frac{\partial}{\partial y}\left(\phi_A T_{yx}\right) + \frac{\partial}{\partial z}\left(\phi_A T_{zx}\right)\right] = \\ &= \rho_f \left[\frac{\partial}{\partial t}\left(\phi_V v_{f,x}\right) + v_{f,x}\frac{\partial}{\partial x}\left(\phi_A v_{f,x}\right) + v_{f,y}\frac{\partial}{\partial y}\left(\phi_A v_{f,x}\right) + v_{f,z}\frac{\partial}{\partial z}\left(\phi_A v_{f,x}\right)\right] \end{aligned} \tag{81}$$

$$\begin{aligned} &\rho_f g_y - \frac{\partial}{\partial x}\left(\phi_A p\right) + \left[\frac{\partial}{\partial x}\left(\phi_A T_{xy}\right) + \frac{\partial}{\partial y}\left(\phi_A T_{yy}\right) + \frac{\partial}{\partial z}\left(\phi_A T_{zy}\right)\right] = \\ &= \rho_f \left[\frac{\partial}{\partial t}\left(\phi_V v_{f,y}\right) + v_{f,x}\frac{\partial}{\partial x}\left(\phi_A v_{f,y}\right) + v_{f,y}\frac{\partial}{\partial y}\left(\phi_A v_{f,y}\right) + v_{f,z}\frac{\partial}{\partial z}\left(\phi_A v_{f,y}\right)\right] \end{aligned} \tag{82}$$

$$\begin{aligned} &\rho_f g_z - \frac{\partial}{\partial x}\left(\phi_A p\right) + \left[\frac{\partial}{\partial x}\left(\phi_A T_{xz}\right) + \frac{\partial}{\partial y}\left(\phi_A T_{yz}\right) + \frac{\partial}{\partial z}\left(\phi_A T_{zz}\right)\right] = \\ &= \rho_f \left[\frac{\partial}{\partial t}\left(\phi_V v_{f,z}\right) + v_{f,x}\frac{\partial}{\partial x}\left(\phi_A v_{f,z}\right) + v_{f,y}\frac{\partial}{\partial y}\left(\phi_A v_{f,z}\right) + v_{f,z}\frac{\partial}{\partial z}\left(\phi_A v_{f,z}\right)\right] \end{aligned} \tag{83}$$

The solution of Eq.(80) usually involves simplifications in the fluid stress tensor $\vec{\vec{T}}'$ through the hypothesis of Newtonian fluid. Additionally, although Eq.(80) is valid for turbulent

flows, it is usually not applicable in this case due to the complexity and irregularity of the fluid dynamic involved. Simulation and solution of turbulent flows are usually based on the adoption of turbulent models, such as the statistical treatment of the turbulent flow, the Reynolds average model, and the $k-\varepsilon$ model, among others [36].

The development of a complete model that address turbulent flows is not part of the scope of this work, although increases in the model complexity (i.e., higher dimensions) will require a thorough turbulent model. In the simulation of a 1-D system, such as the packed-bed reactor studied in this work, a momentum equation that considers both laminar and turbulent flow can be derived from the concept of friction factors on the packed bed, resulting in the well-known Ergun equation [27]:

$$\frac{d}{dz}(p) = -\frac{\left(\rho_f v_{f,z}\right)}{\rho_f d_{pellet}}\left(\frac{1-\phi_V}{\phi_V^{\ 3}}\right)\left[\frac{150\left(1-\phi_V\right)\mu_f}{d_{pellet}} + 1.75\left(\rho_f v_{f,z}\right)\right] \tag{84}$$

3.5.5. Energy conservation

The application of the RTT procedure to derive the energy conservation equations considers the total energy E^t as an extensive property of the system. Consequently, the intensive property or the system, M, is the total energy per mass of the composite system. The total energy of the system is composed by the system's internal energy, kinetic energy and potential energy, and the total energy of the system only varies when heat Q is added to the system and/or work W is done by the system (1st Law of Thermodynamics).

Considering respectively the fluid and the solid subsystems for the pellet domain, these concepts can be presented in the RTT formulation format as follows:

$$\left\{M_{CSys}^{t}=E_{f}^{t}\right\}\Rightarrow\left\{\begin{array}{l}M_{f}=\dfrac{E_{f}^{t}}{m_{f}}=U_{f}+\dfrac{1}{2}\vec{v}_{f}\cdot\vec{v}_{f}-\vec{g}\cdot\vec{r}\wedge\dfrac{dM_{f}^{t}(t)}{dt}=\dfrac{dE_{f}^{t}(t)}{dt}=\dfrac{dQ_{f}}{dt}+\dfrac{dW_{f}}{dt}=\\=\left[\begin{array}{l}\left[-\iiint\limits_{CV}q_{f\to s}\varepsilon_{I}dV^{t}-\iint\limits_{CS}\overrightarrow{q_{f}}\cdot\vec{n}\varepsilon_{A}dA^{t}-\iint\limits_{CS}\sum_{i=1}^{\upsilon}\dfrac{1}{M_{i}}\tilde{h}_{f,i}\vec{j}_{i}\cdot\vec{n}\varepsilon_{A}dA^{t}\right]+\\\left[-\iiint\limits_{CV}w_{f\to s}\varepsilon_{I}dV^{t}-\iint\limits_{CS}\left[\mathrm{T}'_{f}\cdot\overrightarrow{v_{f}}\right]\cdot\vec{n}\varepsilon_{A}dA^{t}-\iint\limits_{CS}\left(p_{f}\overrightarrow{v_{f}}\right)\cdot\vec{n}\varepsilon_{A}dA^{t}\right]\end{array}\right]\end{array}\right\}\tag{85}$$

$$\left\{M_{CSys}^{t}=E_{s}^{t}\right\}\Rightarrow\left\{\begin{array}{l}M_{s}=\dfrac{E_{s}^{t}}{m_{s}}=U_{s}-\vec{g}\cdot\vec{r}\wedge\dfrac{dM_{s}^{t}(t)}{dt}=\dfrac{dE_{s}^{t}(t)}{dt}=\dfrac{dQ_{s}}{dt}+\dfrac{dW_{s}}{dt}=\\=\left[\iiint\limits_{CV}q_{f\to s}\varepsilon_{I}dV^{t}-\iint\limits_{CS}\overrightarrow{q_{s}}\cdot\vec{n}\left(1-\varepsilon_{A}\right)dA^{t}\right]\end{array}\right\}\tag{86}$$

The difference between Eq.(85) and Eq.(86) lies on the fact that the solid phase is considered to be static. Therefore, for the studied system, there is no work done by the solid, or transport of energy due to solid species diffusion or convection.

The RTT for the fluid phase proceeds as follows:

$$\begin{array}{l}\left[\begin{array}{l}\left[-\iiint\limits_{CV}q_{f\to s}\varepsilon_{I}dV^{t}-\iint\limits_{CS}\overrightarrow{q_{f}}\cdot\vec{n}\varepsilon_{A}dA^{t}-\iint\limits_{CS}\sum_{i=1}^{\upsilon}\dfrac{1}{M_{i}}\tilde{h}_{f,i}\vec{j}_{i}\cdot\vec{n}\varepsilon_{A}dA^{t}\right]+\\\left[-\iiint\limits_{CV}w_{f\to s}\varepsilon_{I}dV^{t}-\iint\limits_{CS}\left[\mathrm{T}'_{f}\cdot\overrightarrow{v_{f}}\right]\cdot\vec{n}\varepsilon_{A}dA^{t}-\iint\limits_{CS}\left(p_{f}\overrightarrow{v_{f}}\right)\cdot\vec{n}\varepsilon_{A}dA^{t}\right]\end{array}\right]=\\\qquad=\left[\begin{array}{l}\dfrac{\partial}{\partial t}\left(\iiint\limits_{CV}\left(U_{f}+\dfrac{1}{2}\overrightarrow{v_{f}}\cdot\overrightarrow{v_{f}}-\vec{g}\cdot\vec{r}\right)\rho_{f}\varepsilon_{V}dV^{t}\right)+\\+\iint\limits_{CS}\left(U_{f}+\dfrac{1}{2}\overrightarrow{v_{f}}\cdot\overrightarrow{v_{f}}-\vec{g}\cdot\vec{r}\right)\rho_{f}\overrightarrow{v_{f}}\cdot\vec{n}\,\varepsilon_{A}dA^{t}\end{array}\right]\end{array}\tag{87}$$

Using the Gauss' divergence theorem on Eq.(87):

$$\begin{bmatrix}\left[-\iiint\limits_{CV} q_{f\to s}\varepsilon_I dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left(\varepsilon_A\overrightarrow{q_f}\right)dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left(\varepsilon_A\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,i}\overrightarrow{j_i}\right)dV^t\right]+\\ \left[-\iiint\limits_{CV} w_{f\to s}\varepsilon_I dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left[\varepsilon_A\left(\mathrm{T}'_f\cdot\overrightarrow{v_f}\right)\right]dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left(\varepsilon_A p_f\overrightarrow{v_f}\right)dV^t\right]\end{bmatrix}$$
$$=\begin{bmatrix}\frac{\partial}{\partial t}\left(\iiint\limits_{CV}\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\rho_f\varepsilon_V dV^t\right)+\\ +\iiint\limits_{CV}\vec{\nabla}\cdot\left[\varepsilon_A\rho_f\overrightarrow{v_f}\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\right]dV^t\end{bmatrix} \tag{88}$$

Applying the Leibnitz's theorem on Eq(88):

$$\begin{bmatrix}\left[-\iiint\limits_{CV} q_{f\to s}\varepsilon_I dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left(\varepsilon_A\overrightarrow{q_f}\right)dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left(\varepsilon_A\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,j}\overrightarrow{j_i}\right)dV^t\right]+\\ \left[-\iiint\limits_{CV} w_{f\to s}\varepsilon_I dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left[\varepsilon_A\left(\mathrm{T}'_f\cdot\overrightarrow{v_f}\right)\right]dV^t - \iiint\limits_{CV}\vec{\nabla}\cdot\left(\varepsilon_A p_f\overrightarrow{v_f}\right)dV^t\right]\end{bmatrix}=$$
$$=\begin{bmatrix}\left(\iiint\limits_{CV}\frac{\partial}{\partial t}\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\rho_f\varepsilon_V dV^t\right)+\\ +\iiint\limits_{CV}\vec{\nabla}\cdot\left[\varepsilon_A\rho_f\overrightarrow{v_f}\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\right]dV^t\end{bmatrix} \tag{89}$$

Considering an infinitesimally small control volume CV with volume $\delta V^t=\iiint\limits_{CV} dV^t$,

the resulting energy equation from the RTT for the fluid systems in the pellet domain is:

$$\frac{\partial}{\partial t}\left[\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\rho_f\varepsilon_V\right]+\vec{\nabla}\cdot\left[\varepsilon_A\rho_f\overrightarrow{v_f}\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\right]=$$
$$=-q_{f\to s}\varepsilon_I-\vec{\nabla}\cdot\left(\varepsilon_A\overrightarrow{q_f}\right)-\vec{\nabla}\cdot\left(\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,j}\varepsilon_A\overrightarrow{j_i}\right)-w_{f\to s}\varepsilon_I-\vec{\nabla}\cdot\left[\varepsilon_A\left(\mathrm{T}'_f\cdot\overrightarrow{v_f}\right)\right]-\vec{\nabla}\cdot\left(\varepsilon_A p_f\overrightarrow{v_f}\right) \tag{90}$$

Applying the same procedure for the solid system in the pellet domain, the resulting energy equation from the RTT is:

$$\frac{\partial}{\partial t}\left[\left(1-\varepsilon_V\right)\left(U_s-\vec{g}\cdot\vec{r}\right)\rho_s\right]=q_{f\to s}\varepsilon_I-\vec{\nabla}\cdot\left[\left(1-\varepsilon_A\right)\overrightarrow{q_s}\right] \tag{91}$$

Similarly, the respective energy equations for fluid and the solid subsystems in the reactor domain are:

$$\begin{aligned}&\frac{\partial}{\partial t}\left[\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\rho_f\phi_V\right]+\vec{\nabla}\cdot\left[\phi_A\rho_f\overrightarrow{v_f}\left(U_f+\frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}-\vec{g}\cdot\vec{r}\right)\right]=\\&=-q_{f\to s}\phi_I-\vec{\nabla}\cdot\left(\phi_A\overrightarrow{q_f}\right)-\vec{\nabla}\cdot\left(\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,i}\phi_A\overrightarrow{j_i}\right)-w_{f\to s}\phi_I-\vec{\nabla}\cdot\left[\phi_A\left(\mathrm{T}'_f\cdot\overrightarrow{v_f}\right)\right]-\vec{\nabla}\cdot\left(\phi_A p_f\overrightarrow{v_f}\right)\end{aligned} \tag{92}$$

$$\frac{\partial}{\partial t}\left[\left(1-\phi_V\right)\left(U_s-\vec{g}\cdot\vec{r}\right)\rho_s\right]=q_{f\to s}\phi_I-\vec{\nabla}\cdot\left[\left(1-\phi_A\right)\overrightarrow{q_s}\right] \tag{93}$$

For the fluid subsystem inside the catalyst pellet, considering the convective transport is negligible and assuming very small diffusion velocities, the work term and the terms multiplied by the fluid velocity can be neglected. Nevertheless, these assumptions do not hold for the reactor domain. Thus, with this simplification, Eq.(90) can be written as

$$\frac{\partial}{\partial t}\left[\left(U_f-\vec{g}\cdot\vec{r}\right)\rho_f\varepsilon_V\right]=-q_{f\to s}\varepsilon_I-\vec{\nabla}\cdot\left(\varepsilon_A\overrightarrow{q_f}\right)-\vec{\nabla}\cdot\left(\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,j}\varepsilon_A\overrightarrow{j_i}\right) \tag{94}$$

which is the energy equation for the fluid phase in the pellet domain considering negligible fluid velocity and negligible work of fluid inside the pellet.

Considering negligible potential energy effects and rearranging terms on Eq.(94) as follows

$$\frac{\partial}{\partial t}\left[\left(U_f\right)\rho_f\varepsilon_V\right]=-q_{f\to s}\varepsilon_I-\vec{\nabla}\cdot\left(\varepsilon_A\overrightarrow{q_f}\right)-\left[\sum_{i=1}^{\upsilon}\frac{1}{M_i}\varepsilon_A\overrightarrow{j_i}\left(\vec{\nabla}\cdot\tilde{h}_{f,i}\right)\right]-\left[\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,i}\varepsilon_A\left(\vec{\nabla}\cdot\overrightarrow{j_i}\right)\right] \tag{95}$$

it is possible to further simplify the energy equation when constant porosity and no pressure changes occurring in the system are considered. Moreover, the last term in Eq.(95) can be substituted by using Eq.(40) and Eq.(49). Additionally, Eq.(91) and Eq.(95) are written in terms of the heat capacities by using the relations presented below:

$$\left\{\begin{array}{l} dU = C_V\, dT + \left(\kappa P - \beta T\right) V\, dp \\ \tilde{h}_{f,i} = \Delta h^0_{f,i} + \int_{T^0}^{T_f} C_{P,i}(T')dT' + V_{f,i}\left(1-\beta T_f\right) dp' \end{array}\right\} \tag{96}$$

where $\Delta h^0_{f,i}$ is the species j enthalpy of formation at T^0 and p^0.

Therefore, considering the aforementioned assumptions, the energy equation for the solid and fluid phases in the pellet domain, respectively:

$$\left(1-\varepsilon_V\right)\rho_s C_{V,s}\frac{\partial T_s}{\partial t} = q_{f\to s}\varepsilon_I - \vec{\nabla}\cdot\left[\left(1-\varepsilon_A\right)\overrightarrow{q_s}\right] \tag{97}$$

$$\begin{aligned} \varepsilon_V\left[\rho_f C_{V,f}\frac{\partial T_f}{\partial t} + U_f\frac{\partial \rho_f}{\partial t}\right] - \varepsilon_V\sum_{i=1}^{\upsilon}\frac{1}{M_i}\tilde{h}_{f,i}\frac{\partial}{\partial t}\left(w_i\rho_f\right) = -q_{f\to s}\varepsilon_I - \vec{\nabla}\cdot\left(\varepsilon_A\overrightarrow{q_f}\right) \\ -\left(\sum_{i=1}^{\upsilon}\frac{1}{M_i}\left(\varepsilon_A\overrightarrow{j_i}\right)C_{P,i}\vec{\nabla}T_f\right) - \sum_{i=1}^{\upsilon}\left(\Delta h^0_{f,i} + \int_{T^0}^{T_f} C_{P,i}(T')dT'\right)\bar{R}_i\left(1-\varepsilon_V\right)\rho_s \end{aligned} \tag{98}$$

Heat transfer terms appear in all the energy equations derived. Two terms are recurrent: a conductive heat flux term, namely $\overrightarrow{q_f}$ and $\overrightarrow{q_s}$ for the fluid system and the solid systems respectively, and an interfacial heat transfer term between the fluid and the solid phases $q_{f\to s}$, which has positive sign when the solid absorbs the heat from the fluid.

For the conductive heat flux, Fourier's law can be used to describe heat transfer in each of the individual phases, as shown below for the fluid and the solid systems respectively:

$$\overrightarrow{q_f} = -k_f\vec{\nabla}T_f \tag{99}$$

$$\overrightarrow{q_s} = -k_s\vec{\nabla}T_s \tag{100}$$

In Eq.(99) and Eq.(100), k_s and the fluid k_f are the thermal conductivity of the fluid and the solid respectively. Radiation effects are not explicitly considered in this formulation, but the adoption of a modified thermal conductivity, which includes radiation terms, is often used in reactor simulations [37].

For the pellet domain, it is important to discuss if the calculation of the temperature in each individual phases of the porous media is a reasonable approach. To develop this discussion, considering that the interfacial heat term $q_{f\to s}$ occurs only where the fluid and the solid phases are in contact, a convective heat transfer equation can be applied in this case:

$$q_{f\to s} = -\lambda \left(T_f - T_s\right) \tag{101}$$

Recalling that, in a porous media, the diameter of the pores and the molecules' free path have the same order of magnitude, it is possible to imagine that a molecule can repeatedly hit the porous wall in a short period of time, quickly gaining or releasing energy from and to the solid. Hence, the heat transfer coefficient h_f assumes very high values at the fluid/solid interface inside the pores of the catalytic pellet, going to infinity when $d_{pore} \to 0$. This fact, associated with the observation that diffusion fluxes have very slow velocities, a reasonable hypothesis is that the temperature of the fluid phase T_f and solid phase T_s will quickly reach equilibrium in the pellet domain. This leads to $q_{f\to s}$ equal zero inside the pellet, and the interfacial heat transfer term is important only in the reactor domain.

Therefore, combining Eq.(97) to Eq.(100) and considering $T_f = T_s = T$, the energy equation for the pellet as a whole has the following form:

$$\left\{\begin{array}{l}\left[\varepsilon_V \rho_f C_{V,f} + \left(1-\varepsilon_V\right)\rho_s C_{V,s}\right]\dfrac{\partial T}{\partial t} \\ +\varepsilon_V\left(U_f - \sum_{i=1}^{\upsilon}\dfrac{1}{M_i}\tilde{h}_{f,i}w_i\right)\dfrac{\partial \rho_f}{\partial t} \\ -\varepsilon_V\left(\sum_{i=1}^{\upsilon}\dfrac{1}{M_i}\tilde{h}_{f,i}\rho_f\dfrac{\partial w_i}{\partial t}\right)\end{array}\right\} = \left\{\begin{array}{l}\vec{\nabla}\cdot\left[\left(\varepsilon_A k_f + \left(1-\varepsilon_A\right)k_s\right)\vec{\nabla}T\right] \\ -\varepsilon_A\left[\sum_{i=1}^{\upsilon}\dfrac{1}{M_i}\left(\vec{j_i}\right)C_{P,i}\vec{\nabla}T\right] \\ -\left(1-\varepsilon_V\right)\rho_s\left[\sum_{i=1}^{\upsilon}\left(\Delta h_{f,i}^0 + \int_{T^0}^{T}C_{P,i}(T')dT'\right)\bar{R}_i\right]\end{array}\right\} \quad (102)$$

For the fluid phase in the reactor domain, Eq.(92) can be further simplified by using Eq.(96), Eq.(47) and Eq.(49). Considering that potential energy is negligible, the energy equation for the fluid phase in the reactor domain can be rewritten as:

$$\left\{\begin{array}{l}\left[\sum_{i=1}^{\upsilon}\left(\rho_i\phi_V C_{V,i}\right)\right]\dfrac{\partial T_f}{\partial t} \\ +\dfrac{\partial}{\partial t}\left[\rho_f\phi_V\left(\dfrac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f}\right)\right] \\ +\left[\sum_{i=1}^{\upsilon}\left(\phi_A\rho_i C_{P,i}\right)\right]\overrightarrow{v_f}\cdot\vec{\nabla}T_f \\ +\vec{\nabla}\cdot\left[\dfrac{1}{2}\phi_A\rho_f\overrightarrow{v_f}\cdot\left(\overrightarrow{v_f}\cdot\overrightarrow{v_f}\right)\right]\end{array}\right\} = \left\{\begin{array}{l}-\lambda_f\phi_I\left(T_f - T_s\right) + \sum_{i=1}^{\upsilon}\dfrac{1}{M_i}\tilde{h}_{f,i}\vec{\nabla}\cdot\left(\varepsilon_A w_i\rho_f\overrightarrow{v_f}\right) \\ +\sum_{i=1}^{\upsilon}-\tilde{h}_{f,i}\bar{R}_i\left(1-\varepsilon_V\right)\rho_s + \vec{\nabla}\cdot\left(\phi_A k_f\vec{\nabla}T_f\right) \\ +\sum_{i=1}^{\upsilon}\dfrac{1}{M_i}\tilde{h}_{f,i}\dfrac{\partial}{\partial t}\left(\varepsilon_V w_i\rho_f\right) - \sum_{i=1}^{\upsilon}\dfrac{1}{M_i}\phi_A\vec{j_i}\cdot\left(\vec{\nabla}\tilde{h}_{f,i}\right) \\ -w_{f\to s}\phi_I - \vec{\nabla}\cdot\left[\phi_A\left(\mathrm{T}'_f\cdot\overrightarrow{v_f}\right)\right]\end{array}\right\} \quad (103)$$

In Eq.(103) energy changes dues to heat transport through radiation are not considered. Nevertheless, one procedure used in simulations to account for radiation effects is to lump all heat transfer coefficients in one effective heat transfer coefficient [37], as mentioned previously. The use of Eq.(103) in simulations can be made with additional assumptions, such as negligible viscous effects, negligible kinetic effects, or negligible transport of enthalpy through diffusion, although its application is not part of this work.

4. Simulation I – Catalytic spherical pellet

In this chapter, the modeling approach, parameters and simulation results for a 1-D spherical catalytic pellet are presented. Equations developed in chapter 3 through the application of the RTT on the pellet domain are further simplified, according to the dimensionality of the problem and to the steady-state assumption. The procedure is applied to simulate a catalyst pellet used in steam methane reforming. COMSOL® Multiphysics is used to find the solution of the non-linear system of equations that describes the pellet overall behavior.

4.1. Modeling approach

The equations derived on chapter 3 through the RTT procedure considered that the system's properties are functions of the spatial coordinates and time. In spherical coordinates, this means that, for the general case

$$F = F\left(t, r, \theta, \phi\right) \tag{104}$$

where F represents any property of the studied system. For steady-state operation, F becomes independent of time t, leading to

$$F = F\left(r, \theta, \phi\right) \ \wedge \ \frac{\partial}{\partial t} F = 0 \tag{105}$$

A 3-Dimensional spherical catalytic pellet can be simplified to a 1-D model if no variations associated with the zenith and azimuth angular directions are considered:

$$F = F\left(r\right) \tag{106}$$

This means that all variables of the problem are considered only function of the sphere radius r. Thus, the application of the $\vec{\nabla}$ operator to any of the properties for the studied system leads to:

$$\vec{\nabla}F(r)=\vec{i_r}\frac{\partial}{\partial r}F(r)+\vec{i_\theta}\frac{1}{r}\frac{\partial}{\partial\theta}F(r)+\vec{i_\phi}\frac{1}{r\sin\theta}\frac{\partial}{\partial\phi}F(r)\Rightarrow\vec{\nabla}F(r)=\vec{i_r}\frac{d}{dr}F(r) \qquad (107)$$

The equations described in the next session consider this approach.

4.2. Model equations

Considering the aforementioned approach/assumptions, simplified versions of the general equations presented in chapter 3 are developed. In addition, all porous media characteristics, such as porosity and tortuosity, are considered constant throughout the pellet spatial domain.

4.2.1. Total mass balance

The total mass balance presented for the solid and liquid systems, Eq.(33) and Eq.(34) respectively, assumes the following form:

$$\frac{d}{dr}\left(\rho_f v_f\right)=0 \quad\Leftrightarrow\quad \rho_f v_f=\text{constant} \qquad (108)$$

$$\frac{d}{dr}\left(\rho_s v_s\right)=0 \quad\Leftrightarrow\quad \rho_s v_s=0 \quad \left(\text{stagnant solid phase}\right) \qquad (109)$$

Considering stagnant solid phase and no accumulation inside the pellet, the center of mass of both fluid and solid must be fixed in the space. Thus, the amount of mass in the catalyst pellet is constant and, therefore, the mass average velocity of the fluid $\overrightarrow{v_f}$ is zero in this system:

$$\overrightarrow{v_f}=0 \qquad (110)$$

4.2.2. Component mass balance

The component mass balance presented in both Eq.(40) and Eq.(46) are defined per volume of porous media. Since the equations for the spherical pellet model are defined as functions of the radius r of the sphere, and vectors have components in the direction $\vec{i_r}$ only, adjustments on these equations are necessary to consider this assumption properly. For a sphere, the differential volume is:

$$dV^t = \left(\frac{4\pi}{3}\right) r^2 dr^t \tag{111}$$

Substituting the differential volume for the spherical pellet, Eq.(111), on Eq.(39):

$$\int_{CV} R_i r^2 dr^t - \int_{CV} \vec{\nabla} \cdot \left(\varepsilon_A \vec{j_i}\right) r^2 dr^t = \int_{CV} \frac{\partial}{\partial t}\left(\varepsilon_V w_i \rho_f\right) r^2 dr^t + \int_{CV} \vec{\nabla} \cdot \left(\varepsilon_A w_i \rho_f \vec{v_f}\right) r^2 dr^t \tag{112}$$

For an infinitesimally small spherical control volume with radius r, such that $\delta r^t = \int_{CV} dr^t$, and considering the mass reaction rate as shown in Eq.(49), the component mass balance for the diffusive mass flux $\vec{j_i}$ and the total mass flux $\vec{n_i}$, previously presented in Eq.(40) and Eq.(46) respectively, can be rewritten as follows:

$$M_i \bar{R}_i \left(1-\varepsilon_V\right) \rho_s r^2 = \varepsilon_A \left[\frac{d}{dr}\left(r^2 j_i\right) + \frac{d}{dr}\left(r^2 w_i \rho_f v_f\right)\right] \quad ; \quad i = 1, \upsilon \tag{113}$$

$$M_i \bar{R}_i \left(1-\varepsilon_V\right) \rho_s r^2 = \varepsilon_A \frac{d}{dr}\left(r^2 \vec{n_i}\right) \quad ; \quad i = 1, \upsilon \tag{114}$$

4.2.3. Diffusion equation

The DGM model is used in the pellet domain to account for the diffusion effects, as presented in Eq.(55). Considering the thermal diffusion effects negligible, the DGM assumes the following form:

$$\sum_{j=1}^{\upsilon}\frac{1}{D_{ij}^{eff}}\left(\frac{x_j}{M_i}\overrightarrow{n_i^p}-\frac{x_i}{M_j}\overrightarrow{n_j^p}\right)+\left(\frac{1}{M_iD_{iK}}\right)\overrightarrow{n_i^p}=-\left[c_{Tot}\frac{d}{dr}\left(x_i\right)+\frac{x_i}{\tilde{R}T}\left(1+\frac{p}{D_{iK}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \qquad (115)$$

In this work, a steam methane reforming reaction is considered. The reaction kinetics model used was developed by Xu and Froment [38], and involves 5 species: CO, CO_2, H_2, H_2O and CH_4. Thus, the application of Eq.(115) in this work generates a 5 equation system that are expressed explicitly for the mass fluxes as follows:

$$\sum_{j=1}^{\upsilon}\frac{1}{D_{ij}^{eff}}\left(\frac{x_j}{M_i}\overrightarrow{n_i^p}-\frac{x_i}{M_j}\overrightarrow{n_j^p}\right)+\left(\frac{1}{M_iD_{iK}}\right)\overrightarrow{n_i^p}=-\left[c_{Tot}\frac{d}{dr}\left(x_i\right)+\frac{x_i}{\tilde{R}T}\left(1+\frac{p}{D_{iK}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \quad \overset{\upsilon=5}{\Rightarrow}$$

$$\left\{\begin{array}{l}
\left[\begin{array}{l}\left(\frac{x_1}{M_2D_{12}^{eff}}\overrightarrow{n_2^p}-\frac{x_2}{M_1D_{12}^{eff}}\overrightarrow{n_1^p}\right)+\left(\frac{x_1}{M_3D_{13}^{eff}}\overrightarrow{n_3^p}-\frac{x_3}{M_1D_{13}^{eff}}\overrightarrow{n_1^p}\right)+\\ \left(\frac{x_1}{M_4D_{14}^{eff}}\overrightarrow{n_4^p}-\frac{x_4}{M_1D_{14}^{eff}}\overrightarrow{n_1^p}\right)+\left(\frac{x_1}{M_5D_{15}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_1D_{15}^{eff}}\overrightarrow{n_1^p}\right)-\frac{1}{M_1D_{1K}^{eff}}\overrightarrow{n_1^p}\end{array}\right]=\left[\begin{array}{l}c_{Tot}\frac{d}{dr}\left(x_1\right)+\\ \frac{x_1}{\tilde{R}T}\left(1+\frac{p}{D_{1K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\end{array}\right]\vec{i_r}\\
\left[\begin{array}{l}-\left(\frac{x_1}{M_2D_{12}^{eff}}\overrightarrow{n_2^p}-\frac{x_2}{M_1D_{12}^{eff}}\overrightarrow{n_1^p}\right)+\left(\frac{x_2}{M_3D_{23}^{eff}}\overrightarrow{n_3^p}-\frac{x_3}{M_2D_{23}^{eff}}\overrightarrow{n_2^p}\right)+\\ \left(\frac{x_2}{M_4D_{24}^{eff}}\overrightarrow{n_4^p}-\frac{x_4}{M_2D_{24}^{eff}}\overrightarrow{n_2^p}\right)+\left(\frac{x_2}{M_5D_{25}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_2D_{25}^{eff}}\overrightarrow{n_2^p}\right)-\frac{1}{M_2D_{2K}^{eff}}\overrightarrow{n_2^p}\end{array}\right]=\left[\begin{array}{l}c_{Tot}\frac{d}{dr}\left(x_2\right)+\\ \frac{x_2}{\tilde{R}T}\left(1+\frac{p}{D_{2K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\end{array}\right]\vec{i_r}\\
\left[\begin{array}{l}-\left(\frac{x_1}{M_3D_{13}^{eff}}\overrightarrow{n_3^p}-\frac{x_3}{M_1D_{13}^{eff}}\overrightarrow{n_1^p}\right)-\left(\frac{x_2}{M_3D_{23}^{eff}}\overrightarrow{n_3^p}-\frac{x_3}{M_2D_{23}^{eff}}\overrightarrow{n_2^p}\right)+\\ \left(\frac{x_3}{M_4D_{34}^{eff}}\overrightarrow{n_4^p}-\frac{x_4}{M_3D_{34}^{eff}}\overrightarrow{n_3^p}\right)+\left(\frac{x_3}{M_5D_{35}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_3D_{35}^{eff}}\overrightarrow{n_3^p}\right)-\frac{1}{M_3D_{3K}^{eff}}\overrightarrow{n_3^p}\end{array}\right]=\left[\begin{array}{l}c_{Tot}\frac{d}{dr}\left(x_3\right)+\\ \frac{x_3}{\tilde{R}T}\left(1+\frac{p}{D_{3K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\end{array}\right]\vec{i_r}\\
\left[\begin{array}{l}-\left(\frac{x_1}{M_4D_{14}^{eff}}\overrightarrow{n_4^p}-\frac{x_4}{M_1D_{14}^{eff}}\overrightarrow{n_1^p}\right)-\left(\frac{x_2}{M_4D_{24}^{eff}}\overrightarrow{n_4^p}-\frac{x_4}{M_2D_{24}^{eff}}\overrightarrow{n_2^p}\right)-\\ \left(\frac{x_3}{M_4D_{34}^{eff}}\overrightarrow{n_4^p}-\frac{x_4}{M_3D_{34}^{eff}}\overrightarrow{n_3^p}\right)+\left(\frac{x_4}{M_5D_{45}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_4D_{45}^{eff}}\overrightarrow{n_4^p}\right)-\frac{1}{M_4D_{4K}^{eff}}\overrightarrow{n_4^p}\end{array}\right]=\left[\begin{array}{l}c_{Tot}\frac{d}{dr}\left(x_4\right)+\\ \frac{x_4}{\tilde{R}T}\left(1+\frac{p}{D_{4K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\end{array}\right]\vec{i_r}\\
\left[\begin{array}{l}-\left(\frac{x_1}{M_5D_{15}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_1D_{15}^{eff}}\overrightarrow{n_1^p}\right)-\left(\frac{x_2}{M_5D_{25}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_2D_{25}^{eff}}\overrightarrow{n_2^p}\right)-\\ \left(\frac{x_3}{M_5D_{35}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_3D_{35}^{eff}}\overrightarrow{n_3^p}\right)-\left(\frac{x_4}{M_5D_{45}^{eff}}\overrightarrow{n_5^p}-\frac{x_5}{M_4D_{45}^{eff}}\overrightarrow{n_4^p}\right)-\frac{1}{M_5D_{5K}^{eff}}\overrightarrow{n_5^p}\end{array}\right]=\left[\begin{array}{l}c_{Tot}\frac{d}{dr}\left(x_5\right)+\\ \frac{x_5}{\tilde{R}T}\left(1+\frac{p}{D_{5K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\end{array}\right]\vec{i_r}
\end{array}\right\} \qquad (116)$$

The system of equations in (116) is then written in matrix format:

$$\begin{bmatrix} A_1 & \frac{x_1}{M_2 D_{12}^{eff}} & \frac{x_1}{M_3 D_{13}^{eff}} & \frac{x_1}{M_4 D_{14}^{eff}} & \frac{x_1}{M_2 D_{15}^{eff}} \\ \frac{x_2}{M_1 D_{12}^{eff}} & A_2 & \frac{x_2}{M_3 D_{23}^{eff}} & \frac{x_2}{M_4 D_{24}^{eff}} & \frac{x_2}{M_5 D_{25}^{eff}} \\ \frac{x_3}{M_1 D_{13}^{eff}} & \frac{x_3}{M_2 D_{23}^{eff}} & A_3 & \frac{x_3}{M_4 D_{34}^{eff}} & -\frac{x_5}{D_{45}} \\ \frac{x_4}{M_1 D_{14}^{eff}} & \frac{x_4}{M_2 D_{24}^{eff}} & \frac{x_4}{M_3 D_{34}^{eff}} & A_4 & -\frac{x_5}{D_{45}} \\ \frac{x_5}{M_1 D_{15}^{eff}} & \frac{x_5}{M_2 D_{25}^{eff}} & \frac{x_5}{M_3 D_{35}^{eff}} & \frac{x_5}{M_4 D_{45}^{eff}} & A_4 \end{bmatrix} \cdot \begin{bmatrix} \overrightarrow{n_1^p} \\ \overrightarrow{n_2^p} \\ \overrightarrow{n_3^p} \\ \overrightarrow{n_4^p} \\ \overrightarrow{n_5^p} \end{bmatrix} = \begin{bmatrix} \left[c_{Tot} \frac{d}{dr}(x_1) + \frac{x_1}{\tilde{R}T}\left(1 + \frac{p}{D_{1K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_2) + \frac{x_2}{\tilde{R}T}\left(1 + \frac{p}{D_{2K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_3) + \frac{x_3}{\tilde{R}T}\left(1 + \frac{p}{D_{3K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_4) + \frac{x_4}{\tilde{R}T}\left(1 + \frac{p}{D_{4K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_5) + \frac{x_5}{\tilde{R}T}\left(1 + \frac{p}{D_{5K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \end{bmatrix} \tag{117}$$

where:

$$A_k = -\frac{1}{M_k}\left(\frac{x_2}{D_{k2}^{eff}} + \frac{x_3}{D_{k3}^{eff}} + \frac{x_4}{D_{k4}^{eff}} + \frac{x_5}{D_{k5}^{eff}} + \frac{1}{D_{kK}} \right)$$

Finally, the total species mass fluxes are calculated using the following expression:

$$\begin{bmatrix} \overrightarrow{n_1^p} \\ \overrightarrow{n_2^p} \\ \overrightarrow{n_3^p} \\ \overrightarrow{n_4^p} \\ \overrightarrow{n_5^p} \end{bmatrix} = \left[D^{DGM} \right] \cdot \begin{bmatrix} \left[c_{Tot} \frac{d}{dr}(x_1) + \frac{x_1}{\tilde{R}T}\left(1 + \frac{p}{D_{1K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_2) + \frac{x_2}{\tilde{R}T}\left(1 + \frac{p}{D_{2K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_3) + \frac{x_3}{\tilde{R}T}\left(1 + \frac{p}{D_{3K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_4) + \frac{x_4}{\tilde{R}T}\left(1 + \frac{p}{D_{4K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \\ \left[c_{Tot} \frac{d}{dr}(x_5) + \frac{x_5}{\tilde{R}T}\left(1 + \frac{p}{D_{5K}} \frac{B_O}{\mu_f}\right) \frac{d}{dr}(p) \right] \vec{i}_r \end{bmatrix} \tag{118}$$

Where, in Eq.(118), the following applies:

$$
\begin{cases}
\left[D^{DGM} \right] = \begin{bmatrix}
A_1 & \frac{x_1}{M_2 D_{12}^{eff}} & \frac{x_1}{M_3 D_{13}^{eff}} & \frac{x_1}{M_4 D_{14}^{eff}} & \frac{x_1}{M_2 D_{15}^{eff}} \\
\frac{x_2}{M_1 D_{12}^{eff}} & A_2 & \frac{x_2}{M_3 D_{23}^{eff}} & \frac{x_2}{M_4 D_{24}^{eff}} & \frac{x_2}{M_5 D_{25}^{eff}} \\
\frac{x_3}{M_1 D_{13}^{eff}} & \frac{x_3}{M_2 D_{23}^{eff}} & A_3 & \frac{x_3}{M_4 D_{34}^{eff}} & -\frac{x_5}{D_{45}} \\
\frac{x_4}{M_1 D_{14}^{eff}} & \frac{x_4}{M_2 D_{24}^{eff}} & \frac{x_4}{M_3 D_{34}^{eff}} & A_4 & -\frac{x_5}{D_{45}} \\
\frac{x_5}{M_1 D_{15}^{eff}} & \frac{x_5}{M_2 D_{25}^{eff}} & \frac{x_5}{M_3 D_{35}^{eff}} & \frac{x_5}{M_4 D_{45}^{eff}} & A_5
\end{bmatrix}^{-1} \\
A_k = -\frac{1}{M_k} \left(\frac{x_1}{D_{k2}^{eff}} + \frac{x_2}{D_{k2}^{eff}} + \frac{x_3}{D_{k3}^{eff}} + \frac{x_4}{D_{k4}^{eff}} + \frac{x_5}{D_{k5}^{eff}} + \frac{1}{D_{kK}} \right) \quad \wedge \quad \frac{x_i}{D_{kk}^{eff}} = 0
\end{cases}
\tag{118a}
$$

The estimation of the parameters on Eq.(118) are detailed on chapters 3.5.1 and 3.5.3. Additionally, the following relations are valid throughout this work:

$$
c_{Tot} = \sum_{i=1}^{\upsilon} c_i \quad ; \quad c_i = x_i \, c_{Tot} \quad ; \quad \sum_{i=1}^{\upsilon} x_i = 1 \tag{119}
$$

4.2.4. Momentum balance

As previously presented in chapter 3.5.4, the momentum balance in the pellet is embedded in the species' fluxes calculated by Eq.(118), and is represented by the sum of the mass fluxes equations over all species in the mixture. Nevertheless, the mass velocity of the fluid is already determined for the pellet model, as shown in Eq.(110). Therefore, one can define an expression to map pressure changes inside the pellet, such as the ideal gas law equation of state:

$$
p = c_{Tot} \tilde{R} T \tag{120}
$$

4.2.5. Energy balance

The energy balance equation that considers the solid and the liquid systems as a whole, Eq.(102), is derived per unit of pellet volume. The same procedure applied on the species balance equation needs to be applied to the energy balance. Thus, substituting the differential volume expression, Eq.(111), on the Eq.(89):

$$\begin{bmatrix} \left[-\int_{CV} q_{f\to s}\varepsilon_I r^2 dr^t - \int_{CV} \vec{\nabla}\cdot\left(\varepsilon_A \overrightarrow{q_f}\right) r^2 dr^t - \int_{CV} \vec{\nabla}\cdot\left(\varepsilon_A \sum_{i=1}^{\upsilon} \frac{1}{M_i}\tilde{h}_{f,j}\overrightarrow{j_i}\right) r^2 dr^t \right] + \\ \left[-\int_{CV} w_{f\to s}\varepsilon_I r^2 dr^t - \int_{CV} \vec{\nabla}\cdot\left[\varepsilon_A \left(\mathrm{T}'_f \cdot \overrightarrow{v_f}\right)\right] r^2 dr^t - \int_{CV} \vec{\nabla}\cdot\left(\varepsilon_A p_f \overrightarrow{v_f}\right) r^2 dr^t \right] \end{bmatrix} = \\ = \begin{bmatrix} \left(\int_{CV} \frac{\partial}{\partial t}\left(U_f + \frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f} - \vec{g}\cdot\vec{r}\right)\rho_f \varepsilon_V r^2 dr^t \right) + \\ + \int_{CV} \vec{\nabla}\cdot\left[\varepsilon_A \rho_f \overrightarrow{v_f}\left(U_f + \frac{1}{2}\overrightarrow{v_f}\cdot\overrightarrow{v_f} - \vec{g}\cdot\vec{r}\right)\right] r^2 dr^t \end{bmatrix} \tag{121}$$

Considering the simplification process showed in chapter 3.5.5, associated with the assumptions for the spherical pellet, the energy equation for the spherical pellet has the following expression:

$$\vec{\nabla}\cdot\left[r^2\left(\varepsilon_A k_f + \left(1-\varepsilon_A\right)k_s\right)\vec{\nabla}T\right] - \left[\sum_{i=1}^{\upsilon}\frac{1}{M_i}\left(\varepsilon_A \overrightarrow{j_i}\right) C_{P,i}\vec{\nabla}\left(r^2 T\right)\right] = \\ = \left[\sum_{i=1}^{\upsilon}\left(\Delta h^0_{f,i} + \int_{T^0}^{T} C_{P,i}(T')dT'\right)\left(1-\varepsilon_V\right)\rho_s \overline{R}_i\right] r^2 \tag{122}$$

Furthermore, considering the constant thermal conductivity for both fluid and the solid, and neglecting the transport of energy through diffusion, the final form used in the spherical pellet model is:

$$\frac{d}{dr}\left[r^2\left(\varepsilon_A k_f + \left(1-\varepsilon_A\right)k_s\right)\frac{dT}{dr}\right] = \left[\sum_{i=1}^{\upsilon}\left(\Delta h^0_{f,i} + \int_{T^0}^{T} C_{P,i}(T')dT'\right)\left(1-\varepsilon_V\right)\rho_s \overline{R}_i\right] r^2 \tag{123}$$

4.3. Boundary conditions

The system of differential equations formed by Eq.(114), Eq.(118), Eq.(120) and Eq.(123), defines the model equations for the pellet and is summarized in the box below:

Species balance

$$M_1 \bar{R}_1 \left(1-\varepsilon_V\right)\rho_s r^2 = \varepsilon_A \frac{d}{dr}\left(r^2 \overrightarrow{n_1}\right) = \frac{d}{dr}\left(r^2 \overrightarrow{n_1^p}\right)$$

$$M_2 \bar{R}_2 \left(1-\varepsilon_V\right)\rho_s r^2 = \varepsilon_A \frac{d}{dr}\left(r^2 \overrightarrow{n_2}\right) = \frac{d}{dr}\left(r^2 \overrightarrow{n_2^p}\right)$$

$$M_3 \bar{R}_3 \left(1-\varepsilon_V\right)\rho_s r^2 = \varepsilon_A \frac{d}{dr}\left(r^2 \overrightarrow{n_3}\right) = \frac{d}{dr}\left(r^2 \overrightarrow{n_3^p}\right)$$

$$M_4 \bar{R}_4 \left(1-\varepsilon_V\right)\rho_s r^2 = \varepsilon_A \frac{d}{dr}\left(r^2 \overrightarrow{n_4}\right) = \frac{d}{dr}\left(r^2 \overrightarrow{n_4^p}\right)$$

$$M_5 \bar{R}_5 \left(1-\varepsilon_V\right)\rho_s r^2 = \varepsilon_A \frac{d}{dr}\left(r^2 \overrightarrow{n_5}\right) = \frac{d}{dr}\left(r^2 \overrightarrow{n_5^p}\right)$$

Mass fluxes

$$\begin{bmatrix} \overrightarrow{n_1^p} \\ \overrightarrow{n_2^p} \\ \overrightarrow{n_3^p} \\ \overrightarrow{n_4^p} \\ \overrightarrow{n_5^p} \end{bmatrix} = \left[D^{DGM}\right] \cdot \begin{bmatrix} \left[c_{Tot}\frac{d}{dr}\left(x_1\right) + \frac{x_1}{RT}\left(1+\frac{p}{D_{1K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \\ \left[c_{Tot}\frac{d}{dr}\left(x_2\right) + \frac{x_2}{RT}\left(1+\frac{p}{D_{2K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \\ \left[c_{Tot}\frac{d}{dr}\left(x_3\right) + \frac{x_3}{RT}\left(1+\frac{p}{D_{3K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \\ \left[c_{Tot}\frac{d}{dr}\left(x_4\right) + \frac{x_4}{RT}\left(1+\frac{p}{D_{4K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \\ \left[c_{Tot}\frac{d}{dr}\left(x_5\right) + \frac{x_5}{RT}\left(1+\frac{p}{D_{5K}}\frac{B_O}{\mu_f}\right)\frac{d}{dr}\left(p\right)\right]\vec{i_r} \end{bmatrix}$$

Momentum (pressure from I.G.)

$$\overrightarrow{v_f} = 0$$

$$p = c_{Tot} RT$$

Energy:

$$\left(\varepsilon_A k_f + \left(1-\varepsilon_A\right)k_s\right)\frac{d^2T}{dr^2} = \left[\sum_{i=1}^{5}\left(\Delta h_{f,i}^0 + \int_{T^0}^{T} C_{P,i}\left(T'\right)dT'\right)\left(1-\varepsilon_V\right)\rho_s \bar{R}_i\right]$$

$i = 1,5$

For the spherical pellet, two boundaries are considered. At the center of the spherical pellet ($r = 0$), fluxes are considered constant due to symmetry:

$$@\, r = 0$$

$$\frac{d}{dr}\left(x_i\right) = 0$$

$$\frac{d}{dr}\left(c_i\right) = 0$$

$$\frac{d}{dr}\left(\overrightarrow{n_i^p}\right) = 0$$

$$\frac{d}{dr}\left(T\right) = 0$$

$$\frac{d}{dr}\left(p\right) = 0$$

$$i = 1,5$$

As a first approximation, variables at the external surface of the 1-Dimensional spherical pellet ($r = r_{pellet}$) have the respective value of the property at the bulk flow, although more precise models that account for interfacial resistance, such as the adoption of a heat transfer coefficient on the energy boundary, can be developed.

$$@\, r = r_{pellet}$$

$$x_i = x_i^{bulk}$$

$$c_i = c_i^{bulk}$$

$$T = T^{bulk}$$

$$p = p^{bulk}$$

$$i = 1,5$$

4.4. Simulation parameters

4.4.1. Geometrical parameters from the pellet porous size distribution

Experiments performed in catalyst pellets used in steam reforming reactors [12] generated the following pore size distribution:

Table 1: Pore size distribution of catalyst

Group k	Pore Diameter Å	Δ Volume of Voids Δ V_g, cm^3/g
1	30	0.0040
2	30-50	0.0250
3	50-80	0.0262
4	80-120	0.0379
5	120-200	0.0573
6	200-500	0.0388
7	500-1,000	0.0190
8	1,000-10,000	0.0290
9	10,000-100,000	0.0136
-	Sum	0.2508

Considering a spherical pellet composed only by cylindrical pores, it is possible to find values the geometrical parameters needed for the simulation based on the values presented on Table 1 (adapted from [12]).

An estimation of the volumetric porosity ε_V can be determined using the density of the catalyst ρ_s and the sum of the voids specific volume ΔV_g :

$$\varepsilon_V = \rho_s \sum_{k=1}^{9} \Delta V_{g,k} \tag{124}$$

For each pore diameter group presented in Table 1, an average pore diameter can be determined as follows:

$$d_{pore,k} = \sqrt{\frac{\left[\left(d_{pore,k}^{ub}\right)^2 + \left(d_{pore,k}^{lb}\right)^2\right]}{2}} \tag{125}$$

Following the same approach, for each group in Table 1 the length of the cylindrical pore can be calculated as:

$$L_{pore,k} = \Delta V_{g,k}\rho_s \cdot \left[\frac{16}{3}\frac{\left(r_{pellet}\right)^3}{\left(d_{pore,k}\right)^2}\right] \tag{126}$$

An estimation of the number of holes outside the catalyst pellet can be calculated for each group using the length of the cylindrical pore, the tortuosity factor and the pellet radius:

$$N_{holes,k} = \frac{L_{pore,k}}{\tau\left(r_{pellet}\right)} = \frac{\Delta V_{g,k}}{\tau}\rho_s \cdot \left[\frac{16}{3}\frac{\left(r_{pellet}\right)^2}{\left(d_{pore,k}\right)^2}\right] \tag{127}$$

The total area of pore diameter holes for each the group is:

$$A_{holes,k} = N_{holes,k}\left[\frac{\pi\left(d_{pore,k}\right)^2}{4}\right] = \frac{\Delta V_{g,k}}{\tau}\rho_s \cdot \left[\frac{4\pi}{3}\left(r_{pellet}\right)^2\right] \tag{128}$$

Therefore, the area void fraction ε_A can be estimated by adding the total area of holes in each group divided by the spherical pellet surface area:

$$\varepsilon_A = \frac{\sum_{k=1}^{9} A_{holes,k}}{A_{Pellet}} = \frac{\sum_{k=1}^{9}\frac{\Delta V_{g,k}}{\tau}\rho_{cat} \cdot \left[\frac{4\pi}{3}\left(r_{pellet}\right)^2\right]}{4\pi\left(r_{pellet}\right)^2} = \frac{\rho_{cat}\left(\sum_{k=1}^{9}\Delta V_{g,k}\right)}{3\tau} = \frac{\varepsilon_V}{3\tau} \tag{129}$$

Using the length of the cylindrical pore and the mean pore diameter in each group it is possible to calculate the lateral area of the cylindrical pores for each diameter group:

$$A_{lateral,k} = h_{pore} \cdot \pi d_{pore,k} = \Delta V_{g,i} \rho_s \cdot \left[\frac{16}{3} \frac{\left(r_{pellet}\right)^3}{\left(d_{pore,k}\right)^2} \right] \pi d_{pore,k} = \Delta V_{g,k} \rho_s \cdot \left[\frac{16\pi}{3} \frac{\left(r_{pellet}\right)^3}{d_{pore,k}} \right] \quad (130)$$

The area to volume interfacial factor ε_I can be estimated by summing the cylindrical pore lateral area over all groups and dividing by the pellet total volume:

$$\varepsilon_I = \frac{\sum_{k=1}^{9} A_{lateral,k}}{V_{Pellet}} = \frac{\sum_{k=1}^{9} \Delta V_{g,k} \rho_s \left[\frac{16\pi}{3} \frac{\left(r_{pellet}\right)^3}{d_{pore,k}} \right]}{\frac{4\pi}{3} \left(r_{pellet}\right)^3} = 4\rho_s \sum_{i=1}^{9} \left(\frac{\Delta V_{g,k}}{d_{pore,k}} \right) \quad (131)$$

The tortuosity factor value ranges from 2 to 6 for most catalyst pellets and must be determined experimentally[15]. Xu and Froment [12] estimated a value of 3.54 for the tortuosity in a steam reforming catalyst.

Parameters estimated by using the procedure presented in this section are shown in Table 2. Parameters not informed in this chapter can be found in the attachments section and were extracted from the literature [12, 20, 22, 38, 39].

Table 2: Estimated geometrical parameters for the catalyst pellet

Parameter	Value	Units
Voids volume density	0.2508	cm^3/g
Pellet radius	1.0000	cm
Pellet volume	4.1888	cm^3
Pellet weight	9.8654	g
Voids volume	2.4743	cm^3
Mean pore diameter	4.86E-05	cm
Tortuosity	3.54	dimensionless
ε_V	0.5907	dimensionless
ε_A	0.0556	dimensionless
ε_I	191.653	m^2/cm^3

4.4.2. Reaction equations

The steam reforming reaction equations presented in this work were developed by Xu and Froment [38]. The proposed scheme uses the three main reactions of the steam reforming reaction mechanism to characterize the production of hydrogen and by-products. These three reactions are:

$$\begin{aligned}
&(I) \quad CH_4 + H_2O \rightleftarrows CO + 3H_2 \qquad -\Delta H_{298} = -206.1 \quad [kJ/mol] \\
&(II) \quad CO + H_2O \rightleftarrows CO_2 + H_2 \qquad -\Delta H_{298} = +41.15 \quad [kJ/mol] \\
&(III) \quad CH_4 + 2H_2O \rightleftarrows CO_2 + 4H_2 \qquad -\Delta H_{298} = -165.0 \quad [kJ/mol]
\end{aligned} \tag{132}$$

The reaction rates for reactions (*I*) to (*III*) were defined experimentally, and can be described by the following expressions:

$$\bar{R}_1 = \frac{k_1}{p_{H_2}^{2.5}} \left(p_{CH_4} p_{H_2O} - \frac{p_{H_2}^3 p_{CO}}{K_1} \right) \frac{1}{(DEN)^2} \tag{133}$$

$$\bar{R}_2 = \frac{k_2}{p_{H_2}} \left(p_{CO} p_{H_2O} - \frac{p_{H_2} p_{CO_2}}{K_2} \right) \frac{1}{(DEN)^2} \tag{134}$$

$$\bar{R}_3 = \frac{k_3}{p_{H_2}^{3.5}} \left(p_{CH_4} p_{H_2O}^2 - \frac{p_{H_2}^4 p_{CO_2}}{K_3} \right) \frac{1}{(DEN)^2} \tag{135}$$

$$DEN = 1 + K_{CO} p_{CO} + K_{H_2} p_{H_2} + K_{CH_4} p_{CH_4} + K_{H_2O} \left(\frac{p_{H_2O}}{p_{H_2}} \right) \tag{136}$$

In Eq.(133) to Eq.(136), p_i is the partial pressure of substance i, and K_j and k_j are the adsorption constant and the rate coefficient of the reaction j, respectively. The constants and functions used to model the reforming reactions are presented in Tables 3 and 4, [38].

Table 3: Parameters of the rate coefficients and adsorption constants for related Arrhenius or Van't Hoff Equations

Rate coefficient or adsorption constant	Pre-exponential factor	Units of pre-exponential factor	Activation energy or adsorption enthalpy (kJ/mol)
k_1	4.225×10^{15}	$kmol.bar^{0.5}/(kg_{cat}.h)$	240.1
k_2	1.955×10^{6}	$kmol/(kg_{cat}.h.bar)$	67.13
k_3	1.020×10^{15}	$kmol.bar^{0.5}/(kg_{cat}.h)$	243.9
k_{CO}*	8.23×10^{-5}	bar^{-1}	−70.65*
k_{CH4}*	6.65×10^{-4}	bar^{-1}	−38.28*
k_{H2O}*	1.77×10^{5}	*dimensionless*	88.68*
k_{H2}*	6.12×10^{-9}	bar^{-1}	−82.90*

*Values corresponding to adsorption

Table 4: Reaction equilibrium constants

Equilibrium constant	Function of T(K)	Units of equilibrium constant
K_1	$\exp((26{,}830/T) + 30.114)$	bar^2
K_2	$\exp((4{,}400/T) - 4.036)$	*dimensionless*
K_3	$K_1\times K_2$	bar^2

4.5. Results and discussion – Simulation I – Catalytic pellet

The set of equations and boundary conditions presented in this chapter were modeled using COMSOL® Multiphysics (version 5.0). As a result, the following set of variables were mapped along the pellet radius for different values of the pellet's outside temperature T^{bulk},

which is considered to vary from 600K to 1200K:

- Pellet temperature T
- CH_4 molar fraction x_{CH_4}
- CH_4 concentration c_{CH_4}
- CO molar fraction x_{CO}
- CO concentration c_{CO}
- CO_2 molar fraction x_{CO_2}
- CO_2 concentration c_{CO_2}
- H_2 molar fraction x_{H_2}
- H_2 concentration c_{H_2}
- H_2O molar fraction x_{H_2O}
- H_2O concentration c_{H_2O}
- Pellet pressure p
- CH_4 mass fraction w_{CH_4}
- CH_4 mass flux $\overrightarrow{n^p_{CH_4}}$
- CO mass fraction w_{CO}
- CO mass flux $\overrightarrow{n^p_{CO}}$
- CO_2 mass fraction w_{CO_2}
- CO_2 mass flux $\overrightarrow{n^p_{CO_2}}$
- H_2 mass fraction w_{H_2}
- H_2 mass flux $\overrightarrow{n^p_{H_2}}$
- H_2O mass fraction w_{H_2O}
- H_2O mass flux $\overrightarrow{n^p_{H_2O}}$

The temperature and pressure variation along the pellet radio r is shown in Fig.(1) and Fig.(2) respectively, for a variety of values for T^{bulk}:

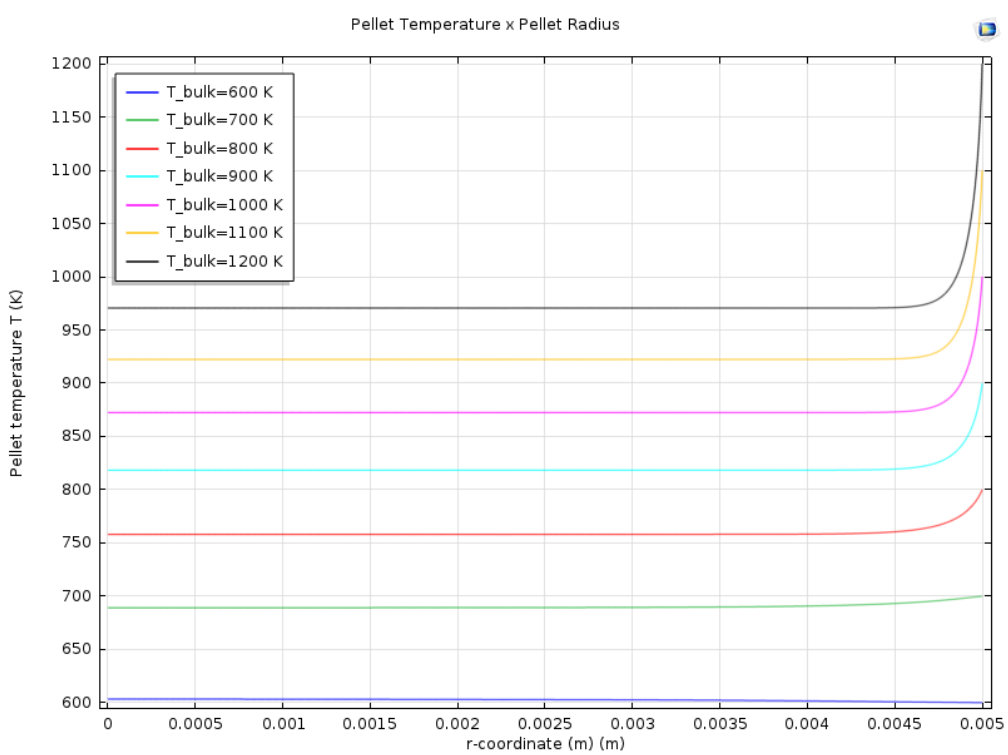

Figure 1 - Pellet temperature as a function of the pellet radius, for different T_bulk

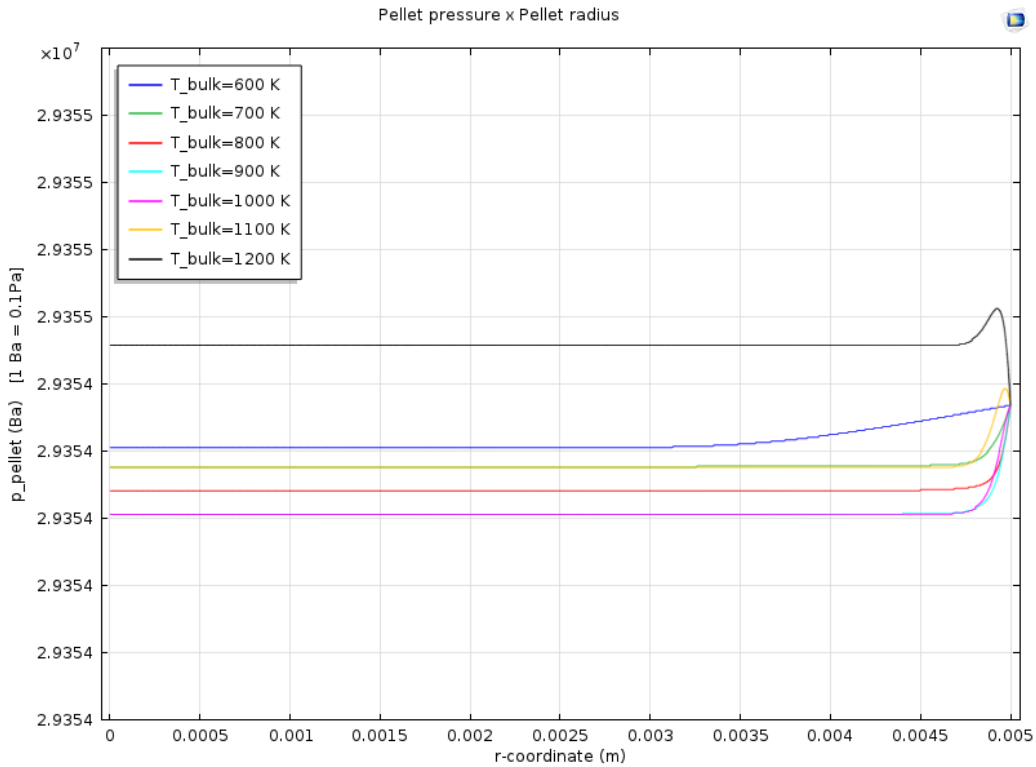

Figure 2- Pellet pressure as a function of the pellet radius, for different T_bulk

Figure (1) shows a drastic reduction of the pellet temperature T in a very small section from the pellet boundary to approximately $r_{pellet} = 4.5mm$. This fact is an indication that, for steam reforming, thermal diffusion in the pellet can have an important role and therefore should be considered in future calculations.

Pressure also changes along the pellet radius, as shown in Fig.(2). Although, variations are relatively small compared to the operating pressure. The variation range for all simulated cases is only $1,000Ba$, which is equivalent to $100Pa$ or $0.001bar$.

The correspondent production/depletion of species in the mixture, for variety of temperatures in the boundary, is expressed in terms of the species' molar fractions and are presented in Fig.(3) to Fig.(5).

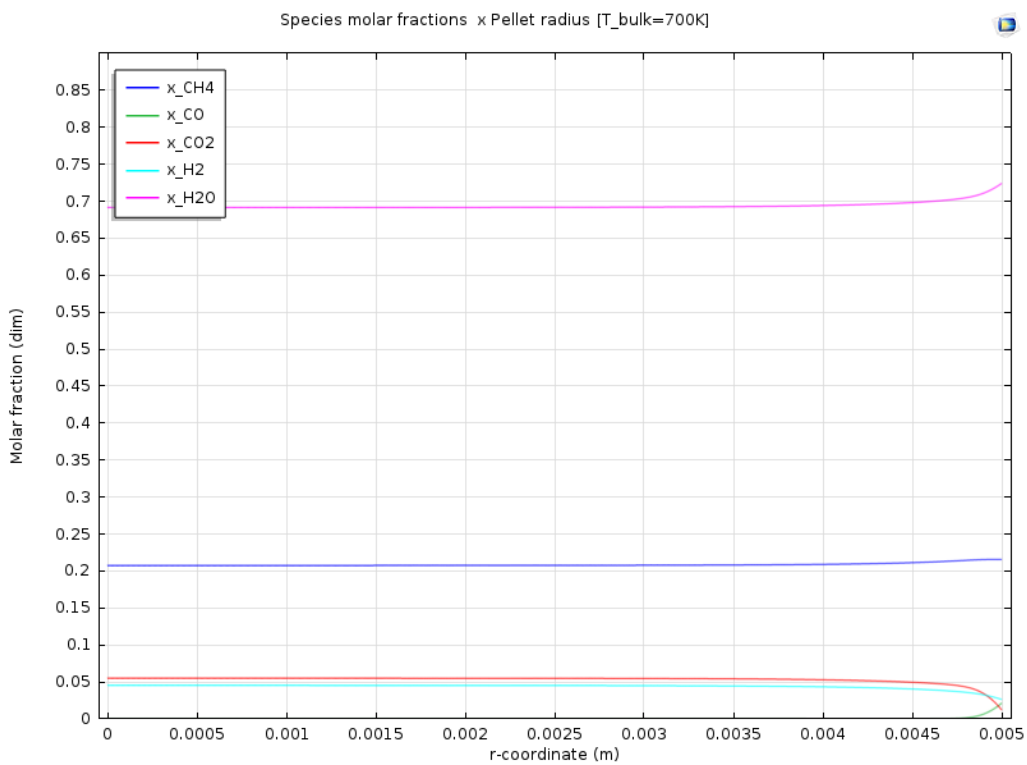

Figure 3 - Species molar fractions as a function of the pellet radius (T_bulk=700K)

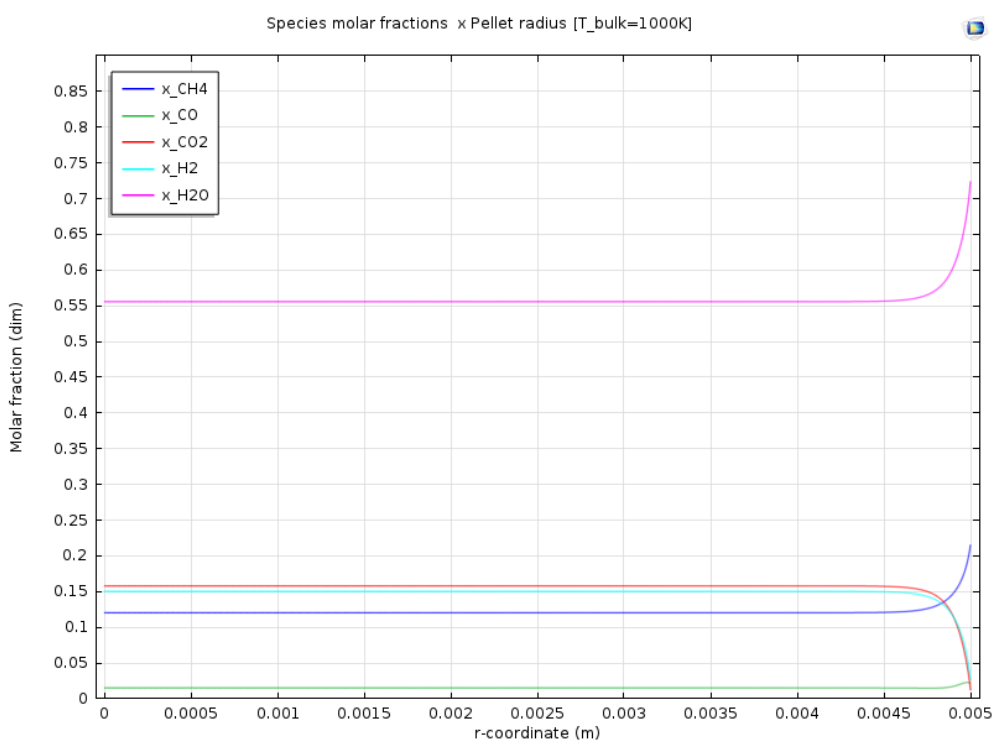

Figure 4 - Species molar fractions as a function of the pellet radius (T_bulk=1000K)

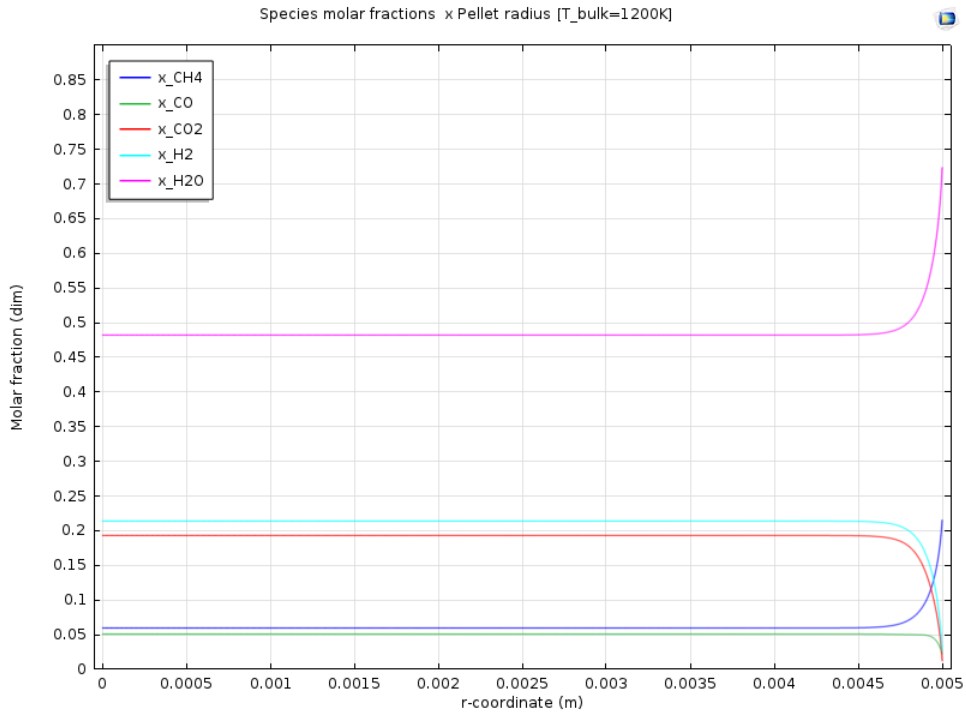

Figure 5 - Species molar fractions as a function of the pellet radius (T_bulk=1200K)

The production of hydrogen increases when the temperature at the boundary is higher. Considering the reaction rates' dependence on temperature, and the fact that the temperature decreases along the pellet radius, one can verify the correspondent hydrogen production in the limit case of constant temperature inside the pellet. This case is shown in Fig.(6).

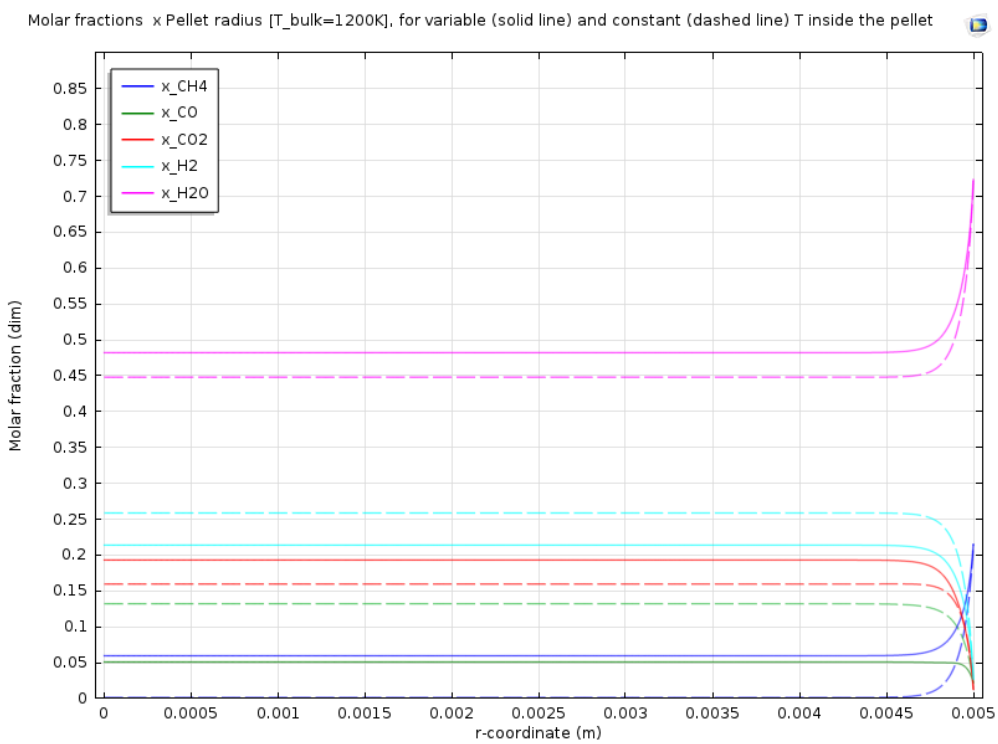

Figure 6 - Comparison between variable T and constant T along the pellet in terms of species' production

From Fig.(6), the production of H_2 increases when the temperature is kept constant throughout the pellet radius. Moreover, CH_4 is completely consumed in this case, and the reverse of the shift reaction (reaction 2) increases the CO production. Thus, the adoption of supporting materials with higher thermal conductivity can increase the effectiveness of the catalyst pellet. Furthermore, the development of catalyst pellets with high thermal conductivity can lead to a reduction in the 3 to 1 steam-methane ratio, usually adopted in the industry, in order to avoid the reverse shift reaction (due to CH_4 absence in the mixture) and, consequently, increase the H_2 output from the system.

In addition to the molar fractions, the model presented in this work can calculate relevant information in terms of concentration, mass fractions and molar fluxes for the species involved

in the steam methane reforming process. For these variables, the results for T^{bulk} equal 1000K and 1200K are presented in Fig.(7) to Fig.(14).

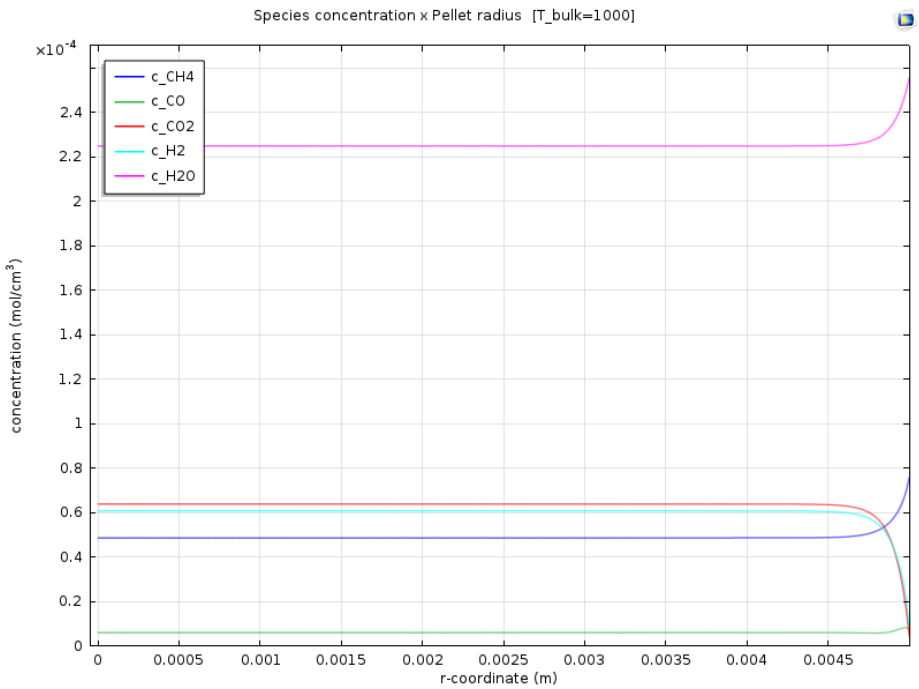

Figure 7 - Species concentrations as a function of the pellet radius (T_bulk=1000K)

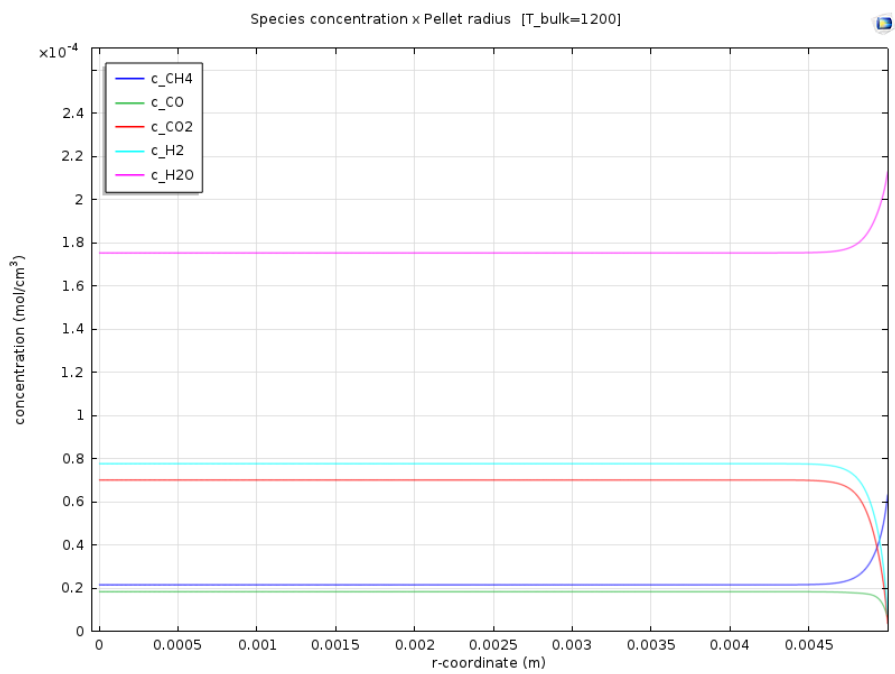

Figure 8 - Species concentrations as a function of the pellet radius (T_bulk=1200K)

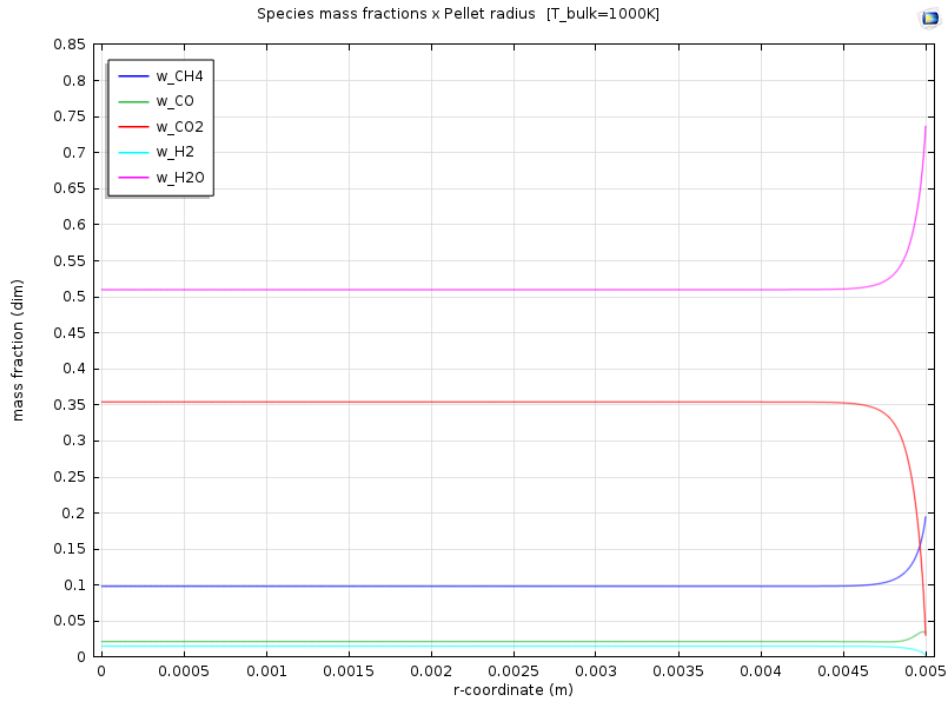

Figure 9 - Species mass fractions as a function of the pellet radius (T_bulk=1000K)

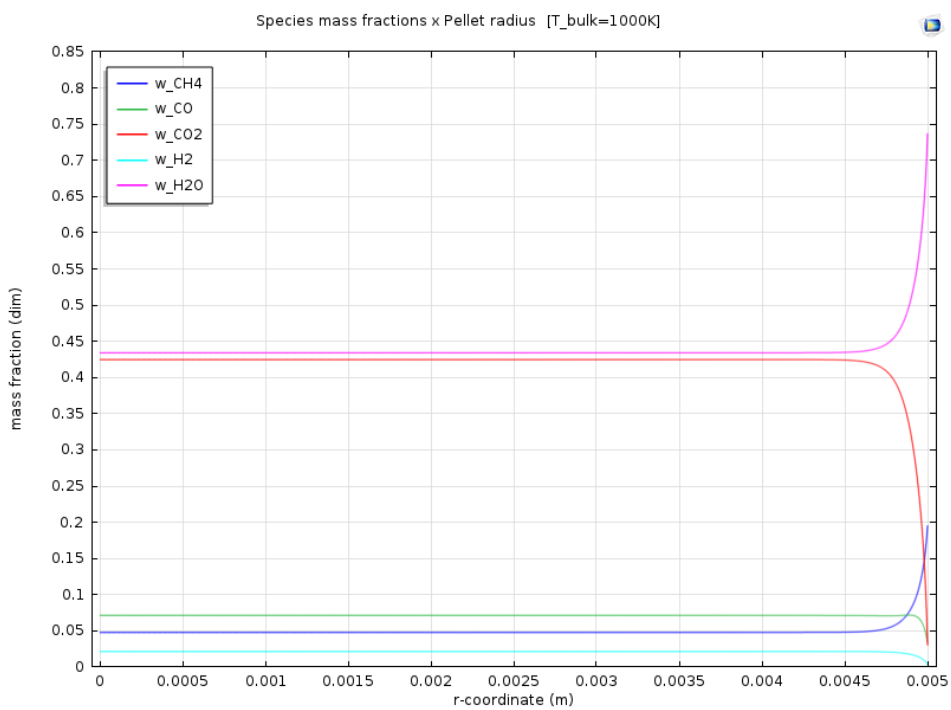

Figure 10 - Species mass fractions as a function of the pellet radius (T_bulk=1200K)

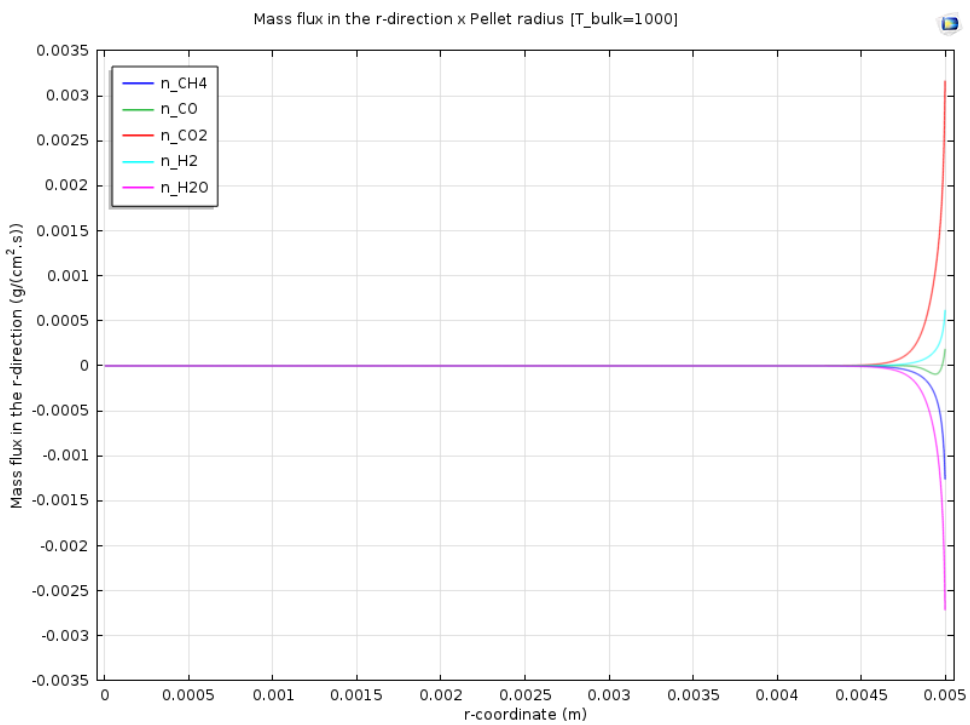

Figure 11 - Species mass flux (in the r-direction) as a function of the pellet radius (T_bulk=1000K)

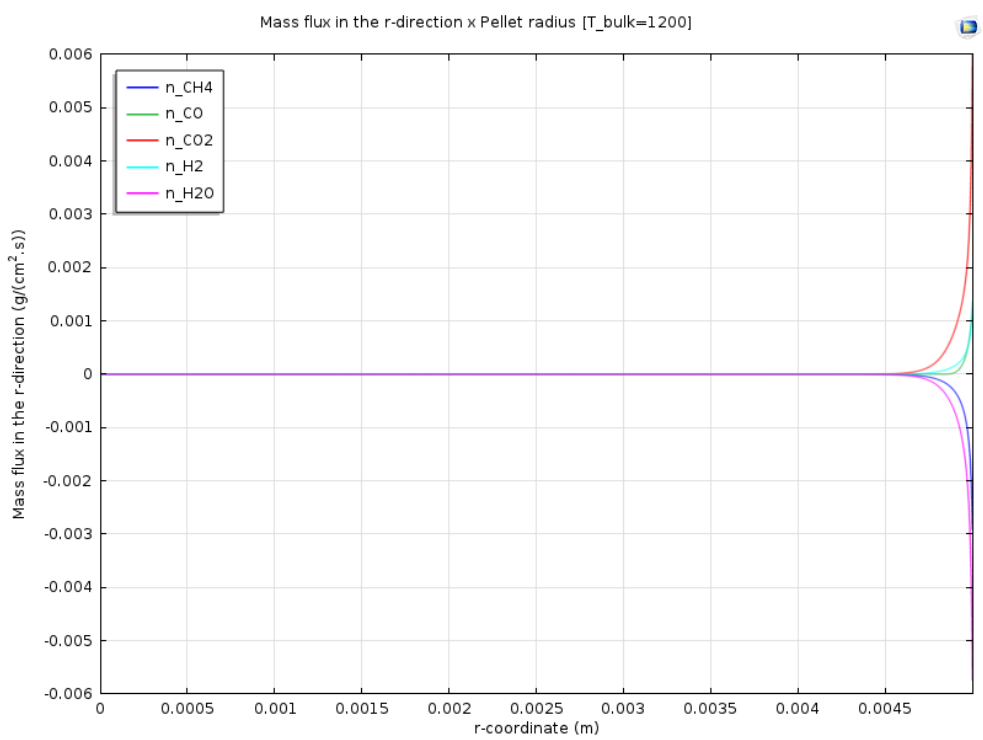

Figure 12 - Species mass flux (in the r-direction) as a function of the pellet radius (T_bulk=1200K)

Since the mass fluxes are zero from $r_{pellet} = 0\,mm$ to $r_{pellet} = 4.5\,mm$ in both Fig.(12) and Fig.(13), graphics focused on the pellet section where the mass fluxes are actually changing were generated and are presented in Fig.(13) and Fig.(14).

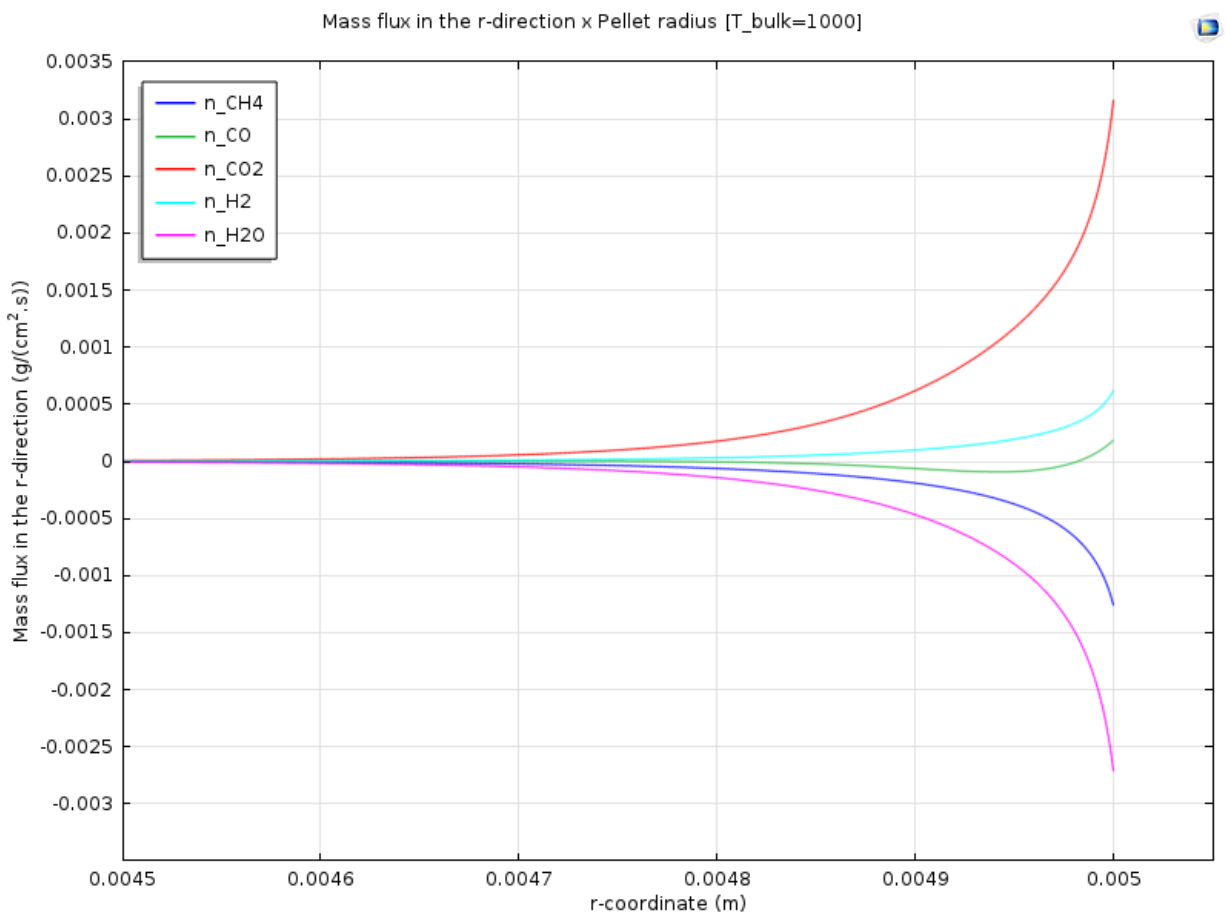

Figure 13 - Species mass flux (in the r-direction) as a function of the pellet radius (4.5mm<r_pellet < 5mm) (T_bulk=1000K)

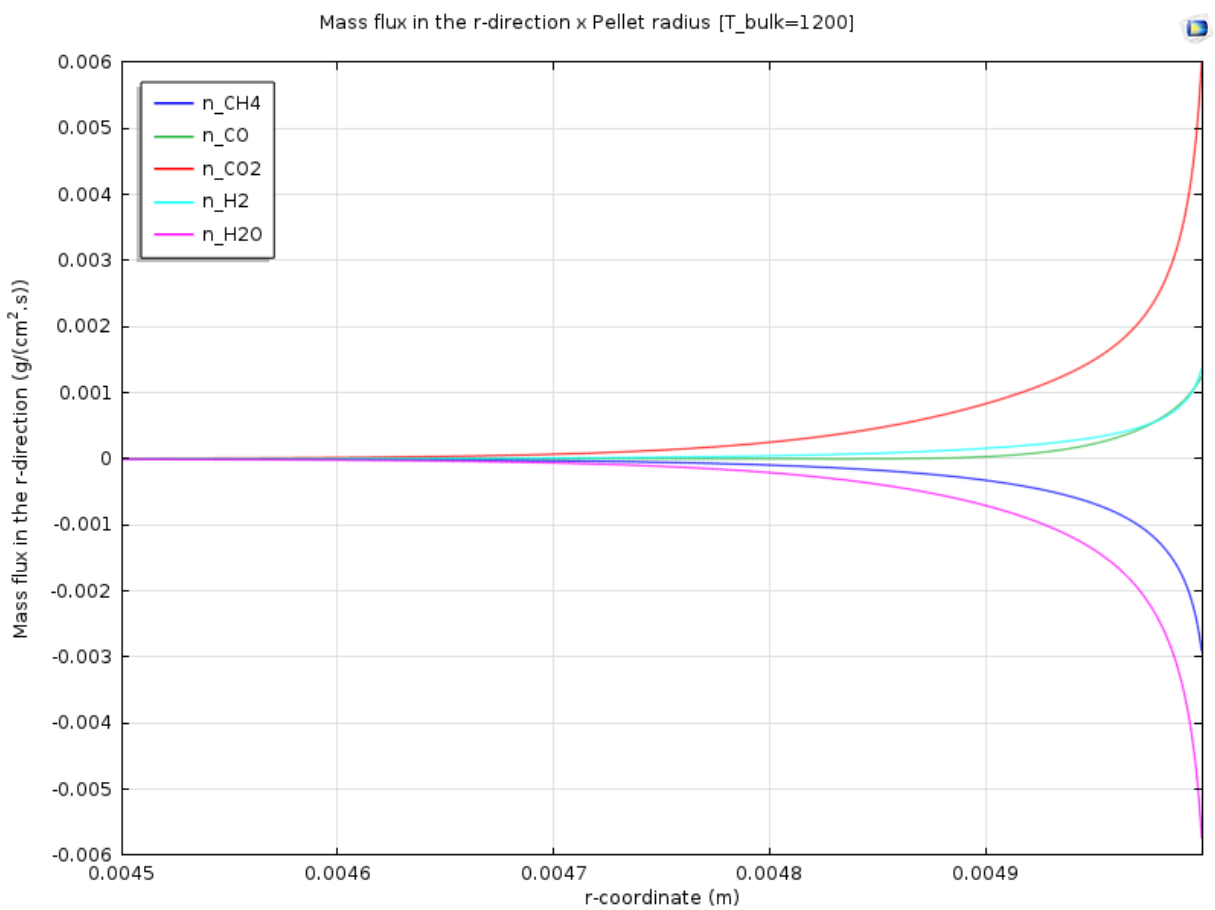

Figure 14 - Species mass flux (in the r-direction) as a function of the pellet radius (4.5mm<r_pellet < 5mm) (T_bulk=1200K)

It was noticed in Fig.13 a stagnant point for the carbon monoxide mass flux, $\overrightarrow{n_{CO}^{p}}$, around the pellet radius coordinate $r_{pellet} = 4.98\,mm$. Additionally, the effect of the temperature on the mass fluxes in this system can be visualized when comparing Fig.(13) and Fig.(14). Mass fluxes at the pellet boundary for $T^{bulk} = 1200K$ are at least two times higher than mass fluxes at $T^{bulk} = 1000K$, for any component in the mixture.

5. Simulation II – Packed-bed reactor

In addition to the catalytic pellet simulation, a packed-bed reactor (PBR) was simulated. In this chapter, the modeling approach, parameters and simulation results for a 1-D PBR, that uses the aforementioned catalytic pellet model to account for reactions, are presented. Equations developed on chapter 3, through the application of the RTT, are further simplified according to the dimensionality of the problem and the steady-state assumption, similarly to the procedure presented in chapter 4 for the catalyst pellet. The resulting equations are then applied in the simulation of a PBR producing hydrogen through steam reforming. COMSOL® Multiphysics is used to find the solution of the non-linear system of equations that describes both the pellet and the reactor's behavior.

5.1. Modeling approach

For the PBR simulation, an approach the couples a 1-D reactor with a set of 1-D pellets, distributed along the reactor length, is proposed. In this approach, each point along the length L of the 1-D reactor works as a source of information about the flow bulk flow properties, such as concentrations, mass fractions, etc., to be used by the catalyst pellet model associated with that specific point L. The catalyst pellet is then solved for the informed conditions and gives back to the reactor model, through mass fluxes, the amount of species generated or consumed in the reaction, multiplied by a factor that accounts for the amount of catalyst pellets in that specific section of the reactor. The mass fluxes from the pellet modify the fluid properties downstream. At the next point downstream the reactor, $L + \Delta L$ for example, another pellet model is solved

using the information of the bulk flow at $L + \Delta L$, and the procedure is repeated until the length of the reactor is completely covered. A schematic figure of this approach is presented in Fig.(15).

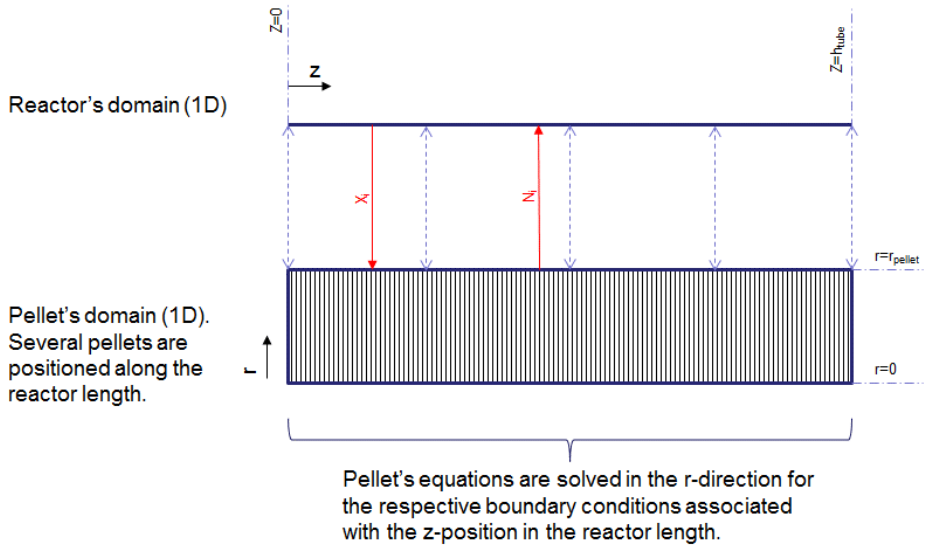

Figure 15 - Coupled reactor and pellet model approach

In the application of the proposed approach, the equations associated the reactor domain, developed through the application of the RTT procedure, are simplified considering the same set of assumptions presented in chapter 4, i.e., steady-state operation and all reactor related expressions only varies with z, the axial coordinate of the tubular reactor. Moreover, the set of equations used in the pellet model, developed in chapter 4, remains valid in this approach.

5.2. Model equations

In addition to the aforementioned assumptions, the reactor and the pellet are considered isothermal, and no pressure changes are considered inside the pellet. The adoption of energy balances for the coupled pellet and reactor case is not part of this thesis and will be pursued in future works. Simplified versions of the general equations for the reactor, presented in chapter 3,

are developed in this section. In addition, since this approach involves both pellet and reactor properties, in this section the subscripts p and r are used for the pellet and the reactor respectively, together with the applicable subscripts used previously in this work.

5.2.1. Total mass balance

The total mass balance presented for the reactor's solid and liquid systems, Eq.(35) and Eq.(36) respectively, assume the following form:

$$\frac{d}{dz}\left(\rho_{r,f} v_{r,f}\right) = 0 \quad \Leftrightarrow \quad \rho_{r,f} v_{r,f} = \text{constant} \tag{137}$$

$$\frac{d}{dz}\left(\rho_{r,s} v_{r,s}\right) = 0 \quad \Leftrightarrow \quad \rho_{r,s} v_{r,s} = 0 \quad \left(\text{stagnant solid phase}\right) \tag{138}$$

5.2.2. Component mass balance

The component mass balance for the reactor domain presented in Eq.(47) and Eq.(48) are defined per volume of reactor. Following the same procedure as presented in chapter 4.2.2, for a tubular reactor with constant cross section area A_{Tube}, the differential volume is expressed as:

$$dV^t = A_{Tube} dz^t \tag{139}$$

Substituting the differential volume for the tubular reactor, Eq.(139), on Eq.(39):

$$\int_{CV} R_{r,i} A_{Tube} dz^t - \int_{CV} \vec{\nabla} \cdot \left(\phi_A \overrightarrow{j_{r,i}}\right) A_{Tube} dz^t = \int_{CV} \vec{\nabla} \cdot \left(\phi_A w_{r,i} \rho_{r,f} \overrightarrow{v_{r,f}}\right) A_{Tube} dz^t \tag{140}$$

For an infinitesimally small tubular control volume with axial length z, such that $\delta z^t = \int_{CV} dz^t$, the component mass balance for the diffusive mass flux $\vec{j_i}$ and the total mass flux $\vec{n_i}$ can be written as follows:

$$R_{r,i} A_{Tube} = \phi_A \left[\frac{d}{dz}\left(A_{Tube} j_{r,i} \right) + \frac{d}{dz}\left(A_{Tube} w_{r,i} \rho_{r,f} v_{r,f} \right) \right] \quad ; \quad i = 1, \upsilon \tag{141}$$

$$R_{r,i} A_{Tube} = \phi_A \frac{d}{dz}\left(A_{Tube} \overrightarrow{n_{r,i}} \right) \quad ; \quad i = 1, \upsilon \tag{142}$$

Since A_{Tube} is considered constant, Eq.(141) and Eq.(142) can finally be rewritten as follows:

$$R_{r,i} = \phi_A \left[\frac{d}{dz}\left(j_{r,i} \right) + \frac{d}{dz}\left(w_{r,i} \rho_{r,f} v_{r,f} \right) \right] \quad ; \quad i = 1, \upsilon \tag{143}$$

$$R_{r,i} = \phi_A \frac{d}{dz}\left(\overrightarrow{n_{r,i}} \right) \quad ; \quad i = 1, \upsilon \tag{144}$$

For the reactor's domain, the volumetric reaction term $R_{r,i}$ in both Eq.(143) and Eq.(144) is considered to be equal to the mass flux of species i from the pellets to the reactor, or from the reactor to the pellets if i is being consumed. Therefore, at the pellet-bulk flow interface in the reactor:

$$R_{r,i} = \left(1 - \phi_V\right) \varepsilon_A \frac{A_p}{V_p} \vec{n} \cdot \overrightarrow{n_{p,i}} \Big|^{@\ pellet's\ surface} \tag{145}$$

In Eq.(145), A_p is the pellet's external area, V_p is the total volume of the pellet (solid plus voids), and $\overrightarrow{n_{p,i}}$ is the mass flux of species i in the pellet evaluated at the pellet's surface.

From the discussion in chapter 4.2.1 (mass continuity), $\overrightarrow{v_{p,f}}$ is considered equal zero in the pellet. Thus, for the pellet domain:

$$\overrightarrow{j_{p,i}} = \rho_{p,i}\left(\overrightarrow{v_{p,i}} - \overrightarrow{v_{p,f}}\right) \overset{\overrightarrow{v_{p,f}}=0}{\Rightarrow} \overrightarrow{j_{p,i}} = \overrightarrow{n_{p,i}} = \rho_{p,i}\overrightarrow{v_{p,i}} \quad ; \quad i = 1, \upsilon \tag{146}$$

$$\sum_{i=1}^{\upsilon} \overrightarrow{j_{p,i}} = 0 \tag{147}$$

Equation (146) shows that the total mass flux and the diffusive mass flux for each of the species i in the mixture are equal; and Eq.(147) comes from the definition of diffusive fluxes.

5.2.3. Diffusion equation

The diffusion equation chosen for the reactor domain is the Maxwell-Stefan equation, Eq.(60). Considering the aforementioned assumptions, the Maxwell-Stefan equation assumes the following form:

$$\sum_{j=1}^{\upsilon} \frac{x_{r,i}\, x_{r,j}}{D_{ij}} \left(\frac{1}{\rho_{r,j}} \overrightarrow{j_{r,j}} - \frac{1}{\rho_{r,i}} \overrightarrow{j_{r,i}} \right) = \left[\frac{d}{dz}\left(x_{r,i}\right) + \left(x_{r,i} - w_{r,i}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right) \right] \vec{i}_z \tag{148}$$

The treatment of Eq.(148) is similar to the one used for the DGM in the pellet. For a five species system, such as the steam methane reforming, this equation can be written in matrix format as follows:

$$\begin{bmatrix} \overrightarrow{j_{r,1}} \\ \overrightarrow{j_{r,2}} \\ \overrightarrow{j_{r,3}} \\ \overrightarrow{j_{r,4}} \\ \overrightarrow{j_{r,5}} \end{bmatrix} = \left[D^{SM} \right] \cdot \begin{bmatrix} \left[\frac{d}{dz}\left(x_{r,1}\right) + \left(x_{r,1} - w_{r,1}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right) \right]\vec{i_r} \\ \left[\frac{d}{dz}\left(x_{r,2}\right) + \left(x_{r,2} - w_{r,2}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right) \right]\vec{i_r} \\ \left[\frac{d}{dz}\left(x_{r,3}\right) + \left(x_{r,3} - w_{r,3}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right) \right]\vec{i_r} \\ \left[\frac{d}{dz}\left(x_{r,4}\right) + \left(x_{r,4} - w_{r,4}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right) \right]\vec{i_r} \\ \left[\frac{d}{dz}\left(x_{r,5}\right) + \left(x_{r,5} - w_{r,5}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right) \right]\vec{i_r} \end{bmatrix}$$

$$\text{where:} \begin{cases} \left[D^{SM} \right] = \begin{bmatrix} B_1 & \frac{x_{r,1}\,x_{r,2}}{\rho_{r,2}D_{12}} & \frac{x_{r,1}\,x_{r,3}}{\rho_{r,3}D_{13}} & \frac{x_{r,1}\,x_{r,4}}{\rho_{r,4}D_{14}} & \frac{x_{r,1}\,x_{r,5}}{\rho_{r,5}D_{15}} \\ \frac{x_{r,1}\,x_{r,2}}{\rho_{r,2}D_{12}} & B_2 & \frac{x_{r,2}\,x_{r,3}}{\rho_{r,3}D_{23}} & \frac{x_{r,2}\,x_{r,4}}{\rho_{r,4}D_{24}} & \frac{x_{r,2}\,x_{r,5}}{\rho_{r,5}D_{25}} \\ \frac{x_{r,1}\,x_{r,3}}{\rho_{r,3}D_{13}} & \frac{x_{r,2}\,x_{r,3}}{\rho_{r,3}D_{23}} & B_3 & \frac{x_{r,3}\,x_{r,4}}{\rho_{r,4}D_{34}} & \frac{x_{r,3}\,x_{r,5}}{\rho_{r,5}D_{35}} \\ \frac{x_{r,1}\,x_{r,4}}{\rho_{r,4}D_{14}} & \frac{x_{r,2}\,x_{r,4}}{\rho_{r,4}D_{24}} & \frac{x_{r,3}\,x_{r,4}}{\rho_{r,4}D_{34}} & B_4 & \frac{x_{r,4}\,x_{r,5}}{\rho_{r,5}D_{45}} \\ \frac{x_{r,1}\,x_{r,5}}{\rho_{r,5}D_{15}} & \frac{x_{r,2}\,x_{r,5}}{\rho_{r,5}D_{25}} & \frac{x_{r,3}\,x_{r,5}}{\rho_{r,5}D_{35}} & \frac{x_{r,4}\,x_{r,5}}{\rho_{r,5}D_{45}} & B_5 \end{bmatrix}^{-1} \\ B_k = -\frac{1}{\rho_{r,k}}\left(\frac{x_{r,k}\,x_{r,1}}{D_{k1}} + \frac{x_{r,k}\,x_{r,2}}{D_{k2}} + \frac{x_{r,k}\,x_{r,3}}{D_{kk}} + \frac{x_{r,k}\,x_{r,4}}{D_{k4}} + \frac{x_{r,k}\,x_{r,5}}{D_{k5}} \right) \wedge \frac{x_{r,k}\,x_{r,i}}{D_{kk}} = 0 \end{cases} \tag{149}$$

5.2.4. Momentum Equation

Ergun equation is the momentum equation for the 1-D reactor, as shown previously in Eq.(84). Since Ergun equation was developed for a 1-D system, no additional treatment on this equation is needed. For the sake of easiness for the reader, Eq.(84) is rewritten in this section as follows:

$$\frac{d}{dz}\left(p_r\right)=-\frac{\left(\rho_{r,f}v_{r,f}\right)}{\rho_{r,f}d_{pellet}}\left(\frac{1-\phi_V}{\phi_V{}^3}\right)\left[\frac{150\left(1-\phi_V\right)\mu_{r,f}}{d_{pellet}}+1.75\left(\rho_{r,f}v_{r,f}\right)\right] \tag{150}$$

5.3. Boundary conditions

In this simulation, boundary conditions for the catalyst pellet and the packed bed reactor are considered.

For the isothermal, isobaric catalyst pellet, the system of differential equations uses Eq.(114) and Eq.(118), which are is summarized in the box below:

Species balance

$$M_1\,\bar{R}_1\,\left(1-\varepsilon_V\right)\rho_s\,r^2=\varepsilon_A\frac{d}{dr}\left(r^2\overrightarrow{n_{p,1}}\right)=\frac{d}{dr}\left(r^2\overrightarrow{n_1^p}\right)$$

$$M_2\,\bar{R}_2\,\left(1-\varepsilon_V\right)\rho_s r^2=\varepsilon_A\frac{d}{dr}\left(r^2\overrightarrow{n_{p,2}}\right)=\frac{d}{dr}\left(r^2\overrightarrow{n_2^p}\right)$$

$$M_3\,\bar{R}_3\,\left(1-\varepsilon_V\right)\rho_s r^2=\varepsilon_A\frac{d}{dr}\left(r^2\overrightarrow{n_{p,3}}\right)=\frac{d}{dr}\left(r^2\overrightarrow{n_3^p}\right)$$

$$M_4\,\bar{R}_4\,\left(1-\varepsilon_V\right)\rho_s r^2=\varepsilon_A\frac{d}{dr}\left(r^2\overrightarrow{n_{p,4}}\right)=\frac{d}{dr}\left(r^2\overrightarrow{n_4^p}\right)$$

$$M_5\,\bar{R}_5\,\left(1-\varepsilon_V\right)\rho_s r^2=\varepsilon_A\frac{d}{dr}\left(r^2\overrightarrow{n_{p,5}}\right)=\frac{d}{dr}\left(r^2\overrightarrow{n_5^p}\right)$$

Mass fluxes

$$\begin{bmatrix}\overrightarrow{n_1^p}\\ \overrightarrow{n_2^p}\\ \overrightarrow{n_3^p}\\ \overrightarrow{n_4^p}\\ \overrightarrow{n_5^p}\end{bmatrix}=\left[D^{DGM}\right]\cdot\begin{bmatrix}\left[c_{Tot}\frac{d}{dr}\left(x_{p,1}\right)\right]\vec{i}_r\\ \left[c_{Tot}\frac{d}{dr}\left(x_{p,2}\right)\right]\vec{i}_r\\ \left[c_{Tot}\frac{d}{dr}\left(x_{p,3}\right)\right]\vec{i}_r\\ \left[c_{Tot}\frac{d}{dr}\left(x_{p,4}\right)\right]\vec{i}_r\\ \left[c_{Tot}\frac{d}{dr}\left(x_{p,5}\right)\right]\vec{i}_r\end{bmatrix}$$

$i=1,5$

Boundaries for the pellet in this simulation are similar to the boundaries presented in Simulation I. At the center of the spherical pellet ($r = 0$), fluxes are considered constant due to symmetry:

$$@\, r = 0$$

$$\frac{d}{dr}\left(x_{p,i}\right) = 0$$

$$\frac{d}{dr}\left(\overrightarrow{n_i^p}\right) = 0$$

$$i = 1,5$$

At the external surface of the spherical pellet ($r = r_{pellet}$), variables are considered to have the property's value at the bulk flow, although more precise models that account for interfacial resistance, such as the adoption of a heat transfer coefficient on the energy boundary, can be used.

$$@\, r = r_{pellet}$$

$$x_{p,i} = x_{r,i} \quad @ \;\; the\ correspondent\ \ z - coordinte$$

$$i = 1,5$$

For the isothermal reactor, the system of differential equations is composed by Eq.(141), Eq.(145), Eq.(149) and Eq.(150), and is summarized in the following box:

Species balance

$$\phi_A\left[\frac{d}{dz}\left(A_{Tube}\,j_{r,1}\right)+\frac{d}{dz}\left(A_{Tube}\,w_{r,1}\rho_{r,f}\,v_{r,f}\right)\right]=A_{Tube}\left(1-\phi_V\right)\varepsilon_A\frac{A_p}{V_p}\vec{n}\cdot\overrightarrow{n_{p,1}}\Big|^{@\ pellet's\ surface}$$

$$\phi_A\left[\frac{d}{dz}\left(A_{Tube}\,j_{r,2}\right)+\frac{d}{dz}\left(A_{Tube}\,w_{r,2}\rho_{r,f}\,v_{r,f}\right)\right]=A_{Tube}\left(1-\phi_V\right)\varepsilon_A\frac{A_p}{V_p}\vec{n}\cdot\overrightarrow{n_{p,2}}\Big|^{@\ pellet's\ surface}$$

$$\phi_A\left[\frac{d}{dz}\left(A_{Tube}\,j_{r,3}\right)+\frac{d}{dz}\left(A_{Tube}\,w_{r,3}\rho_{r,f}\,v_{r,f}\right)\right]=A_{Tube}\left(1-\phi_V\right)\varepsilon_A\frac{A_p}{V_p}\vec{n}\cdot\overrightarrow{n_{p,3}}\Big|^{@\ pellet's\ surface}$$

$$\phi_A\left[\frac{d}{dz}\left(A_{Tube}\,j_{r,4}\right)+\frac{d}{dz}\left(A_{Tube}\,w_{r,4}\rho_{r,f}\,v_{r,f}\right)\right]=A_{Tube}\left(1-\phi_V\right)\varepsilon_A\frac{A_p}{V_p}\vec{n}\cdot\overrightarrow{n_{p,4}}\Big|^{@\ pellet's\ surface}$$

$$\phi_A\left[\frac{d}{dz}\left(A_{Tube}\,j_{r,5}\right)+\frac{d}{dz}\left(A_{Tube}\,w_{r,5}\rho_{r,f}\,v_{r,f}\right)\right]=A_{Tube}\left(1-\phi_V\right)\varepsilon_A\frac{A_p}{V_p}\vec{n}\cdot\overrightarrow{n_{p,5}}\Big|^{@\ pellet's\ surface}$$

Mass fluxes

$$\begin{bmatrix}\overrightarrow{j_{r,1}}\\ \overrightarrow{j_{r,2}}\\ \overrightarrow{j_{r,3}}\\ \overrightarrow{j_{r,4}}\\ \overrightarrow{j_{r,5}}\end{bmatrix}=\left[D^{SM}\right]\cdot\begin{bmatrix}\left[\frac{d}{dz}\left(x_{r,1}\right)+\left(x_{r,1}-w_{r,1}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right)\right]\vec{i_r}\\ \left[\frac{d}{dz}\left(x_{r,2}\right)+\left(x_{r,2}-w_{r,2}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right)\right]\vec{i_r}\\ \left[\frac{d}{dz}\left(x_{r,3}\right)+\left(x_{r,3}-w_{r,3}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right)\right]\vec{i_r}\\ \left[\frac{d}{dz}\left(x_{r,4}\right)+\left(x_{r,4}-w_{r,4}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right)\right]\vec{i_r}\\ \left[\frac{d}{dz}\left(x_{r,5}\right)+\left(x_{r,5}-w_{r,5}\right)\left(\frac{1}{p_r}\right)\frac{d}{dz}\left(p_r\right)\right]\vec{i_r}\end{bmatrix}$$

The boundary conditions for the reactor are given in the table below, representing the value of the properties at the entrance of the reactor for $h=0$.

$@\,h=0$

$$x_{r,i}=x_{r,i}^0$$
$$c_{r,i}=c_{r,i}^0$$
$$p_r=p_{inlet}$$
$$v_f=v_{f,inlet}$$
$$\rho_f=\rho_{f,inlet}$$

$i=1,5$

The boundaries for the reactor exit, $h = h_{Tube}$, are presented in the box below:

$$@\, h = h_{Tube}$$

$$\frac{d}{dz}\left(x_{r,i}\right) = 0$$

$$\frac{d}{dz}\left(c_{r,i}\right) = 0$$

$$i = 1,5$$

5.4. Effectiveness Factor

The species' effectiveness factor η_i is defined as the flux of the species i from (or to) the pellet surface, divided by the production (or consumption) of the species i if the catalyst area was totally exposed to the pellet's surface condition:

$$\eta_i = \frac{\varepsilon_A \left(4\pi r_{pellet}{}^2\right) \overrightarrow{n_{p,i}}\Big|^{@\, pellet's\, surface}}{M_i \left(1-\varepsilon_V\right) \bar{R}_i \rho_s \left(\frac{4}{3}\pi r_{pellet}{}^3\right)} \tag{151}$$

Since the reaction rates in are given per volume of pellet [37], the use of the total area per volume parameter for the pellet is not necessary.

5.5. Simulation parameters

For the pellet, the parameters used in this simulation are the same as the ones specified in chapter 4.3. Parameters for the reactor are shown in Table 5.

Table 5: Parameters for the coupled PRF/Pellet simulation

Parameter	Value	Units	Description
Temperature	1100	K	System's temperature
Pressure	28.9	bar	System's pressure
r_{pellet}	0.5	cm	Pellet radius
c_{H2O}/c_{CH4}	3	dim	Steam to methane molar ratio (@ inlet)
c_{H2}/c_{CH4}	1.00×10^{-4}	dim	Hydrogen to methane molar ratio (@ inlet)
ρ_s	2355.2	kg/m^3	Catalyst density
τ	3.54	dim	Tortuosity factor

Parameters that are not listed in Table 5, nor in previous chapters, can be found in the attachments section and were extracted from the literature [12, 20, 22, 38, 39].

5.6. Results and discussion – Simulation II – Packed-bed reactor

The results for the packed-bed reactor simulation are presented in this section. The system of differential equations defined in the previous section was solved using COMSOL® Multiphysics for a variety of temperatures at the reactor, ranging from 600K to 1100K.

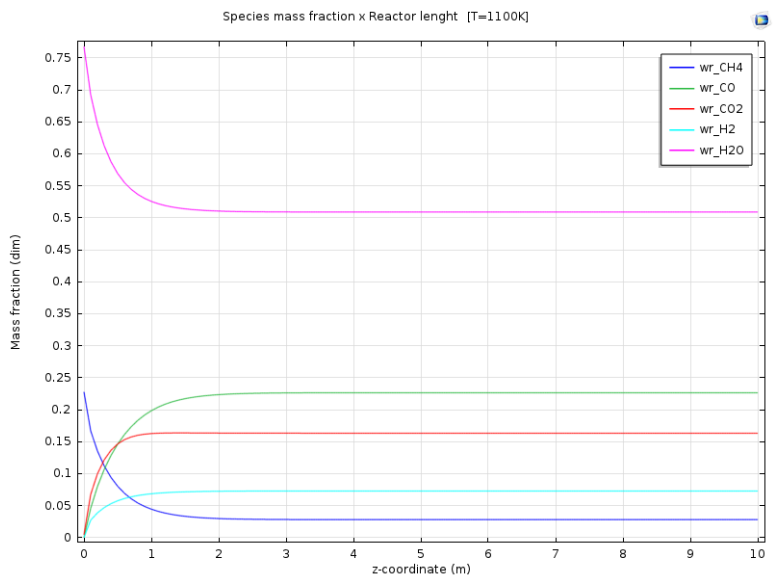

Figure 16 - Species mass fractions as a function of the reactor's length, for T=1100K

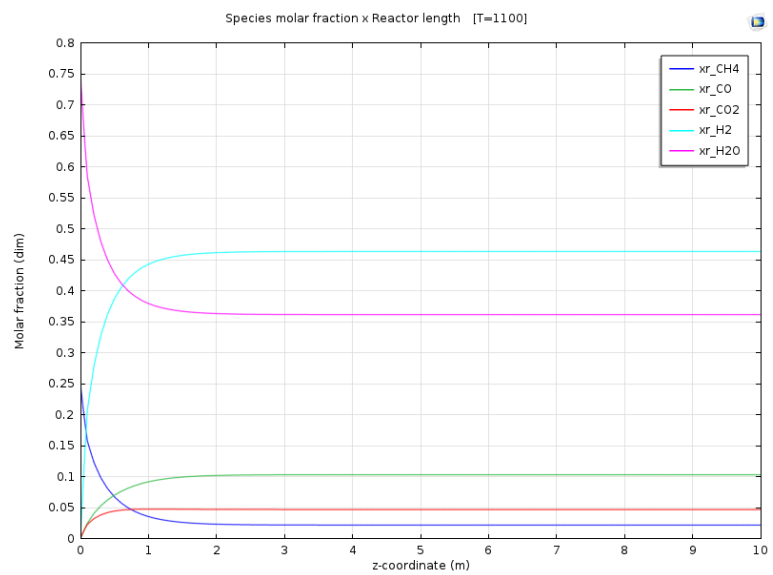

Figure 17 - Species mass fraction as a function of the reactor's length, for T=1100K

The species' mass and molar fractions remains constant from $z = 4m$ to the reactor's outlet, as it can be observed in Fig.(16) and Fig.(17). Thus, for the simulation parameters used, approximately 70% of the packed-bed in the isothermal reactor tube is not used to produce hydrogen.

The effective use of the catalyst pellet along the reactor length is also evaluated in this model. The molar fraction's variation along the reactor (x-z plan) and the associated variation along the pellet radius (y-z plan) are presented in Fig.(18) to Fig.(22), for all species i in the mixture.

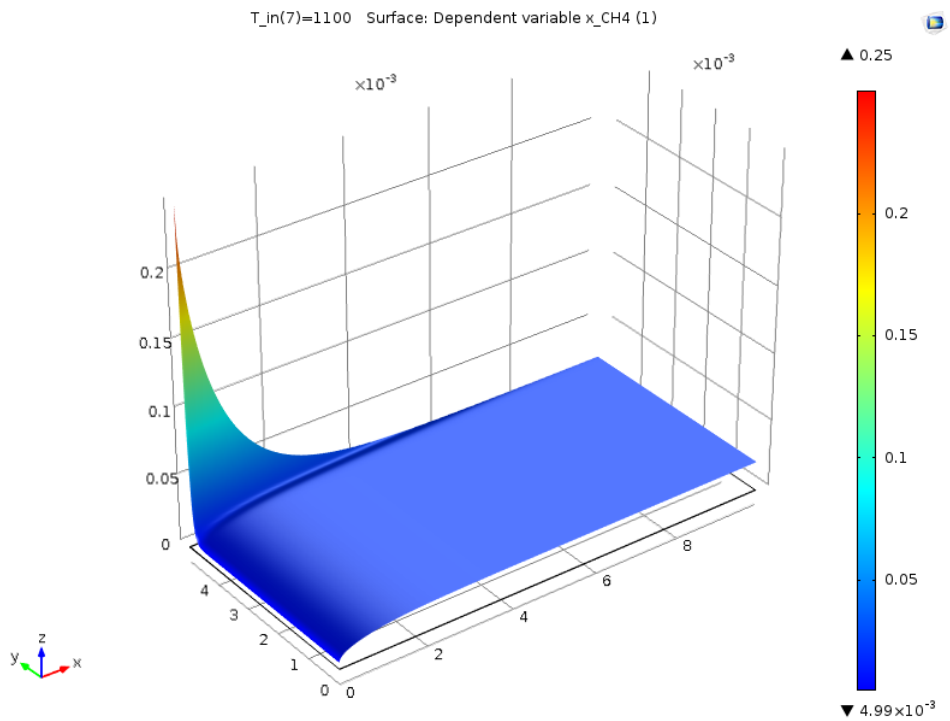

Figure 18 - CH4 molar fractions x reactor length (xz-plan@y=0) and CH4 molar fraction x pellet radius (yz-plan) along the reactor (x-axis)

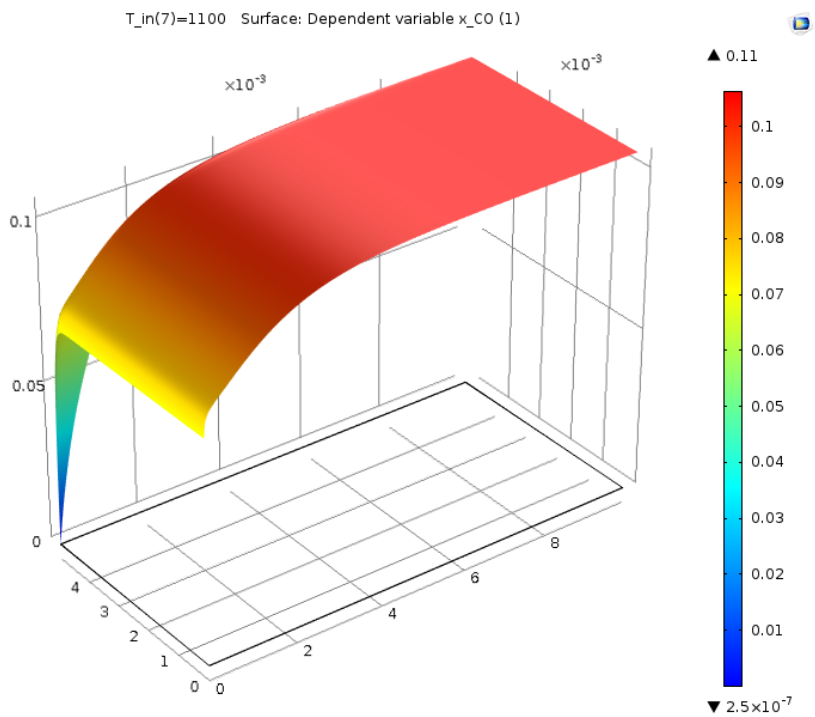

Figure 19 - CO molar fractions x reactor length (xz-plan@y=0) and CO molar fraction x pellet radius (yz-plan) along the reactor (x-axis)

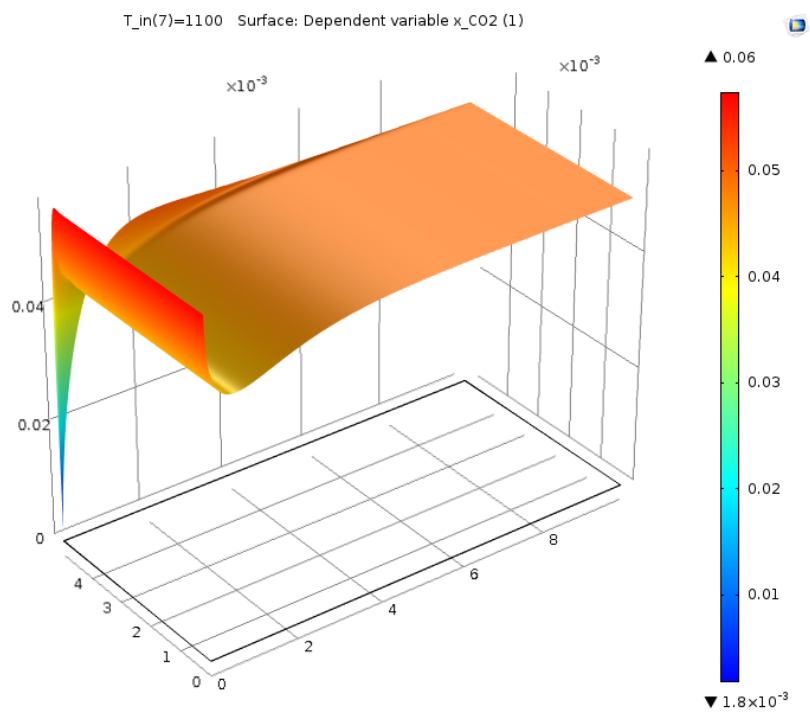

Figure 20 - CO2 molar fractions x reactor length (xz-plan@y=0) and CO2 molar fraction x pellet radius (yz-plan) along the reactor (x-axis)

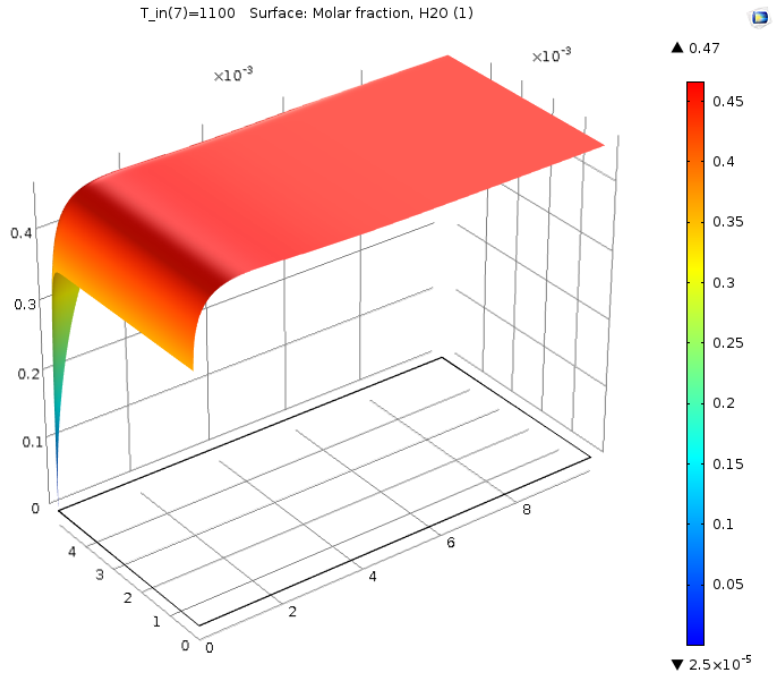

Figure 21 - H2 molar fractions x reactor length (xz-plan@y=0) and H2 molar fraction x pellet radius (yz-plan) along the reactor (x-axis)

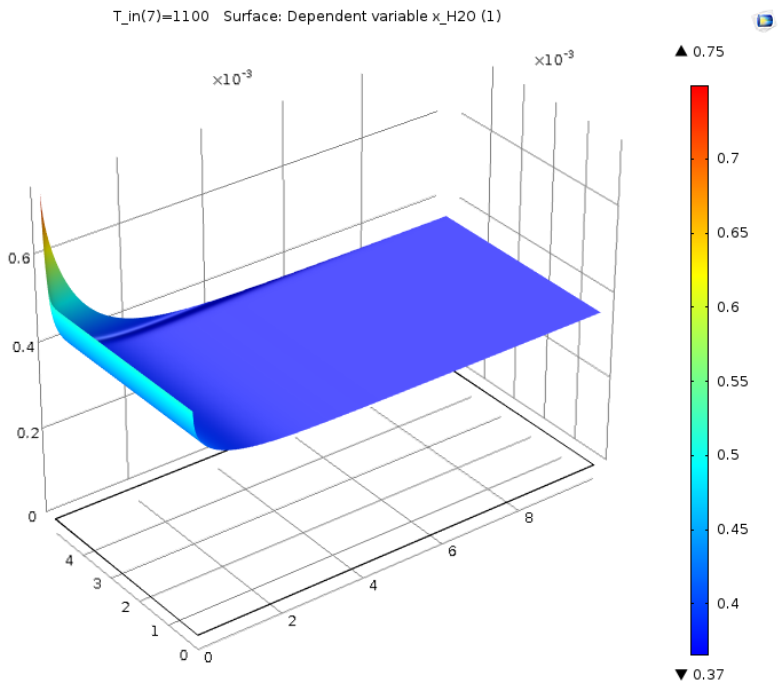

Figure 22 - H2O molar fractions x reactor length (xz-plan@y=0) and H2O molar fraction x pellet radius (yz-plan) along the reactor (x-axis)

The graphics presented in Fig.(18) to Fig.(22) shows that just a small portion of the pellet radius is used throughout the reactor length, and even in the entrance of the reactor where the reactions are far from reach equilibrium. The reason for this behavior is that the reactions in the pellet occurs faster than the diffusive transport of the reactants, i.e., there is no time for the reactants' molecules to reach the interior of the pellet without reacting. This indicates that the size of the pellet can be reduced without interfere in the reaction output, although it can reduce the reactor's void fraction and increase the pressure drop along the reactor.

The effectiveness factor was calculated along the length of the reactor using Eq.(151). The results for $T = 700K$, $T = 900K$ and $T = 1100K$ are presented in Fig.(23) to Fig.(25).

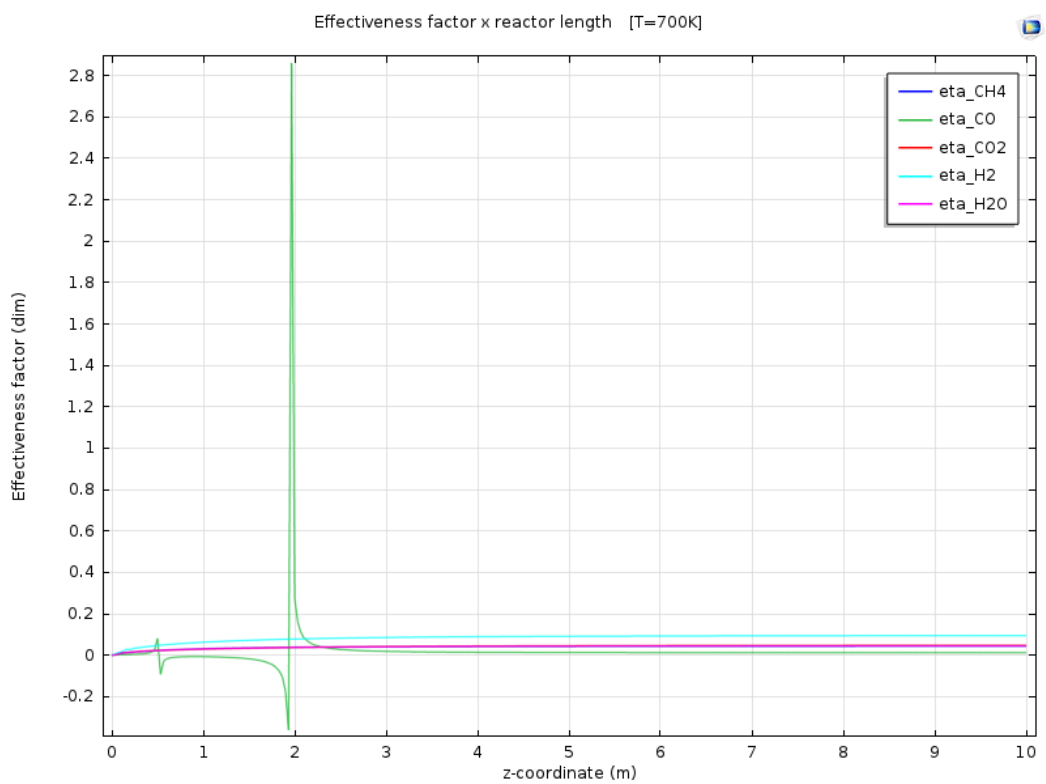

Figure 23 - Effectiveness factor variation along the reactor lenght at T=700K

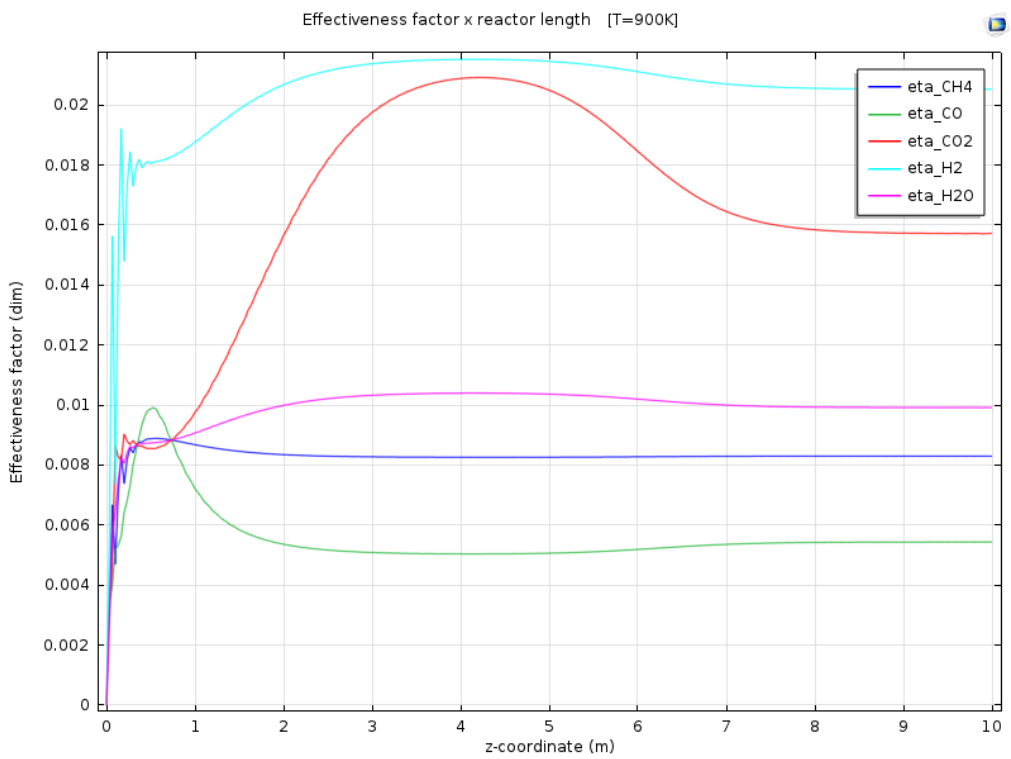

Figure 24 - Effectiveness factor variation along the reactor lenght at T=900K

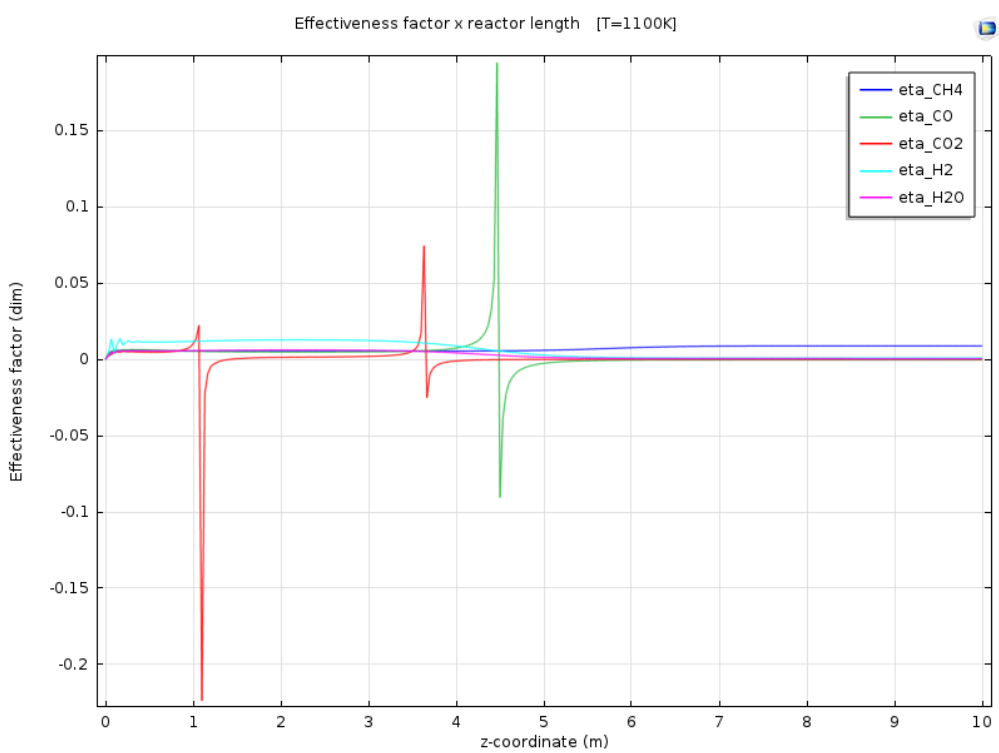

Figure 25 - Effectiveness factor variation along the reactor lenght at T=1100K

The results presented in Fig.(23) to Fig.(25) are consistent to simulations presented in previous works [37], which has observed the effectiveness factor variation along the reactor length. The existence of points where the effectiveness factor is higher than the unity, as shown in Fig.(23), is usually not intuitive since the expected upper limit for the effectiveness factor is equal 1, or 100%.

In fact, this behavior comes from the definition of the effectiveness factor, that considers the actual mass flux of species i leaving the pellet divided by an instantaneous reaction expression, which calculates the reaction production considering all the pellet's reactive surface exposed. In reality, even though the rate of production of i may not exist at some condition, species i generated prior to that time are still traveling inside the pellet towards the surface. In other words, the denominator and the numerator of Eq.(151) have different time responses when the reaction ceases; the denominator acts instantaneously and the numerator has a delay related to the diffusion velocity.

This indicates that the consideration of constant effectiveness factor in simulations, although computationally efficient, are not suitable for the development of local improvements in the reactor's tube. The local efficiency of the reactor can be enhanced by the use of variable effectiveness factor models, consequently increasing the overall efficiency of the system.

6. Conclusions

In this work, a consistent mathematical model was derived for the investigation of porous catalyst pellets and packed-bed reactors. The Reynolds Transport Theorem was applied in the derivation of the conservation laws, i.e., mass conservation, species conservation, momentum conservation and energy conservation, considering their application in problems involving porous catalyst pellets and packed-bed reactors (PBR). The Dusty-Gas Model (DGM) formulation was defined as the diffusion model for the porous catalytic pellet. The general conservation equations developed were then applied in two different models, a spherical catalytic pellet and a packed-bed reactor, considering some assumptions. Production of hydrogen through steam methane reforming was examined in both models at a variety of temperatures. The value of the effectiveness factor along the packed-bed reactor was examined. Further research must be conducted in the investigation of the effectiveness factor to develop models that consider its variation along the reactor length.

7. Appendix

7.1. Appendix A – Model Parameters

Properties used in the pellet and reactor models:

Bed_Parameters_Geometry		
void_fraction	0.7	Void Fraction of the bed reactor
r_tube	50[mm]	Tube's radius
h_tube	10[m]	Tube's height
Ac_tube	pi*r_tube^2	Tube's cross sectional area
As_tube	2*pi*r_tube*h_tube	Tube's side area
V_tube	Ac_tube*h_tube	Tube's total volume
Bed_Parameters_process	0	Bed_Parameters - Process operation
U_ave	0.3[m/s]	Average velocity of the fluid inside the pipe
p_in	28.9[kg]*g_const/1[cm^2]+p_atm	Inlet pressure
p_out	25.5[kg]*g_const/1[cm^2]+p_atm	Outlet pressure
p_atm	1[atm]	Atmosferic pressure
T_in	550[K]	Inlet gas temperature
T_furnace	1100[K]	Furnace temperature
rho_gas_0	c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO	Gas density at the inlet
Concentratons	0	Species inlet/initial concentrations
CO.CH4_0	0.1	Inlet CO-CH4 molar ratio, nCO/nCH4
CO2.CH4_0	0.056	Inlet CO2-CH4 molar ratio, nCO2/nCH4
H2.CH4_0	0.122	Inlet H2-CH4 molar ratio, nH2/nCH4
H2O.CH4_0	3.358	Inlet H2O-CH4 molar ratio, nH2O/nCH4
c_CH4_0	p_in/(R_const*T_in*(1+CO.CH4_0+CO2.CH4_0+H2.CH4_0+H2O.CH4_0))	Inlet concentration, CH4
c_CO_0	CO.CH4_0*c_CH4_0	Inlet concentration, CO
c_CO2_0	CO2.CH4_0*c_CH4_0	Inlet concentration, CO2
c_H2_0	H2.CH4_0*c_CH4_0	Inlet concentration, H2
c_H2O_0	H2O.CH4_0*c_CH4_0	Inlet concentration, H2O
c_TOT_0	c_CH4_0+c_CO_0+c_CO2_0+c_H2_0+c_H2O_0	Total inlet concentration
Molar_fractions	0	Species inlet/initial molar fractions
x_CH4_0	c_CH4_0/c_TOT_0	Molar fraction (initial), CH4

x_CO_0	c_CO_0/c_TOT_0	Molar fraction (initial), CO
x_CO2_0	c_CO2_0/c_TOT_0	Molar fraction (initial), CO2
x_H2_0	c_H2_0/c_TOT_0	Molar fraction (initial), H2
x_H2O_0	c_H2O_0/c_TOT_0	Molar fraction (initial), H2O
Mass_fractions	0	Species inlet/initial mass fractions
w_CH4_0	c_CH4_0*M_CH4/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)	Mass fraction (initial), CH4
w_CO_0	c_CO_0*M_CO/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)	Mass fraction (initial), CO
w_CO2_0	c_CO2_0*M_CO2/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)	Mass fraction (initial), CO2
w_H2_0	c_H2_0*M_H2/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)	Mass fraction (initial), H2
w_H2O_0	c_H2O_0*M_H2O/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)	Mass fraction (initial), H2O
Pellet_parameters	0	Catalyst pellet paramenters
rho_bulk	rho_catalyst*(1-void_fraction)	Catalyst bulk density (weight of catalyst per volume of the bed)
rho_catalyst	2355.2[kg/m^3]	Catalyst density (weight of catalyst per volume of catalyst pellet)
Pellet_parameters_geometry	0	Pellet's parameters - Geometry
r_pellet	5[mm]	Pellet's radius
r_pore	4.86e-5[cm]	Pore radius from pore size distribution
Ax_pellet	4*pi*r_pellet^2	Pellet's external surface area
A_pores	9.3[m^2/g]	Total pore area per weight
Vm_pellet	(4/3)*pi*r_pellet^3	Pellet's monolitic volume
eps_V	rho_catalyst*0.2508[cm^3/g]	Pellet's volumetric porosity
eps_A	eps_V/Tau	Pellet's surface porosity
eps_I	A_pores*rho_catalyst	Area to volume interfacial factor
Tau	3.54	Tortuosity
lambda_solid	98.6[W/(m*K)]	Heat transfer coeff - Conduction (Solid Phase)
Species_parameters	0	Species parameters
M_CH4	16.044[kg/kmol]	Molar Mass, CH4
M_CO	28.001[kg/kmol]	Molar Mass, CO
M_CO2	44.01[kg/kmol]	Molar Mass, CO2
M_H2	2.016[kg/kmol]	Molar Mass, H2
M_H2O	18.02[kg/kmol]	Molar Mass, H2O
Reaction_factors	0	Reaction rate contants

A_k1	4.255*1e15[(kmol*bar)/(kg*h)]	Preexponential factor of rate coefficient, reaction 1
A_k2	1.955*1e6*1[kmol/(bar*kg*h)]	Preexponential factor of rate coefficient, reaction 2
A_k3	1.02*1e15[(kmol*bar)/(kg*h)]	Preexponential factor of rate coefficient, reaction 3
A_CO	8.23*1e-5[1/bar]	Preexponential factor of adsorption contant, CO
A_H2	6.12*1e-9[1/bar]	Preexponential factor of adsorption contant, H2
A_CH4	6.65*1e-4[1/bar]	Preexponential factor of adsorption contant, CH4
A_H2O	1.77*1e5[bar/bar]	Preexponential factor of adsorption contant, H2O
Actv_energy	0	Activation energy of the reactions
E_1	240.1[kJ/mol]	Activation Energy, reaction 1
E_2	67.13[kJ/mol]	Activation Energy, reaction 2
E_3	243.9[kJ/mol]	Activation Energy, reaction 3
Energy_prop	0	Energy balance related properties
dH_CO	-70.65[kJ/mol]	Adsorption enthalpy, CO
dH_H2	-82.9[kJ/mol]	Adsorption enthalpy, H2
dH_CH4	-38.28[kJ/mol]	Adsorption enthalpy, CH4
dH_H2O	88.68[kJ/mol]	Adsorption enthalpy, H2O
dH01	206.1[kJ/mol]	Enthalpy change of reaction I at 298K
dH02	-41.15[kJ/mol]	Enthalpy change of reaction II at 298K
dH03	164.9[kJ/mol]	Enthalpy change of reaction III at 298K
Model_parameters	0	Model Parameters
kk	0.208[W/(m*K)]	Pellet overall conductivity
dh_CH4	-74.81[kJ/mol]	
dh_CO	-110.525[kJ/mol]	
dh_CO2	-393.509[kJ/mol]	
dh_H2	0[kJ/mol]	
dh_H2O	-285.830[kJ/mol]	
k1	A_k1*exp(-E_1/(R_const*T))	Rate coefficient of reaction I
k2	A_k2*exp(-E_2/(R_const*T))	Rate coefficient of reaction II
k3	A_k3*exp(-E_3/(R_const*T))	Rate coefficient of reaction III
K_CO	A_CO*exp(-dH_CO/(R_const*T))	Adsorption constant for CO

K_H2	A_H2*exp(-dH_H2/(R_const*T))	Adsorption constant for H2
K_CH4	A_CH4*exp(-dH_CH4/(R_const*T))	Adsorption constant for CH4
K_H2O	A_H2O*exp(-dH_H2O/(R_const*T))	Dissociative adsorption constant for H2O
K1	exp(30.114-(26830[K]/T))[bar^2]	Equilibrium constant of reaction I
K2	exp(-4.036+(4400[K]/T))	Equilibrium constant of reaction II
K3	K1*K2	Equilibrium constant of reaction III
r1	k1*sqrt(p_pellet*x_H2/1[bar])*((p_pellet*(x_CH4*x_H2O)/x_H2)-((p_pellet^3)*(x_H2^2)*x_CO/K1))/(DEN_star)^2	Rate of reaction I [kmol/(kg_cat*h)]
r2	k2*p_pellet*x_H2*(((p_pellet^2)*x_CO*x_H2O)-((p_pellet^2)*x_H2*x_CO2/K2))/(DEN_star)^2	Rate of reaction II [kmol/(kg_cat*h)]
r3	k3*sqrt(p_pellet*x_H2/1[bar])*((p_pellet*x_CH4*x_H2O/(x_H2^2))-((p_pellet^3)*(x_H2^2)*x_CO2/K3))/(DEN_star)^2	Rate of reaction III [kmol/(kg_cat*h)]
DEN_star	(p_pellet*x_H2)+((p_pellet^2)*K_CO*x_H2*x_CO)+((p_pellet^2)*K_H2*(x_H2^2))+((p_pellet^2)*K_CH4*x_H2*x_CH4)+(p_pellet*K_H2O*x_H2O)	Denominator of reaction equaitons
A_11	#NAME?	
A_12	x_CH4/(M_CO*Deff_CH4.CO)	
A_13	x_CH4/(M_CO2*Deff_CH4.CO2)	
A_14	x_CH4/(M_H2*Deff_CH4.H2)	
A_15	x_CH4/(M_H2O*Deff_CH4.H2O)	
A_21	x_CO/(M_CH4*Deff_CH4.CO)	
A_22	#NAME?	
A_23	x_CO/(M_CO2*Deff_CO.CO2)	
A_24	x_CO/(M_H2*Deff_CO.H2)	
A_25	x_CO/(M_H2O*Deff_CO.H2O)	
A_31	x_CO2/(M_CH4*Deff_CH4.CO2)	
A_32	x_CO2/(M_CO*Deff_CO.CO2)	
A_33	#NAME?	
A_34	x_CO2/(M_H2*Deff_CO2.H2)	
A_35	x_CO2/(M_H2O*Deff_CO2.H2O)	
A_41	x_H2/(M_CH4*Deff_CH4.H2)	
A_42	x_H2/(M_CO*Deff_CO.H2)	
A_43	x_H2/(M_CO2*Deff_CO2.H2)	
A_44	#NAME?	
A_45	x_H2/(M_H2O*Deff_H2.H2O)	
A_51	x_H2O/(M_CH4*Deff_CH4.H2O)	
A_52	x_H2O/(M_CO*Deff_CO.H2O)	
A_53	x_H2O/(M_CO2*Deff_CO2.H2O)	
A_54	x_H2O/(M_H2*Deff_H2.H2O)	

A_55	#NAME?	
D_CH4.CO	D_CH4_CO(T)	Binary diffusivity pair CH4-CO
D_CH4.CO2	D_CH4_CO2(T)	Binary diffusivity pair CH4-CO2
D_CH4.H2	D_CH4_H2(T)	Binary diffusivity pair CH4-H2
D_CH4.H2O	D_CH4_H2O(T)	Binary diffusivity pair CH4-H2O
D_CO.CO2	D_CO_CO2(T)	Binary diffusivity pair CO-CO2
D_CO.H2	D_CO_H2(T)	Binary diffusivity pair CO-H2
D_CO.H2O	D_CO_H2O(T)	Binary diffusivity pair CO-H2O
D_CO2.H2	D_CO2_H2(T)	Binary diffusivity pair CO2-H2
D_CO2.H2O	D_CO2_H2O(T)	Binary diffusivity pair CO2-H2O
D_H2.H2O	D_H2_H2O(T)	Binary diffusivity pair H2-H2O
Knudsen_Diffusivity	0	{{Knudsen diffusivities}}
DK_CH4	((eps_V/Tau)*on3+1*(1-on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CH4))	Knudsen diffusivity CH4
DK_CO	((eps_V/Tau)*on3+1*(1-on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO))	Knudsen diffusivity CO
DK_CO2	((eps_V/Tau)*on3+1*(1-on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO2))	Knudsen diffusivity CO2
DK_H2	((eps_V/Tau)*on3+1*(1-on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2))	Knudsen diffusivity H2
DK_H2O	((eps_V/Tau)*on3+1*(1-on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2O))	Knudsen diffusivity H2O
Effective_diff	0	
Deff_CH4.CO	(eps_V/Tau)*D_CH4.CO	Effective binary diffusivity pair CH4-CO
Deff_CH4.CO2	(eps_V/Tau)*D_CH4.CO2	Effective binary diffusivity pair CH4-CO2
Deff_CH4.H2	(eps_V/Tau)*D_CH4.H2	Effective binary diffusivity pair CH4-H2
Deff_CH4.H2O	(eps_V/Tau)*D_CH4.H2O	Effective binary diffusivity pair CH4-H2O
Deff_CO.CO2	(eps_V/Tau)*D_CO.CO2	Effective binary diffusivity pair CO-CO2
Deff_CO.H2	(eps_V/Tau)*D_CO.H2	Effective binary diffusivity pair CO-H2
Deff_CO.H2O	(eps_V/Tau)*D_CO.H2O	Effective binary diffusivity pair CO-H2O
Deff_CO2.H2	(eps_V/Tau)*D_CO2.H2	Effective binary diffusivity pair CO2-H2

Deff_CO2.H2O	(eps_V/Tau)*D_CO2.H2O	Effective binary diffusivity pair CO2-H2O
Deff_H2.H2O	(eps_V/Tau)*D_H2.H2O	Effective binary diffusivity pair H2-H2O
cp_CH4	14.15[kJ/kmol/K]+75.499e-3*T[kJ/kmol/K^2]-17.991e-6*T^2[kJ/kmol/K^3]	Heat capacity, CH4
S.cp.dT_CH4	R_const*((1.702*T)+(((9.081e-3)/2)*T^2/1[K])-(((2.164e-6)/3)*T^3/1[K^2]))	Integration of cp(T)dt from 298 to T, CH4 - gives the change in enthalpy for the specie from ref. state to T at CONSTANT PRESSURE
S.cp.dT_CO	R_const*((3.376*T)+(((0.557e-3)/2)*T^2/1[K])+(((0.031e5)/3)*T^3/1[K^2]))	Integration of cp(T)dt from 298 to T, CO - gives the change in enthalpy for the specie from ref. state to T at CONSTANT PRESSURE
S.cp.dT_CO2	R_const*((5.457*T)+(((1.045e-3)/2)*T^2/1[K])+(((1.157e5)/3)*T^3/1[K^2]))	Integration of cp(T)dt from 298 to T, CO2 - gives the change in enthalpy for the specie from ref. state to T at CONSTANT PRESSURE
S.cp.dT_H2	R_const*((3.249*T)+(((0.422e-3)/2)*T^2/1[K])-(((0.083e5)/3)*T^3/1[K^2]))	Integration of cp(T)dt from 298 to T, H2 - gives the change in enthalpy for the specie from ref. state to T at CONSTANT PRESSURE
S.cp.dT_H2O	R_const*((3.470*T)+(((1.450e-3)/2)*T^2/1[K])-(((0.121e5)/3)*T^3/1[K^2]))	Integration of cp(T)dt from 298 to T, H2O - gives the change in enthalpy for the specie from ref. state to T at CONSTANT PRESSURE
dH1	3*Dh_H2+Dh_CO-Dh_CH4-Dh_H2O	Total change in enthalpy for reaction 1 (assuming constant pressure)
dH2	Dh_CO2+Dh_H2-Dh_CO-Dh_H2O	Total change in enthalpy for reaction 2 (assuming constant pressure)
dH3	4*Dh_H2+Dh_CO2-Dh_CH4-2*Dh_H2O	Total change in enthalpy for reaction 3 (assuming constant pressure)
Dh_CH4	dh_CH4+(-0.703029*(T/1000)[kJ/(mol*K)]+((108.4773/2)*(T/1000)^2)[kJ/(mol*K^2)]+((-42.52157/3)*(T/1000)^3)[kJ/(mol*K^3)]+((5.862788/4)*(T/1000)^4)[kJ/(mol*K^4)]+(0.678565/(T/1000))[kJ*K/mol])-76.84376[kJ/mol]+74.87310[kJ/mol]	

Dh_CO	dh_CO+(25.56759*(T/1000)[kJ/(mol*K)]+((6.096130/2)*(T/1000)^2)[kJ/(mol*K^2)]+((4.054656/3)*(T/1000)^3)[kJ/(mol*K^3)]-((2.671301/4)*(T/1000)^4)[kJ/(mol*K^4)]-(0.310021/(T/1000))[kJ*K/mol])-118.0089[kJ/mol]+110.5271[kJ/mol]	
Dh_CO2	dh_CO2+(24.99735*(T/1000)[kJ/(mol*K)]+((55.18696/2)*(T/1000)^2)[kJ/(mol*K^2)]-((33.69137/3)*(T/1000)^3)[kJ/(mol*K^3)]+((7.948387/4)*(T/1000)^4)[kJ/(mol*K^4)]+(0.136638/(T/1000))[kJ*K/mol])-403.6075[kJ/mol]+393.5224[kJ/mol]	
Dh_H2	dh_H2+(33.066178*(T/1000)[kJ/(mol*K)]-((11.363417/2)*(T/1000)^2)[kJ/(mol*K^2)]+((11.432816/3)*(T/1000)^3)[kJ/(mol*K^3)]-((2.772874/4)*(T/1000)^4)[kJ/(mol*K^4)]+(0.158558/(T/1000))[kJ*K/mol])-9.980797[kJ/mol]+0[kJ/mol]	
Dh_H2O	dh_H2O+(30.092*(T/1000)[kJ/(mol*K)]+((6.832514/2)*(T/1000)^2)[kJ/(mol*K^2)]+((6.793435/3)*(T/1000)^3)[kJ/(mol*K^3)]-((2.534480/4)*(T/1000)^4)[kJ/(mol*K^4)]-(0.082139/(T/1000))[kJ*K/mol])-250.8810[kJ/mol]+241.8264[kJ/mol]	
c_TOT	c_CH4+c_CO+c_CO2+c_H2+c_H2O	
pelletpressure	0	
r_CH4	0	
r_CO	r1-r2	
r_CO2	r2+r3	
r_H2	3*r1+r2+4*r3	
r_H2O	0	
x_CH4	c_CH4/c_TOT	
x_CO	c_CO/c_TOT	
x_CO2	c_CO2/c_TOT	
x_H2	c_H2/c_TOT	
x_H2O	c_H2O/c_TOT	
Mn	x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O	
w_CH4	x_CH4*M_CH4/Mn	
w_CO	x_CO*M_CO/Mn	
w_CO2	x_CO2*M_CO2/Mn	
w_H2	x_H2*M_H2/Mn	
w_H2O	x_H2O*M_H2O/Mn	
Bo	((2*r_pore)^2)/32	
mu_CH4	2.53*1e-5[Pa*s]	Dinamic viscosity of CH4 @873K
mu_CO	3.86*1e-5[Pa*s]	Dinamic viscosity of CO @873K
mu_CO2	3.74*1e-5[Pa*s]	Dinamic viscosity of CO2 @873K

mu_H2	1.84*1e-5[Pa*s]	Dinamic viscosity of H2 @873K
mu_H2O	3.26*1e-5[Pa*s]	Dinamic viscosity of H2O @873K
mu_gas	(x_CH4*mu_CH4*sqrt(M_CH4/1[kg/kmol])+x_CO*mu_CO*sqrt(M_CO/1[kg/kmol])+x_CO2*mu_CO2*sqrt(M_CO2/1[kg/kmol])+x_H2*mu_H2*sqrt(M_H2/1[kg/kmol])+x_H2O*mu_H2O*sqrt(M_H2O/1[kg/kmol]))/(x_CH4*sqrt(M_CH4/1[kg/kmol])+x_CO*sqrt(M_CO/1[kg/kmol])+x_CO2*sqrt(M_CO2/1[kg/kmol])+x_H2*sqrt(M_H2/1[kg/kmol])+x_H2O*sqrt(M_H2O/1[kg/kmol]))	Mean dinamic viscosity of the mixture @873K
on1	1	
on2	1	
on3	1	

7.2. Appendix B – Simulation codes

Java code: Simulation I – Pellet

```
/*
 * Final_pellet_model_java.java
 */

import com.comsol.model.*;
import com.comsol.model.util.*;

/** Model exported on May 27 2016, 11:13 by COMSOL 5.0.0.244. */
public class Final_pellet_model_java {

  public static Model run() {
    Model model = ModelUtil.create("Model");

    model
         .modelPath("C:\\Users\\Fl\u00e1vio\\OneDrive\\Research\\Reforming Reactor\\Master Thesis\\COMSOL-
Models\\Master Final Model");

    model.modelNode().create("comp1");

    model.geom().create("geom1", 1);

    model.mesh().create("mesh1", "geom1");

    model.physics().create("c", "CoefficientFormPDE", "geom1", new String[][]{{"u"}});

    model.study().create("std1");
    model.study("std1").create("stat", "Stationary");
    model.study("std1").feature("stat").activate("c", true);

    model.baseSystem("cgs");

    model.param()
         .set("Bed_Parameters_Geometry", "0000000000000000000000000000000000000000", "Bed_Parameters -
Geometry");
    model.param().set("void_fraction", "0.7", "Void Fraction of the bed reactor");
    model.param().set("r_tube", "50[mm]", "Tube's radius");
    model.param().set("h_tube", "10[m]", "Tube's height");
    model.param().set("Ac_tube", "pi*r_tube^2", "Tube's cross sectional area");
    model.param().set("As_tube", "2*pi*r_tube*h_tube", "Tube's side area");
    model.param().set("V_tube", "Ac_tube*h_tube", "Tube's total volume");
    model.param()
         .set("Bed_Parameters_process", "0000000000000000000000000000000000000000000000", "Bed_Parameters -
Process operation");
    model.param().set("U_ave", "1.0[m/s]", "Average velocity of the fluid inside the pipe");
    model.param().set("p_in", "28.9[kg]*g_const/1[cm^2]+p_atm", "Inlet pressure");
    model.param().set("p_out", "25.5[kg]*g_const/1[cm^2]+p_atm", "Outlet pressure");
```

```
    model.param().set("p_atm", "1[atm]", "Atmosferic pressure");
    model.param().set("T_in", "550[K]", "Inlet gas temperature");
    model.param().set("T_furnace", "1100[K]", "Furnace temperature");
    model.param()
         .set("rho_gas_0", "c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO", "Gas density at
the inlet");
    model.param()
         .set("Concentratons", "0000000000000000000000000000000000000000000000", "Species inlet/initial
concentrations");
    model.param().set("CO.CH4_0", "0", "Inlet CO-CH4 molar ratio, nCO/nCH4");
    model.param().set("CO2.CH4_0", "0", "Inlet CO2-CH4 molar ratio, nCO2/nCH4");
    model.param().set("H2.CH4_0", "0.0001", "Inlet H2-CH4 molar ratio, nH2/nCH4");
    model.param().set("H2O.CH4_0", "3", "Inlet H2O-CH4 molar ratio, nH2O/nCH4");
    model.param()
         .set("c_CH4_0", "p_in/(R_const*T_in*(1+CO.CH4_0+CO2.CH4_0+H2.CH4_0+H2O.CH4_0))", "Inlet concentration,
CH4");
    model.param().set("c_CO_0", "CO.CH4_0*c_CH4_0", "Inlet concentration, CO");
    model.param().set("c_CO2_0", "CO2.CH4_0*c_CH4_0", "Inlet concentration, CO2");
    model.param().set("c_H2_0", "H2.CH4_0*c_CH4_0", "Inlet concentration, H2");
    model.param().set("c_H2O_0", "H2O.CH4_0*c_CH4_0", "Inlet concentration, H2O");
    model.param().set("c_TOT_0", "c_CH4_0+c_CO_0+c_CO2_0+c_H2_0+c_H2O_0", "Total inlet concentration");
    model.param()
         .set("Molar_fractions", "0000000000000000000000000000000000000000000000", "Species inlet/initial
molar fractions");
    model.param().set("x_CH4_0", "c_CH4_0/c_TOT_0", "Molar fraction (initial), CH4");
    model.param().set("x_CO_0", "c_CO_0/c_TOT_0", "Molar fraction (initial), CO");
    model.param().set("x_CO2_0", "c_CO2_0/c_TOT_0", "Molar fraction (initial), CO2");
    model.param().set("x_H2_0", "c_H2_0/c_TOT_0", "Molar fraction (initial), H2");
    model.param().set("x_H2O_0", "c_H2O_0/c_TOT_0", "Molar fraction (initial), H2O");
    model.param()
         .set("Mass_fractions", "0000000000000000000000000000000000000000000000", "Species inlet/initial
mass fractions");
    model.param()
         .set("w_CH4_0", "c_CH4_0*M_CH4/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), CH4");
    model.param()
         .set("w_CO_0", "c_CO_0*M_CO/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), CO");
    model.param()
         .set("w_CO2_0", "c_CO2_0*M_CO2/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), CO2");
    model.param()
         .set("w_H2_0", "c_H2_0*M_H2/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), H2");
    model.param()
         .set("w_H2O_0", "c_H2O_0*M_H2O/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), H2O");
    model.param().set("Pellet_parameters", "0000000000000000000000000000000000000000", "Catalyst pellet
paramenters");
    model.param()
         .set("rho_bulk", "rho_catalyst*(1-void_fraction)", "Catalyst bulk density (weight of catalyst per
volume of the bed)");
    model.param()
         .set("rho_catalyst", "2355.2[kg/m^3]", "Catalyst density (weight of catalyst per volume of catalyst
pellet)");
    model.param()
         .set("Pellet_parameters_geometry", "0000000000000000000000000000000000000000", "Pellet's parameters -
Geometry");
    model.param().set("r_pellet", "5[mm]", "Pellet's radius");
    model.param().set("Ax_pellet", "4*pi*r_pellet^2", "Pellet's external surface area");
    model.param().set("A_pores", "792679.4825[cm^2/g]", "Total pore area per weight");
    model.param().set("Vm_pellet", "(4/3)*pi*r_pellet^3", "Pellet's \"monolitic\" volume");
    model.param().set("eps_V", "rho_catalyst*0.2508[cm^3/g]", "Pellet's volumetric porosity");
    model.param().set("eps_A", "eps_V/Tau", "Pellet's surface porosity");
    model.param().set("eps_I", "A_pores*rho_catalyst", "Area to volume interfacial factor");
    model.param().set("Tau", "3.54", "Tortuosity");
    model.param().set("lambda_solid", "98.6[W/(m*K)]", "Heat transfer coeff - Conduction (Solid Phase)");
    model.param().set("Species_parameters", "0000000000000000000000000000000000000000000000.", "Species
parameters");
    model.param().set("M_CH4", "16.044[kg/kmol]", "Molar Mass, CH4");
    model.param().set("M_CO", "28.001[kg/kmol]", "Molar Mass, CO");
    model.param().set("M_CO2", "44.01[kg/kmol]", "Molar Mass, CO2");
    model.param().set("M_H2", "2.016[kg/kmol]", "Molar Mass, H2");
    model.param().set("M_H2O", "18.02[kg/kmol]", "Molar Mass, H2O");
    model.param().set("Reaction_factors", "0000000000000000000000000000000000000000", "Reaction rate contants");
    model.param()
         .set("A_k1", "4.255*1e15[(kmol*bar)/(kg*h)]", "Preexponential factor of rate coefficient, reaction
1");
    model.param()
         .set("A_k2", "1.955*1e6*1[kmol/(bar*kg*h)]", "Preexponential factor of rate coefficient, reaction 2");
    model.param()
```

```
        .set("A_k3", "1.02*1e15[(kmol*bar)/(kg*h)]", "Preexponential factor of rate coefficient, reaction 3");
    model.param().set("A_CO", "8.23*1e-5[1/bar]", "Preexponential factor of adsorption contant, CO");
    model.param().set("A_H2", "6.12*1e-9[1/bar]", "Preexponential factor of adsorption contant, H2");
    model.param().set("A_CH4", "6.65*1e-4[1/bar]", "Preexponential factor of adsorption contant, CH4");
    model.param().set("A_H2O", "1.77*1e5[bar/bar]", "Preexponential factor of adsorption contant, H2O");
    model.param()
        .set("Actv_energy", "0000000000000000000000000000000000000000000000000000000000000000", "Activation
energy of the reactions");
    model.param().set("E_1", "240.1[kJ/mol]", "Activation Energy, reaction 1");
    model.param().set("E_2", "67.13[kJ/mol]", "Activation Energy, reaction 2");
    model.param().set("E_3", "243.9[kJ/mol]", "Activation Energy, reaction 3");
    model.param()
        .set("Energy_prop", "0000000000000000000000000000000000000000000000000000000000000000", "Energy
balance related properties");
    model.param().set("dH_CO", "-70.65[kJ/mol]", "Adsorption enthalpy, CO");
    model.param().set("dH_H2", "-82.9[kJ/mol]", "Adsorption enthalpy, H2");
    model.param().set("dH_CH4", "-38.28[kJ/mol]", "Adsorption enthalpy, CH4");
    model.param().set("dH_H2O", "88.68[kJ/mol]", "Adsorption enthalpy, H2O");
    model.param().set("dH01", "206.1[kJ/mol]", "Enthalpy change of reaction I at 298K");
    model.param().set("dH02", "-41.15[kJ/mol]", "Enthalpy change of reaction II at 298K");
    model.param().set("dH03", "164.9[kJ/mol]", "Enthalpy change of reaction III at 298K");
    model.param()
        .set("Model_parameters", "0000000000000000000000000000000000000000000000000000000000000000", "Model
Parameters");
    model.param().set("p_TOT", "p_in", "Total pressure over the domain");
    model.param().set("T", "T_in", "Temperature over the domain");
    model.param().set("rrr", "R_const");
    model.param().remove("T");
    model.param().remove("p_TOT");

    model.variable().create("var1");
    model.variable("var1").model("comp1");

    model.geom("geom1").run();

    model.variable("var1").set("k1", "A_k1*exp(-E_1/(R_const*T))", "Rate coefficient of reaction I");
    model.variable("var1").set("k2", "A_k2*exp(-E_2/(R_const*T))", "Rate coefficient of reaction II");
    model.variable("var1").set("k3", "A_k3*exp(-E_3/(R_const*T))", "Rate coefficient of reaction III");
    model.variable("var1").set("K_CO", "A_CO*exp(-dH_CO/(R_const*T))", "Adsorption constant for CO");
    model.variable("var1").set("K_H2", "A_H2*exp(-dH_H2/(R_const*T))", "Adsorption constant for H2");
    model.variable("var1").set("K_CH4", "A_CH4*exp(-dH_CH4/(R_const*T))", "Adsorption constant for CH4");
    model.variable("var1").set("K_H2O", "A_H2O*exp(-dH_H2O/(R_const*T))", "Dissociative adsorption constant for
H2O");
    model.variable("var1").set("K1", "exp(30.114-(26830[K]/T))[bar^2]", "Equilibrium constant of reaction I");
    model.variable("var1").set("K2", "exp(-4.036+(4400[K]/T))", "Equilibrium constant of reaction II");
    model.variable("var1").set("K3", "K1*K2", "Equilibrium constant of reaction III");
    model.variable("var1")
        .set("r1", "k1*sqrt(p_TOT*x_H2/1[bar])*((p_TOT*(x_CH4*x_H2O)/x_H2)-
((p_TOT^3)*(x_H2^2)*x_CO/K1))/(DEN_star)^2", "Rate of reaction I");
    model.variable("var1")
        .set("r2", "k2*p_TOT*x_H2*(((p_TOT^2)*x_CO*x_H2O)-((p_TOT^2)*x_H2*x_CO2/K2))/(DEN_star)^2", "Rate of
reaction II");
    model.variable("var1")
        .set("r3", "k3*sqrt(p_TOT*x_H2/1[bar])*((p_TOT*x_CH4*(x_H2O^2)/(x_H2^2))-
((p_TOT^3)*(x_H2^2)*x_CO2/K3))/(DEN_star)^2", "Rate of reaction III");
    model.variable("var1")
        .set("DEN_star",
"(p_TOT*x_H2)+((p_TOT^2)*K_CO*x_H2*x_CO)+((p_TOT^2)*K_H2*(x_H2^2))+((p_TOT^2)*K_CH4*x_H2*x_CH4)+(p_TOT*K_H2O*x_
H2O)", "Denominator of reaction equaitons");
    model.variable("var1").set("r1_A", "r1*rho_catalyst/eps_I", "Rate of reaction I, per unit of porous area");
    model.variable("var1").set("r2_A", "r2*rho_catalyst/eps_I", "Rate of reaction II, per unit of porous
area");
    model.variable("var1").set("r3_A", "r3*rho_catalyst/eps_I", "Rate of reaction III, per unit of porous
area");
    model.variable("var1").label("Reaction rates and reaction kinectic constants");
    model.variable().create("var2");
    model.variable("var2").model("comp1");
    model.variable("var2").label("Inverse matrix A (5x5)");
    model.variable("var2").set("A_11", "-
((x_CO/Deff_CH4.CO)+(x_CO2/Deff_CH4.CO2)+(x_H2/Deff_CH4.H2)+(1/DKeff_CH4))");
    model.variable("var2").set("A_12", "x_CH4/Deff_CH4.CO");
    model.variable("var2").set("A_13", "x_CH4/Deff_CH4.CO2");
    model.variable("var2").set("A_14", "x_CH4/Deff_CH4.H2");
    model.variable("var2").set("A_21", "x_CO/Deff_CH4.CO");
    model.variable("var2").set("A_22", "-
((x_CH4/Deff_CH4.CO)+(x_CO2/Deff_CO.CO2)+(x_H2/Deff_CO.H2)+(1/DKeff_CO))");
    model.variable("var2").set("A_23", "x_CO/Deff_CO.CO2");
    model.variable("var2").set("A_24", "x_CO/Deff_CO.H2");
    model.variable("var2").set("A_31", "x_CO2/Deff_CH4.CO2");
    model.variable("var2").set("A_32", "x_CO2/Deff_CO.CO2");
```

```
    model.variable("var2").set("A_33", "-
((x_CH4/Deff_CH4.CO2)+(x_CO/Deff_CO.CO2)+(x_H2/Deff_CO2.H2)+(1/DKeff_CO2))");
    model.variable("var2").set("A_34", "x_CO2/Deff_CO2.H2");
    model.variable("var2").set("A_41", "x_H2/Deff_CH4.H2");
    model.variable("var2").set("A_42", "x_H2/Deff_CO.H2");
    model.variable("var2").set("A_43", "x_H2/Deff_CO2.H2");
    model.variable("var2").set("A_44", "-
((x_CH4/Deff_CH4.H2)+(x_CO/Deff_CO.H2)+(x_CO2/Deff_CO2.H2)+(1/DKeff_H2))");
    model.variable("var2").set("AI_11", "1");
    model.variable("var2").set("B_22", "A_11*A_22-A_21*A_12");
    model.variable("var2").set("B_23", "A_11*A_23-A_21*A_13");
    model.variable("var2").set("B_24", "A_11*A_24-A_21*A_14");
    model.variable("var2").set("B_32", "A_11*A_32-A_31*A_12");
    model.variable("var2").set("B_33", "A_11*A_33-A_31*A_13");
    model.variable("var2").set("B_34", "A_11*A_34-A_31*A_14");
    model.variable("var2").set("B_42", "A_11*A_42-A_41*A_12");
    model.variable("var2").set("B_43", "A_11*A_43-A_41*A_13");
    model.variable("var2").set("B_44", "A_11*A_44-A_41*A_14");
    model.variable("var2").set("BI_21", "A_21*AI_11");
    model.variable("var2").set("BI_31", "A_31*AI_11");
    model.variable("var2").set("BI_41", "A_41*AI_11");
    model.variable("var2").set("C_33", "B_22*B_33-B_32*B_23");
    model.variable("var2").set("C_34", "B_22*B_34-B_32*B_24");
    model.variable("var2").set("C_43", "B_22*B_43-B_42*B_23");
    model.variable("var2").set("C_44", "B_22*B_44-B_42*B_24");
    model.variable("var2").set("CI_31", "B_22*BI_31-B_32*BI_21");
    model.variable("var2").set("CI_32", "-B_32*A_11");
    model.variable("var2").set("CI_33", "B_22*A_11");
    model.variable("var2").set("CI_41", "B_22*BI_41-B_42*BI_21");
    model.variable("var2").set("CI_42", "-B_42*A_11");
    model.variable("var2").set("CI_44", "B_22*A_11");
    model.variable("var2").set("D_44", "C_33*C_44-C_43*C_34");
    model.variable("var2").set("DI_41", "C_33*CI_41-C_43*CI_31");
    model.variable("var2").set("DI_42", "C_33*CI_42-C_43*CI_32");
    model.variable("var2").set("DI_43", "-C_43*CI_33");
    model.variable("var2").set("DI_44", "C_33*CI_44");
    model.variable("var2").set("E_11", "A_11*D_44");
    model.variable("var2").set("E_12", "A_12*D_44");
    model.variable("var2").set("E_13", "A_13*D_44");
    model.variable("var2").set("E_22", "B_22*D_44");
    model.variable("var2").set("E_23", "B_23*D_44");
    model.variable("var2").set("E_33", "C_33*D_44");
    model.variable("var2").set("EI_11", "D_44*AI_11-DI_41*A_14");
    model.variable("var2").set("EI_12", "-DI_42*A_14");
    model.variable("var2").set("EI_13", "-DI_43*A_14");
    model.variable("var2").set("EI_14", "-DI_44*A_14");
    model.variable("var2").set("EI_21", "D_44*BI_21-DI_41*B_24");
    model.variable("var2").set("EI_22", "D_44*A_11-DI_42*B_24");
    model.variable("var2").set("EI_23", "-DI_43*B_24");
    model.variable("var2").set("EI_24", "-DI_44*B_24");
    model.variable("var2").set("EI_31", "D_44*CI_31-DI_41*C_34");
    model.variable("var2").set("EI_32", "D_44*CI_32-DI_42*C_34");
    model.variable("var2").set("EI_33", "D_44*CI_33-DI_43*C_34");
    model.variable("var2").set("EI_34", "-DI_44*C_34");
    model.variable("var2").set("F_11", "E_33*E_11");
    model.variable("var2").set("F_12", "E_33*E_12");
    model.variable("var2").set("F_22", "E_33*E_22");
    model.variable("var2").set("FI_11", "E_33*EI_11-EI_31*E_13");
    model.variable("var2").set("FI_12", "E_33*EI_12-EI_32*E_13");
    model.variable("var2").set("FI_13", "E_33*EI_13-EI_33*E_13");
    model.variable("var2").set("FI_14", "E_33*EI_14-EI_34*E_13");
    model.variable("var2").set("FI_21", "E_33*EI_21-EI_31*E_23");
    model.variable("var2").set("FI_22", "E_33*EI_22-EI_32*E_23");
    model.variable("var2").set("FI_23", "E_33*EI_23-EI_33*E_23");
    model.variable("var2").set("FI_24", "E_33*EI_24-EI_34*E_23");
    model.variable("var2").set("G_11", "F_22*F_11");
    model.variable("var2").set("GI_11", "F_22*FI_11-FI_21*F_12");
    model.variable("var2").set("GI_12", "F_22*FI_12-FI_22*F_12");
    model.variable("var2").set("GI_13", "F_22*FI_13-FI_23*F_12");
    model.variable("var2").set("GI_14", "F_22*FI_14-FI_24*F_12");
    model.variable("var2").set("invA_11", "GI_11/G_11");
    model.variable("var2").set("invA_12", "GI_12/G_11");
    model.variable("var2").set("invA_13", "GI_13/G_11");
    model.variable("var2").set("invA_14", "GI_14/G_11");
    model.variable("var2").set("invA_21", "FI_21/F_22");
    model.variable("var2").set("invA_22", "FI_22/F_22");
    model.variable("var2").set("invA_23", "FI_23/F_22");
    model.variable("var2").set("invA_24", "FI_24/F_22");
    model.variable("var2").set("invA_31", "EI_31/E_33");
    model.variable("var2").set("invA_32", "EI_32/E_33");
    model.variable("var2").set("invA_33", "EI_33/E_33");
```

```
    model.variable("var2").set("invA_34", "EI_34/E_33");
    model.variable("var2").set("invA_41", "DI_41/D_44");
    model.variable("var2").set("invA_42", "DI_42/D_44");
    model.variable("var2").set("invA_43", "DI_43/D_44");
    model.variable("var2").set("invA_44", "DI_44/D_44");
    model.variable("var2")
         .set("A_11", "-
((c_CO/(c_CH4*D_CH4.CO))+(c_CO2/(c_CH4*D_CH4.CO2))+(c_H2/(c_CH4*D_CH4.H2))+(c_H2O/(c_CH4*D_CH4.H2O))+(1/(x_CH4*
DK_CH4)))");
    model.variable("var2").set("A_12", "1/D_CH4.CO");
    model.variable("var2").set("A_13", "1/D_CH4.CO2");
    model.variable("var2").descr("A_11", "(1) CH4");
    model.variable("var2").descr("A_12", "(2)");
    model.variable("var2").descr("A_12", "(2) CO");
    model.variable("var2").descr("A_13", "(3) CO2");
    model.variable("var2").descr("A_14", "(4) H2");
    model.variable("var2").descr("A_21", "(5) H2O");
    model.variable("var2").set("A_14", "1/D_CH4.H2");
    model.variable("var2").set("A_21", "1/D_CH4.CO");
    model.variable("var2").set("A_51", "51");
    model.variable("var2").set("A_52", "52");
    model.variable("var2").set("A_53", "53");
    model.variable("var2").set("A_54", "54");
    model.variable("var2").set("A_55", "55");
    model.variable("var2").rename("A_51", "A_15");
    model.variable("var2").set("A_15", "1/D_CH4.H2O");
    model.variable("var2").descr("A_21", "");
    model.variable("var2").descr("A_15", "(5) H2O");
    model.variable("var2").set("A_21", "12");
    model.variable("var2")
         .set("A_22", "-
((c_CH4/(c_CO*D_CH4.CO))+(c_CO2/(c_CO*D_CO.CO2))+(c_H2/(c_CO*D_CO.H2))+(c_H2O/(c_CO*D_CO.H2O))+(1/(x_CO*DK_CO))
)");
    model.variable("var2").set("A_23", "1/D_CO.CO2");
    model.variable("var2").set("A_24", "1/D_CO.H2");
    model.variable("var2").set("A_31", "1/D_CH4.CO2");
    model.variable("var2").set("A_21", "1/D_CO.H2O");
    model.variable("var2").rename("A_21", "A_25");
    model.variable("var2").set("A_25", "1/D_CO.H2O");
    model.variable("var2").set("A_31", "13");
    model.variable("var2").set("A_32", "23");
    model.variable("var2").descr("A_31", "###");
    model.variable("var2").descr("A_32", "###");
    model.variable("var2")
         .set("A_33", "-
((c_CH4/(c_CO2*D_CH4.CO2))+(c_CO/(c_CO2*D_CO.CO2))+(c_H2/(c_CO2*D_CO2.H2))+(c_H2O/(c_CO2*D_CO2.H2O))+(1/(x_CO2*
DK_CO2)))");
    model.variable("var2").set("A_34", "1/D_CO2.H2");
    model.variable("var2").set("A_41", "1/D_CH4.H2");
    model.variable("var2").rename("A_31", "A_");
    model.variable("var2").set("A_", "13", "###");
    model.variable("var2").rename("A_", "A_35");
    model.variable("var2").set("A_35", "13", "###");
    model.variable("var2").set("A_35", "1/D_CO2.H2O");
    model.variable("var2").descr("A_35", "");
    model.variable("var2").set("A_41", "14");
    model.variable("var2").set("A_42", "24");
    model.variable("var2").set("A_43", "34");
    model.variable("var2").descr("A_41", "###");
    model.variable("var2").descr("A_42", "###");
    model.variable("var2").descr("A_43", "###");
    model.variable("var2")
         .set("A_44", "-
((c_CH4/(c_H2*D_CH4.H2))+(c_CO/(c_H2*D_CO.H2))+(c_CO2/(c_H2*D_CO2.H2))+(c_H2O/(c_H2*D_H2.H2O))+(1/(x_H2*DK_H2))
)");
    model.variable("var2").rename("A_41", "A_45");
    model.variable("var2").set("A_45", "14", "###");
    model.variable("var2").set("A_45", "1/D_H2.H2O");
    model.variable("var2").descr("A_45", "");
    model.variable("var2").remove("A_42");
    model.variable("var2").remove("A_52");
    model.variable("var2").remove("A_53");
    model.variable("var2").remove("A_54");
    model.variable("var2")
         .set("A_55", "-
((c_CH4/(c_H2O*D_CH4.H2O))+(c_CO/(c_H2O*D_CO.H2O))+(c_CO2/(c_H2O*D_CO2.H2O))+(c_H2/(c_H2O*D_H2.H2O))+(1/(x_H2O*
DK_H2O)))");
    model.variable("var2").set("B_22", "A_11*A_22-A_12*A_12");
    model.variable("var2").set("B_23", "A_11*A_23-A_12*A_13");
    model.variable("var2").set("B_24", "A_11*A_24-A_12*A_14");
    model.variable("var2").rename("A_43", "B_25");
```

```
model.variable("var2").set("B_25", "34", "###");
model.variable("var2").descr("B_25", "");
model.variable("var2").set("B_25", "A_11*A_25-A_12*A_15");
model.variable("var2").set("B_32", "A_11*A_32-A_13*A_12");
model.variable("var2").set("B_33", "A_11*A_33-A_13*A_13");
model.variable("var2").set("B_34", "A_11*A_34-A_13*A_14");
model.variable("var2").set("B_32", "A_11*A_23-A_13*A_12");
model.variable("var2").rename("A_32", "B_35");
model.variable("var2").set("B_35", "23", "###");
model.variable("var2").set("B_35", "A_11*A_35-A_13*A_15");
model.variable("var2").descr("B_35", "");
model.variable("var2").set("B_42", "A_11*A_24-A_14*A_12");
model.variable("var2").set("B_43", "A_11*A_34-A_14*A_13");
model.variable("var2").set("B_44", "A_11*A_44-A_14*A_14");
model.variable("var2").descr("B_44", "xx");
model.variable("var2").remove("CI_44");
model.variable("var2").set("DI_44", "C_33*CI_33");
model.variable("var2").rename("E_11", "DI_51");
model.variable("var2").set("DI_51", "A_11*D_44");
model.variable("var2").rename("E_12", "DI_52");
model.variable("var2").set("DI_52", "A_12*D_44");
model.variable("var2").rename("E_13", "DI_53");
model.variable("var2").set("DI_53", "A_13*D_44");
model.variable("var2").set("DI_51", "C_33*CI_51-C_53*CI_31");
model.variable("var2").set("DI_52", "C_33*CI_52-C_53*CI_32");
model.variable("var2").set("DI_53", "-C_53*CI_33");
model.variable("var2").rename("E_22", "D_45");
model.variable("var2").set("D_45", "B_22*D_44");
model.variable("var2").rename("E_23", "D_54");
model.variable("var2").set("D_54", "B_23*D_44");
model.variable("var2").rename("E_33", "D_55");
model.variable("var2").set("D_55", "C_33*D_44");
model.variable("var2").set("D_45", "C_33*C_45-C_43*C_35");
model.variable("var2").set("D_54", "C_33*C_54-C_53*C_34");
model.variable("var2").set("D_55", "C_33*C_55-C_53*C_35");
model.variable("var2").rename("EI_11", "E_55");
model.variable("var2").set("E_55", "D_44*AI_11-DI_41*A_14");
model.variable("var2").rename("EI_12", "EI_51");
model.variable("var2").set("EI_51", "-DI_42*A_14");
model.variable("var2").rename("EI_13", "EI_52");
model.variable("var2").set("EI_52", "-DI_43*A_14");
model.variable("var2").rename("EI_14", "EI_53");
model.variable("var2").set("EI_53", "-DI_44*A_14");
model.variable("var2").rename("EI_21", "EI_54");
model.variable("var2").set("EI_54", "D_44*BI_21-DI_41*B_24");
model.variable("var2").rename("EI_22", "EI_55");
model.variable("var2").set("EI_55", "D_44*A_11-DI_42*B_24");
model.variable("var2").descr("EI_55", "xx");
model.variable("var2").set("E_55", "D_44*D_55-D_54*D_45");
model.variable("var2").set("EI_51", "D_44*DI_51-D_54*DI_41");
model.variable("var2").set("EI_52", "D_44*DI_52-D_54*DI_42");
model.variable("var2").set("EI_53", "D_44*DI_53-D_54*DI_43");
model.variable("var2").set("EI_54", "-D_54*DI_44");
model.variable("var2").set("EI_55", "D_44*DI_44");
model.variable("var2").set("EI_23", "0");
model.variable("var2").set("EI_24", "0");
model.variable("var2").set("EI_31", "0");
model.variable("var2").set("EI_32", "0");
model.variable("var2").set("EI_33", "0");
model.variable("var2").set("EI_34", "0");
model.variable("var2").set("F_11", "E_55*A_11");
model.variable("var2").set("F_12", "E_55*A_12");
model.variable("var2").set("EI_23", "A_13*E_55");
model.variable("var2").rename("EI_23", "F_13");
model.variable("var2").set("F_13", "A_13*E_55");
model.variable("var2").rename("EI_24", "F_14");
model.variable("var2").set("F_14", "A_14*E_55");
model.variable("var2").set("F_22", "E_55*B_22");
model.variable("var2").rename("EI_31", "F_23");
model.variable("var2").set("F_23", "0");
model.variable("var2").rename("EI_32", "F_24");
model.variable("var2").set("F_24", "0");
model.variable("var2").rename("EI_33", "F_33");
model.variable("var2").set("F_33", "0");
model.variable("var2").rename("EI_34", "F_34");
model.variable("var2").set("F_34", "0");
model.variable("var2").set("F_23", "B_23*E_55");
model.variable("var2").set("F_24", "B_24*E_55");
model.variable("var2").set("F_33", "C_33*E_55");
model.variable("var2").set("F_34", "C_34*E_55");
model.variable("var2").set("FI_11", "E_55*AI_11-EI_51*A_15");
```

```
model.variable("var2").set("FI_12", "-EI_52*A_15");
model.variable("var2").set("FI_13", "-EI_53*A_15");
model.variable("var2").set("FI_14", "-EI_54*A_15");
model.variable("var2").set("FI_21", "E_55*BI_21-EI_51*B_25");
model.variable("var2").set("FI_22", "E_55*BI_22-EI_52*B_25");
model.variable("var2").set("FI_23", "-EI_53*B_25");
model.variable("var2").set("FI_24", "-EI_54*B_25");
model.variable("var2").set("G_11", "F_44*F_11");
model.variable("var2").set("GI_11", "F_44*FI_11-F_14*FI_41");
model.variable("var2").set("GI_12", "F_44*FI_12-F_14*FI_42");
model.variable("var2").set("GI_13", "F_44*FI_13-F_14*FI_43");
model.variable("var2").set("GI_14", "F_44*FI_14-F_14*FI_44");
model.variable("var2").descr("B_44", "");
model.variable("var2").set("B_45", "A_11*A_45-A_41*A_15");
model.variable("var2").set("B_52", "A_11*A_52-A_51*A_12");
model.variable("var2").set("B_53", "A_11*A_53-A_51*A_13");
model.variable("var2").set("B_54", "A_11*A_54-A_51*A_14");
model.variable("var2").set("B_55", "A_11*A_55-A_51*A_15");
model.variable("var2").set("BI_51", "-A_51*AI_11");
model.variable("var2").set("C_35", "B_22*B_35-B_32*B_25");
model.variable("var2").set("C_45", "B_22*B_45-B_42*B_25");
model.variable("var2").set("C_53", "B_22*B_53-B_52*B_23");
model.variable("var2").set("C_54", "B_22*B_54-B_52*B_24");
model.variable("var2").set("C_55", "B_22*B_55-B_52*B_25");
model.variable("var2").set("CI_51", "B_22*BI_51-B_52*BI_21");
model.variable("var2").set("CI_52", "-B_52*A_11");
model.variable("var2").set("F_44", "D_44*E_55");
model.variable("var2").set("FI_15", "-EI_55*A_15");
model.variable("var2").set("FI_25", "-EI_55*B_25");
model.variable("var2").set("FI_31", "E_55*CI_31-EI_51*C_35");
model.variable("var2").set("FI_32", "E_55*CI_32-EI_52*C_35");
model.variable("var2").set("FI_33", "E_55*CI_33-EI_53*C_35");
model.variable("var2").set("FI_34", "-EI_54*C_35");
model.variable("var2").set("FI_35", "-EI_55*C_35");
model.variable("var2").set("FI_41", "E_55*DI_41-EI_51*D_45");
model.variable("var2").set("FI_42", "E_55*DI_42-EI_52*D_45");
model.variable("var2").set("FI_43", "E_55*DI_43-EI_53*D_45");
model.variable("var2").set("FI_44", "E_55*DI_44-EI_54*D_45");
model.variable("var2").set("FI_45", "-EI_55*D_45");
model.variable("var2").set("G_12", "F_44*F_12");
model.variable("var2").set("G_13", "F_44*F_13");
model.variable("var2").set("G_22", "F_44*F_22");
model.variable("var2").set("G_23", "F_44*F_23");
model.variable("var2").set("G_33", "F_44*F_33");
model.variable("var2").set("GI_15", "F_44*FI_15-F_14*FI_45");
model.variable("var2").set("GI_21", "F_44*FI_21-F_24*FI_41");
model.variable("var2").set("GI_22", "F_44*FI_22-F_24*FI_42");
model.variable("var2").set("GI_23", "F_44*FI_23-F_24*FI_43");
model.variable("var2").set("GI_24", "F_44*FI_24-F_24*FI_44");
model.variable("var2").set("GI_25", "F_44*FI_25-F_24*FI_45");
model.variable("var2").set("GI_31", "F_44*FI_31-F_34*FI_41");
model.variable("var2").set("GI_32", "F_44*FI_32-F_34*FI_42");
model.variable("var2").set("GI_33", "F_44*FI_33-F_34*FI_43");
model.variable("var2").set("GI_34", "F_44*FI_34-F_34*FI_44");
model.variable("var2").set("GI_35", "F_44*FI_35-F_34*FI_45");
model.variable("var2").set("H_11", "G_33*G_11");
model.variable("var2").set("H_12", "G_33*G_12");
model.variable("var2").set("H_22", "G_33*G_22");
model.variable("var2").set("HI_11", "G_33*GI_11-G_13*GI_31");
model.variable("var2").set("HI_12", "G_33*GI_12-G_13*GI_32");
model.variable("var2").set("HI_13", "G_33*GI_13-G_13*GI_33");
model.variable("var2").set("HI_14", "G_33*GI_14-G_13*GI_34");
model.variable("var2").set("HI_15", "G_33*GI_15-G_13*GI_35");
model.variable("var2").set("HI_21", "G_33*GI_21-G_23*GI_31");
model.variable("var2").set("HI_22", "G_33*GI_22-G_23*GI_32");
model.variable("var2").set("HI_23", "G_33*GI_23-G_23*GI_33");
model.variable("var2").set("HI_24", "G_33*GI_24-G_23*GI_34");
model.variable("var2").set("HI_25", "G_33*GI_25-G_23*GI_35");
model.variable("var2").set("HH_11", "H_22*H_11");
model.variable("var2").set("HHI_11", "H_22*HI_11-H_12*HI_21");
model.variable("var2").set("HHI_12", "H_22*HI_12-H_12*HI_22");
model.variable("var2").set("HHI_13", "H_22*HI_13-H_12*HI_23");
model.variable("var2").set("HHI_14", "H_22*HI_14-H_12*HI_24");
model.variable("var2").set("HHI_15", "H_22*HI_15-H_12*HI_25");

model.label("New Pellet Model 2016.mph");

model.variable("var2").descr("EI_55", "");
model.variable("var2").set("invA_15", "1");
model.variable("var2").set("invA_25", "1");
model.variable("var2").set("invA_35", "1");
```

```
    model.variable("var2").set("invA_45", "1");
    model.variable("var2").set("invA_51", "1");
    model.variable("var2").set("invA_52", "2");
    model.variable("var2").set("invA_53", "3");
    model.variable("var2").set("invA_54", "4");
    model.variable("var2").set("invA_55", "1");
    model.variable("var2").set("invA_11", "HHI_11/HH_11");
    model.variable("var2").set("invA_12", "HHI_12/HH_11");
    model.variable("var2").set("invA_13", "HHI_13/HH_11");
    model.variable("var2").set("invA_14", "HHI_14/HH_11");
    model.variable("var2").set("invA_15", "HHI_15/HH_11");
    model.variable("var2").set("invA_21", "HI_21/H_22");
    model.variable("var2").set("invA_22", "HI_22/H_22");
    model.variable("var2").set("invA_23", "HI_23/H_22");
    model.variable("var2").set("invA_24", "HI_24/H_22");
    model.variable("var2").set("invA_25", "HI_25/H_22");
    model.variable("var2").set("invA_31", "GI_31/G_33");
    model.variable("var2").set("invA_32", "GI_32/G_33");
    model.variable("var2").set("invA_33", "GI_33/G_33");
    model.variable("var2").set("invA_34", "GI_34/G_33");
    model.variable("var2").set("invA_35", "GI_35/G_33");
    model.variable("var2").set("invA_41", "FI_41/F_44");
    model.variable("var2").set("invA_42", "FI_41/F_44");
    model.variable("var2").set("invA_43", "FI_41/F_44");
    model.variable("var2").set("invA_42", "FI_42/F_44");
    model.variable("var2").set("invA_43", "FI_43/F_44");
    model.variable("var2").set("invA_44", "FI_44/F_44");
    model.variable("var2").set("invA_45", "FI_45/F_44");
    model.variable("var2").set("invA_51", "EI_51/E_55");
    model.variable("var2").set("invA_52", "EI_52/E_55");
    model.variable("var2").set("invA_53", "EI_53/E_55");
    model.variable("var2").set("invA_54", "EI_54/E_55");
    model.variable("var2").set("invA_55", "EI_55/E_55");
    model.variable().create("var3");
    model.variable("var3").model("comp1");
    model.variable("var3").label("Diffusivities and Knudsen diffusivities");
    model.variable("var3").set("D_CH4.CO", "D_CH4_CO(T)", "Binary diffusivity pair CH4-CO");
    model.variable("var3").set("D_CH4.CO2", "D_CH4_CO2(T)", "Binary diffusivity pair CH4-CO2");
    model.variable("var3").set("D_CH4.H2", "D_CH4_H2(T)", "Binary diffusivity pair CH4-H2");
    model.variable("var3").set("D_CH4.H2O", "D_CH4_H2O(T)", "Binary diffusivity pair CH4-H2O");
    model.variable("var3").set("D_CO.CO2", "D_CO_CO2(T)", "Binary diffusivity pair CO-CO2");
    model.variable("var3").set("D_CO.H2", "D_CO_H2(T)", "Binary diffusivity pair CO-H2");

    return model;
  }

  public static Model run2(Model model) {
    model.variable("var3").set("D_CO.H2O", "D_CO_H2O(T)", "Binary diffusivity pair CO-H2O");
    model.variable("var3").set("D_CO2.H2", "D_CO2_H2(T)", "Binary diffusivity pair CO2-H2");
    model.variable("var3").set("D_CO2.H2O", "D_CO2_H2O(T)", "Binary diffusivity pair CO2-H2O");
    model.variable("var3").set("D_H2.H2O", "D_H2_H2O(T)", "Binary diffusivity pair H2-H2O");
    model.variable("var3")
          .set("Knudsen_Diffusivity", "0000000000000000000000000000000000000000", "Knudsen diffusivities");
    model.variable("var3").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
    model.variable("var3").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))");
    model.variable("var3").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))");
    model.variable("var3").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))");
    model.variable("var3").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))");
    model.variable("var3").descr("DK_CH4", "Knudsen diffusivity CH4");
    model.variable("var3").descr("DK_CO", "Knudsen diffusivity CO");
    model.variable("var3").descr("DK_CO2", "Knudsen diffusivity CO2");
    model.variable("var3").descr("DK_H2", "Knudsen diffusivity H2");
    model.variable("var3").descr("DK_H2O", "Knudsen diffusivity H2O");

    model.param().set("r_pore", "4.86e-5[cm]");
    model.param().descr("r_pore", "Pore radius from pore size distribution");

    model.variable("var3").set("DK_CH4", "(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CH4))");
    model.variable("var3").set("DK_CO", "(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO))");
    model.variable("var3").set("DK_CO2", "(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO2))");
    model.variable("var3").set("DK_H2", "(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2))");
    model.variable("var3").set("DK_H2O", "(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2O))");
    model.variable("var3").descr("Knudsen_Diffusivity", "{{Knudsen diffusivities}}");

    model.func().create("int1", "Interpolation");
    model.func("int1").model("comp1");
    model.func("int1")
          .set("table", new String[][]{{"300", "7.94241E-03"}, {"400", "1.32151E-02"}, {"500", "1.94561E-02"},
{"600", "2.65884E-02"}, {"700", "3.45523E-02"}, {"800", "4.32979E-02"}, {"900", "5.27835E-02"}, {"1000",
"6.29753E-02"}, {"1100", "7.38460E-02"}, {"1200", "8.53726E-02"},
          {"1300", "9.75361E-02"}});
```

```
    model.func("int1").label("Diffusivity CH4-CO");
    model.func("int1").set("funcname", "D_CH4_CO");
    model.func().create("int2", "Interpolation");
    model.func("int2").model("comp1");
    model.func("int2").label("Diffusivity CH4-CO2");
    model.func("int2").set("funcname", "D_CH4_CO2");
    model.func("int2")
         .set("table", new String[][]{{"300", "6.10147E-03"}, {"400", "1.03713E-02"}, {"500", "1.54683E-02"},
{"600", "2.13075E-02"}, {"700", "2.78367E-02"}, {"800", "3.50184E-02"}, {"900", "4.28212E-02"}, {"1000",
"5.12175E-02"}, {"1100", "6.01823E-02"}, {"1200", "6.96940E-02"},
         {"1300", "7.97337E-02"}});
    model.func().create("int3", "Interpolation");
    model.func("int3").model("comp1");
    model.func("int3").label("Diffusivity CH4-H2");
    model.func("int3").set("funcname", "D_CH4_H2");
    model.func("int3")
         .set("table", new String[][]{{"300", "2.57271E-02"}, {"400", "4.24235E-02"}, {"500", "6.21612E-02"},
{"600", "8.46912E-02"}, {"700", "1.09814E-01"}, {"800", "1.37373E-01"}, {"900", "1.67247E-01"}, {"1000",
"1.99340E-01"}, {"1100", "2.33575E-01"}, {"1200", "2.69887E-01"},
         {"1300", "3.08217E-01"}});
    model.func().create("int4", "Interpolation");
    model.func("int4").model("comp1");
    model.func("int4").label("Diffusivity CH4-H2O");
    model.func("int4").set("funcname", "D_CH4_H2O");
    model.func("int4")
         .set("table", new String[][]{{"300", "7.55730E-03"}, {"400", "1.33939E-02"}, {"500", "2.06594E-02"},
{"600", "2.92083E-02"}, {"700", "3.89218E-02"}, {"800", "4.97034E-02"}, {"900", "6.14777E-02"}, {"1000",
"7.41867E-02"}, {"1100", "8.77864E-02"}, {"1200", "1.02242E-01"},
         {"1300", "1.17527E-01"}});
    model.func().create("int5", "Interpolation");
    model.func("int5").model("comp1");
    model.func().create("int6", "Interpolation");
    model.func("int6").model("comp1");
    model.func().create("int7", "Interpolation");
    model.func("int7").model("comp1");
    model.func().create("int8", "Interpolation");
    model.func("int8").model("comp1");
    model.func().create("int9", "Interpolation");
    model.func("int9").model("comp1");
    model.func().create("int10", "Interpolation");
    model.func("int10").model("comp1");
    model.func("int5").label("Diffusivity CO-CO2");
    model.func("int5").set("funcname", "D_CO_CO2");
    model.func("int5")
         .set("table", new String[][]{{"300", "5.59985E-03"}, {"400", "9.38342E-03"}, {"500", "1.38693E-02"},
{"600", "1.89985E-02"}, {"700", "2.47294E-02"}, {"800", "3.10275E-02"}, {"900", "3.78630E-02"}, {"1000",
"4.52106E-02"}, {"1100", "5.30492E-02"}, {"1200", "6.13611E-02"},
         {"1300", "7.01315E-02"}});
    model.func("int6").label("Diffusivity CO-H2");
    model.func("int6").set("funcname", "D_CO_H2");
    model.func("int6")
         .set("table", new String[][]{{"300", "2.71700E-02"}, {"400", "4.44794E-02"}, {"500", "6.49076E-02"},
{"600", "8.81834E-02"}, {"700", "1.14109E-01"}, {"800", "1.42539E-01"}, {"900", "1.73362E-01"}, {"1000",
"2.06490E-01"}, {"1100", "2.41846E-01"}, {"1200", "2.79362E-01"},
         {"1300", "3.18979E-01"}});
    model.func("int7").label("Diffusivity CO-H2O");
    model.func("int7").set("funcname", "D_CO_H2O");
    model.func("int7")
         .set("table", new String[][]{{"300", "7.81705E-03"}, {"400", "1.37584E-02"}, {"500", "2.10820E-02"},
{"600", "2.96364E-02"}, {"700", "3.93029E-02"}, {"800", "4.99923E-02"}, {"900", "6.16392E-02"}, {"1000",
"7.41957E-02"}, {"1100", "8.76251E-02"}, {"1200", "1.01897E-01"},
         {"1300", "1.16987E-01"}});
    model.func("int8").label("Diffusivity CO2-H2");
    model.func("int8").set("funcname", "D_CO2_H2");
    model.func("int8")
         .set("table", new String[][]{{"300", "2.26297E-02"}, {"400", "3.75182E-02"}, {"500", "5.51309E-02"},
{"600", "7.52520E-02"}, {"700", "9.77087E-02"}, {"800", "1.22358E-01"}, {"900", "1.49086E-01"}, {"1000",
"1.77801E-01"}, {"1100", "2.08427E-01"}, {"1200", "2.40904E-01"},
         {"1300", "2.75178E-01"}});
    model.func("int9").label("Diffusivity CO2-H2O");
    model.func("int9").set("funcname", "D_CO2_H2O");
    model.func("int9")
         .set("table", new String[][]{{"300", "5.42252E-03"}, {"400", "9.66518E-03"}, {"500", "1.49969E-02"},
{"600", "2.13157E-02"}, {"700", "2.85357E-02"}, {"800", "3.65849E-02"}, {"900", "4.54033E-02"}, {"1000",
"5.49427E-02"}, {"1100", "6.51647E-02"}, {"1200", "7.60389E-02"},
         {"1300", "8.75414E-02"}});
    model.func("int10").label("Diffusivity H2-H2O");
    model.func("int10").set("funcname", "D_H2_H2O");
    model.func("int10")
```

```
        .set("table", new String[][]{{"300", "2.77926E-02"}, {"400", "4.80037E-02"}, {"500", "7.24387E-02"},
{"600", "1.00604E-01"}, {"700", "1.32171E-01"}, {"800", "1.66924E-01"}, {"900", "2.04704E-01"}, {"1000",
"2.45387E-01"}, {"1100", "2.88863E-01"}, {"1200", "3.35033E-01"},
        {"1300", "3.83805E-01"}});
    model.func("int10").set("argunit", "K");
    model.func("int10").set("fununit", "cm^2/s");

    model.variable().create("var4");
    model.variable("var4").model("comp1");
    model.variable("var4").label("Enthalpy of reactions");
    model.variable("var4")
        .set("cp_CH4", "14.15[kJ/kmol/K]+75.499e-3*T[kJ/kmol/K^2]-17.991e-6*T^2[kJ/kmol/K^3]", "Heat capacity,
CH4");
    model.variable("var4")
        .set("S.cp.dT_CH4", "R_const*((1.702*T)+(((9.081e-3)/2)*T^2/1[K])-(((2.164e-6)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, CH4 - gives the change in enthalpy for the specie from ref. state to T
at CONSTANT PRESSURE");
    model.variable("var4")
        .set("S.cp.dT_CO", "R_const*((3.376*T)+(((0.557e-3)/2)*T^2/1[K])+(((0.031e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, CO - gives the change in enthalpy for the specie from ref. state to T at
CONSTANT PRESSURE");
    model.variable("var4")
        .set("S.cp.dT_CO2", "R_const*((5.457*T)+(((1.045e-3)/2)*T^2/1[K])+(((1.157e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, CO2 - gives the change in enthalpy for the specie from ref. state to T
at CONSTANT PRESSURE");
    model.variable("var4")
        .set("S.cp.dT_H2", "R_const*((3.249*T)+(((0.422e-3)/2)*T^2/1[K])-(((0.083e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, H2 - gives the change in enthalpy for the specie from ref. state to T at
CONSTANT PRESSURE");
    model.variable("var4")
        .set("S.cp.dT_H2O", "R_const*((3.470*T)+(((1.450e-3)/2)*T^2/1[K])-(((0.121e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, H2O - gives the change in enthalpy for the specie from ref. state to T
at CONSTANT PRESSURE");
    model.variable("var4")
        .set("dH1", "dH01+3*S.cp.dT_H2+S.cp.dT_CO-S.cp.dT_CH4-S.cp.dT_H2O", "Total change in enthalpy for
reaction 1 (assuming constant pressure)");
    model.variable("var4")
        .set("dH2", "dH02+S.cp.dT_CO2+S.cp.dT_H2-S.cp.dT_CO-S.cp.dT_H2O", "Total change in enthalpy for
reaction 2 (assuming constant pressure)");
    model.variable("var4")
        .set("dH3", "dH03+4*S.cp.dT_H2+S.cp.dT_CO2-S.cp.dT_CH4-2*S.cp.dT_H2O", "Total change in enthalpy for
reaction 3 (assuming constant pressure)");

    model.geom("geom1").create("i1", "Interval");
    model.geom("geom1").feature("i1").set("p2", "r_pellet");
    model.geom("geom1").runPre("fin");
    model.geom("geom1").run();

    model.variable("var1")
        .set("r1", "k1*sqrt(c_H2*R_const*T/1[bar])*(((c_CH4*c_H2O)/c_H2)*(R_const*T)-
(((R_const*T)^3)*(c_H2^2)*c_CO/K1))/(DEN_star)^2");
    model.variable("var1")
        .set("r2", "k2*c_H2*R_const*T*((((R_const*T)^2)*c_CO*c_H2O)-
(((R_const*T)^2)*c_H2*c_CO2/K2))/(DEN_star)^2");
    model.variable("var1")
        .set("r3", "k3*sqrt(c_H2*R_const*T/1[bar])*(((R_const*T)*c_CH4*(c_H2O^2)/(c_H2^2))-
(((R_const*T)^3)*(c_H2^2)*c_CO2/K3))/(DEN_star)^2");
    model.variable("var1")
        .set("DEN_star",
"(c_H2*R_const*T)+(((R_const*T)^2)*K_CO*c_H2*c_CO)+(((R_const*T)^2)*K_H2*(c_H2^2))+(((R_const*T)^2)*K_CH4*c_H2*
c_CH4)+(R_const*T*K_H2O*c_H2O)");
    model.variable("var1").descr("r1", "Rate of reaction I [kmol/(kg_cat*h)]");
    model.variable("var1").descr("r2", "Rate of reaction II [kmol/(kg_cat*h)]");
    model.variable("var1").descr("r3", "Rate of reaction III [kmol/(kg_cat*h)]");

    model.label("New Pellet Model 2016.mph");

    model.physics("c").field("dimensionless").component(new String[]{"u", "u2", "u3", "u4", "u5"});
    model.physics("c").field("dimensionless").field("c");
    model.physics("c").field("dimensionless").field("c_");
    model.physics("c").field("dimensionless").component(1, "c_CH4");
    model.physics("c").field("dimensionless").component(2, "c_CO");
    model.physics("c").field("dimensionless").component(3, "c_CO2");
    model.physics("c").field("dimensionless").component(4, "c_H2");
    model.physics("c").field("dimensionless").component(5, "c_H2O");
    model.physics("c").prop("Units").set("SourceTermQuantity", "velocity");
    model.physics("c").field("dimensionless").field("u");
    model.physics("c").field("dimensionless").field("c");
    model.physics("c").feature().create("cons1", "Constraint", 0);
    model.physics("c").feature().create("flux1", "FluxBoundary", 0);
    model.physics("c").feature().remove("cons1");
```

```
    model.physics("c").feature().remove("flux1");
    model.physics("c").feature().create("dir1", "DirichletBoundary", 0);
    model.physics().create("g", "GeneralFormPDE", "geom1", new String[][]{{"u"}});

    model.study("std1").feature("stat").activate("g", true);

    model.physics("g").feature("gfeq1").setIndex("da", "0", 0);
    model.physics("g").feature("gfeq1").setIndex("f", "0", 0);
    model.physics("c").field("dimensionless")
         .component(new String[]{"c_CH4", "c_CO", "c_CO2", "c_H2", "c_H2O", "c6"});
    model.physics("c").field("dimensionless").component(6, "T");
    model.physics("g").field("dimensionless").component(new String[]{"u", "u2", "u3", "u4", "u5"});
    model.physics("g").field("dimensionless").component(1, "u1");
    model.physics("g").feature("gfeq1").setIndex("Ga", "-u1x", 0);

    model.label("New Pellet Model 2016.mph");
    model.label("New Pellet Model 2016.mph");

    model.variable("var2")
         .set("A_11", "-
(1/M_CH4)*((x_CO/D_CH4.CO)+(c_CO2/(c_CH4*D_CH4.CO2))+(c_H2/(c_CH4*D_CH4.H2))+(c_H2O/(c_CH4*D_CH4.H2O))+(1/(x_CH
4*DK_CH4)))");

    model.param().remove("rrr");

    model.variable("var3").set("Effective_diff", "0000000000");
    model.variable("var3").set("Deff_CH4.CO", "(eps_V/Tau)*D_CH4.CO");
    model.variable("var3").descr("Deff_CH4.CO", "Effective binary diffusivity pair CH4-CO2");
    model.variable("var3").set("Deff_CH4.CO2", "(eps_V/Tau)*D_CH4.CO2");
    model.variable("var3").set("Deff_CH4.H2", "(eps_V/Tau)*D_CH4.H2");
    model.variable("var3").set("Deff_CH4.H2O", "(eps_V/Tau)*D_CH4.H2O");
    model.variable("var3").set("Deff_CO.CO2", "(eps_V/Tau)*D_CO.CO2");
    model.variable("var3").set("Deff_CO.H2", "(eps_V/Tau)*D_CO.H2");
    model.variable("var3").set("Deff_CO.H2O", "(eps_V/Tau)*D_CO.H2O");
    model.variable("var3").set("Deff_CO2.H2", "(eps_V/Tau)*D_CO2.H2");
    model.variable("var3").set("Deff_CO2.H2O", "(eps_V/Tau)*D_CO2.H2O");
    model.variable("var3").set("Deff_H2.H2O", "(eps_V/Tau)*D_H2.H2O");
    model.variable("var3").descr("Deff_CH4.CO2", "Effective binary diffusivity pair CH4-CO2");
    model.variable("var3").descr("Deff_CH4.CO", "Effective binary diffusivity pair CH4-CO");
    model.variable("var3").descr("Deff_CH4.H2", "Effective binary diffusivity pair CH4-H2");
    model.variable("var3").descr("Deff_CH4.H2O", "Effective binary diffusivity pair CH4-H2O");
    model.variable("var3").descr("Deff_CO.CO2", "Effective binary diffusivity pair CO-CO2");
    model.variable("var3").descr("Deff_CO.H2", "Effective binary diffusivity pair CO-H2");
    model.variable("var3").descr("Deff_CO.H2O", "Effective binary diffusivity pair CO-H2O");
    model.variable("var3").descr("Deff_CO2.H2", "Effective binary diffusivity pair CO2-H2");
    model.variable("var3").descr("Deff_CO2.H2O", "Effective binary diffusivity pair CO2-H2O");
    model.variable("var3").descr("Deff_H2.H2O", "Effective binary diffusivity pair H2-H2O");

    model.physics("c").field("dimensionless").component(new String[]{"c_CH4", "c_CO", "c_CO2", "c_H2",
"c_H2O"});
    model.physics("g").field("dimensionless").component(new String[]{"u1"});
    model.physics("g").field("dimensionless").field("T");
    model.physics("g").field("dimensionless").component(1, "T");
    model.physics("g").prop("Units").set("DependentVariableQuantity", "temperature");
    model.physics("g").prop("Units").set("CustomSourceTermUnit", "(g*cm^2)/(s^3*cm)");

    model.func("int9").set("fununit", "cm^2/s");
    model.func("int9").set("argunit", "K");
    model.func("int8").set("fununit", "cm^2/s");
    model.func("int8").set("argunit", "K");
    model.func("int7").set("fununit", "cm^2/s");
    model.func("int7").set("argunit", "K");
    model.func("int6").set("fununit", "cm^2/s");
    model.func("int6").set("argunit", "K");
    model.func("int5").set("fununit", "cm^2/s");
    model.func("int5").set("argunit", "K");
    model.func("int4").set("fununit", "cm^2/s");
    model.func("int4").set("argunit", "K");
    model.func("int3").set("fununit", "cm^2/s");
    model.func("int3").set("argunit", "K");
    model.func("int2").set("fununit", "cm^2/s");
    model.func("int2").set("argunit", "K");
    model.func("int1").set("fununit", "cm^2/s");
    model.func("int1").set("argunit", "K");

    model.physics("c").field("dimensionless").field("X");
    model.physics("c").field("dimensionless").component(1, "x_CH4");
    model.physics("c").field("dimensionless").component(2, "x_CO");
    model.physics("c").field("dimensionless").component(3, "x_CO2");
    model.physics("c").field("dimensionless").component(4, "x_H2");
    model.physics("c").field("dimensionless").component(5, "x_H2O");
```

```
    model.variable("var2")
          .set("A_11", "-
(1/M_CH4)*((x_CO/D_CH4.CO)+(x_CO2/D_CH4.CO2)+(x_H2/D_CH4.H2)+(x_H2O/D_CH4.H2O)+(1/DK_CH4))");
    model.variable("var2").set("A_12", "x_CH4/(M_CO*D_CH4.CO)");
    model.variable("var2").set("A_13", "x_CH4/(M_CO2*D_CH4.CO2)");
    model.variable("var2").set("A_14", "x_CH4/(M_H2*D_CH4.H2)");
    model.variable("var2").set("A_15", "x_CH4/(M_H2O*D_CH4.H2O)");
    model.variable("var2").remove("invA_55");
    model.variable("var2").remove("A_11");
    model.variable("var2").remove("A_12");
    model.variable("var2").remove("A_13");
    model.variable("var2").remove("A_14");
    model.variable("var2").remove("A_15");
    model.variable("var2").remove("A_22");
    model.variable("var2").remove("A_23");
    model.variable("var2").remove("A_24");
    model.variable("var2").remove("A_25");
    model.variable("var2").remove("A_33");
    model.variable("var2").remove("A_34");
    model.variable("var2").remove("A_35");
    model.variable("var2").remove("A_44");
    model.variable("var2").remove("A_45");
    model.variable("var2").remove("A_55");
    model.variable("var2").remove("AI_11");
    model.variable("var2").remove("B_22");
    model.variable("var2").remove("B_23");
    model.variable("var2").remove("B_24");
    model.variable("var2").remove("B_25");
    model.variable("var2").remove("B_32");
    model.variable("var2").remove("B_33");
    model.variable("var2").remove("B_34");
    model.variable("var2").remove("B_35");
    model.variable("var2").remove("B_42");
    model.variable("var2").remove("B_43");
    model.variable("var2").remove("B_44");
    model.variable("var2").remove("B_45");
    model.variable("var2").remove("B_52");
    model.variable("var2").remove("B_53");
    model.variable("var2").remove("B_54");
    model.variable("var2").remove("B_55");
    model.variable("var2").remove("BI_21");
    model.variable("var2").remove("BI_31");
    model.variable("var2").remove("BI_41");
    model.variable("var2").remove("BI_51");
    model.variable("var2").remove("C_33");
    model.variable("var2").remove("C_34");
    model.variable("var2").remove("C_35");
    model.variable("var2").remove("C_43");
    model.variable("var2").remove("C_44");
    model.variable("var2").remove("C_45");
    model.variable("var2").remove("C_53");
    model.variable("var2").remove("C_54");
    model.variable("var2").remove("C_55");
    model.variable("var2").remove("CI_31");
    model.variable("var2").remove("CI_32");
    model.variable("var2").remove("CI_33");
    model.variable("var2").remove("CI_41");
    model.variable("var2").remove("CI_42");
    model.variable("var2").remove("CI_51");
    model.variable("var2").remove("CI_52");
    model.variable("var2").remove("D_44");
    model.variable("var2").remove("D_45");
    model.variable("var2").remove("D_54");
    model.variable("var2").remove("D_55");
    model.variable("var2").remove("DI_41");
    model.variable("var2").remove("DI_42");
    model.variable("var2").remove("DI_43");
    model.variable("var2").remove("DI_44");
    model.variable("var2").remove("DI_51");
    model.variable("var2").remove("DI_52");
    model.variable("var2").remove("DI_53");
    model.variable("var2").remove("E_55");
    model.variable("var2").remove("EI_51");
    model.variable("var2").remove("EI_52");
    model.variable("var2").remove("EI_53");
    model.variable("var2").remove("EI_54");
    model.variable("var2").remove("EI_55");
    model.variable("var2").remove("F_11");
    model.variable("var2").remove("F_12");
    model.variable("var2").remove("F_13");
```

```
model.variable("var2").remove("F_14");
model.variable("var2").remove("F_22");
model.variable("var2").remove("F_23");
model.variable("var2").remove("F_24");
model.variable("var2").remove("F_33");
model.variable("var2").remove("F_34");
model.variable("var2").remove("F_44");
model.variable("var2").remove("FI_11");
model.variable("var2").remove("FI_12");
model.variable("var2").remove("FI_13");
model.variable("var2").remove("FI_14");
model.variable("var2").remove("FI_15");
model.variable("var2").remove("FI_21");
model.variable("var2").remove("FI_22");
model.variable("var2").remove("FI_23");
model.variable("var2").remove("FI_24");
model.variable("var2").remove("FI_25");
model.variable("var2").remove("FI_31");
model.variable("var2").remove("FI_32");
model.variable("var2").remove("FI_33");
model.variable("var2").remove("FI_34");
model.variable("var2").remove("FI_35");
model.variable("var2").remove("FI_41");
model.variable("var2").remove("FI_42");
model.variable("var2").remove("FI_43");
model.variable("var2").remove("FI_44");
model.variable("var2").remove("FI_45");
model.variable("var2").remove("G_11");
model.variable("var2").remove("G_12");
model.variable("var2").remove("G_13");
model.variable("var2").remove("G_22");
model.variable("var2").remove("G_23");
model.variable("var2").remove("G_33");
model.variable("var2").remove("GI_11");
model.variable("var2").remove("GI_12");
model.variable("var2").remove("GI_13");
model.variable("var2").remove("GI_14");
model.variable("var2").remove("GI_15");
model.variable("var2").remove("GI_21");
model.variable("var2").remove("GI_22");
model.variable("var2").remove("GI_23");
model.variable("var2").remove("GI_24");
model.variable("var2").remove("GI_25");
model.variable("var2").remove("GI_31");
model.variable("var2").remove("GI_32");
model.variable("var2").remove("GI_33");
model.variable("var2").remove("GI_34");
model.variable("var2").remove("GI_35");
model.variable("var2").remove("H_11");
model.variable("var2").remove("H_12");
model.variable("var2").remove("H_22");
model.variable("var2").remove("HI_11");
model.variable("var2").remove("HI_12");
model.variable("var2").remove("HI_13");
model.variable("var2").remove("HI_14");
model.variable("var2").remove("HI_15");
model.variable("var2").remove("HI_21");
model.variable("var2").remove("HI_22");
model.variable("var2").remove("HI_23");
model.variable("var2").remove("HI_24");
model.variable("var2").remove("HI_25");
model.variable("var2").remove("HH_11");
model.variable("var2").remove("HHI_11");
model.variable("var2").remove("HHI_12");
model.variable("var2").remove("HHI_13");
model.variable("var2").remove("HHI_14");
model.variable("var2").remove("HHI_15");
model.variable("var2").remove("invA_11");
model.variable("var2").remove("invA_12");
model.variable("var2").remove("invA_13");
model.variable("var2").remove("invA_14");
model.variable("var2").remove("invA_15");
model.variable("var2").remove("invA_21");
model.variable("var2").remove("invA_22");
model.variable("var2").remove("invA_23");
model.variable("var2").remove("invA_24");
model.variable("var2").remove("invA_25");
model.variable("var2").remove("invA_31");
model.variable("var2").remove("invA_32");
model.variable("var2").remove("invA_33");
model.variable("var2").remove("invA_34");
```

```
    model.variable("var2").remove("invA_35");
    model.variable("var2").remove("invA_41");
    model.variable("var2").remove("invA_42");
    model.variable("var2").remove("invA_43");
    model.variable("var2").remove("invA_44");
    model.variable("var2").remove("invA_45");
    model.variable("var2").remove("invA_51");
    model.variable("var2").remove("invA_52");
    model.variable("var2").remove("invA_53");
    model.variable("var2").remove("invA_54");
    model.variable("var2")
         .set("A_11", "-
(1/M_CH4)*((x_CO/D_CH4.CO)+(x_CO2/D_CH4.CO2)+(x_H2/D_CH4.H2)+(x_H2O/D_CH4.H2O)+(1/DK_CH4))", "(1) CH4");
    model.variable("var2").set("A_12", "x_CH4/(M_CO*D_CH4.CO)", "(2) CO");
    model.variable("var2").set("A_13", "x_CH4/(M_CO2*D_CH4.CO2)", "(3) CO2");
    model.variable("var2").set("A_14", "x_CH4/(M_H2*D_CH4.H2)", "(4) H2");
    model.variable("var2").set("A_15", "x_CH4/(M_H2O*D_CH4.H2O)", "(5) H2O");
    model.variable("var2").set("A_21", "x_CO/(M_CH4*D_CH4.CO)");
    model.variable("var2")
         .set("A_22", "-
(1/M_CO)*((x_CH4/D_CH4.CO)+(x_CO2/D_CO.CO2)+(x_H2/D_CO.H2)+(x_H2O/D_CO.H2O)+(1/DK_CO))");
    model.variable("var2").set("A_23", "x_CO/(M_CO2*D_CO.CO2)");
    model.variable("var2").set("A_24", "x_CO/(M_H2*D_CO.H2)");
    model.variable("var2").set("A_25", "x_CO/(M_H2O*D_CO.H2O)");
    model.variable("var2").set("A_31", "x_CO2/(M_CH4*D_CH4.CO2)");
    model.variable("var2").set("A_32", "x_CO2/(M_CO*D_CO.CO2)");
    model.variable("var2")
         .set("A_33", "-
(1/M_CO2)*((x_CH4/D_CH4.CO2)+(x_CO/D_CO.CO2)+(x_H2/D_CO2.H2)+(x_H2O/D_CO2.H2O)+(1/DK_CO2))");
    model.variable("var2").set("A_34", "x_CO2/(M_H2*D_CO2.H2)");
    model.variable("var2").set("A_35", "x_CO2/(M_H2O*D_CO2.H2O)");
    model.variable("var2").set("A_41", "x_H2/(M_CH4*D_CH4.H2)");
    model.variable("var2").set("A_42", "x_H2/(M_CO*D_CO.H2)");
    model.variable("var2").set("A_43", "x_H2/(M_CO2*D_CO2.H2)");
    model.variable("var2")
         .set("A_44", "-
(1/M_H2)*((x_CH4/D_CH4.H2)+(x_CO/D_CO.H2)+(x_CO2/D_CO2.H2)+(x_H2O/D_H2.H2O)+(1/DK_H2))");
    model.variable("var2").set("A_45", "x_H2/(M_H2O*D_H2.H2O)");
    model.variable("var2").set("A_51", "x_H2O/(M_CH4*D_CH4.H2O)");
    model.variable("var2").set("A_52", "x_H2O/(M_CO*D_CO.H2O)");
    model.variable("var2").set("A_53", "x_H2O/(M_CO2*D_CO2.H2O)");
    model.variable("var2").set("A_54", "x_H2O/(M_H2*D_H2.H2O)");
    model.variable("var2")
         .set("A_55", "-
(1/M_H2O)*((x_CH4/D_CH4.H2O)+(x_CO/D_CO.H2O)+(x_CO2/D_CO2.H2O)+(x_H2/D_H2.H2O)+(1/DK_H2O))");
    model.variable("var2").set("AI_11", "1");
    model.variable("var2").set("B_22", "A_11*A_22-A_21*A_12");
    model.variable("var2").set("B_23", "A_11*A_23-A_21*A_13");
    model.variable("var2").set("B_24", "A_11*A_24-A_21*A_14");
    model.variable("var2").set("B_25", "A_11*A_25-A_21*A_15");
    model.variable("var2").set("B_32", "A_11*A_32-A_31*A_12");
    model.variable("var2").set("B_33", "A_11*A_33-A_31*A_13");
    model.variable("var2").set("B_34", "A_11*A_34-A_31*A_14");
    model.variable("var2").set("B_35", "A_11*A_35-A_31*A_15");
    model.variable("var2").set("B_42", "A_11*A_42-A_41*A_12");
    model.variable("var2").set("B_43", "A_11*A_43-A_41*A_13");
    model.variable("var2").set("B_44", "A_11*A_44-A_41*A_14");
    model.variable("var2").set("B_45", "A_11*A_45-A_41*A_15");
    model.variable("var2").set("B_52", "A_11*A_52-A_51*A_12");
    model.variable("var2").set("B_53", "A_11*A_53-A_51*A_13");
    model.variable("var2").set("B_54", "A_11*A_54-A_51*A_14");
    model.variable("var2").set("B_55", "A_11*A_55-A_51*A_15");
    model.variable("var2").set("BI_21", "-A_21*AI_11");
    model.variable("var2").set("BI_31", "-A_31*AI_11");
    model.variable("var2").set("BI_41", "-A_41*AI_11");
    model.variable("var2").set("BI_51", "-A_51*AI_11");
    model.variable("var2").set("C_33", "B_22*B_33-B_32*B_23");
    model.variable("var2").set("C_34", "B_22*B_34-B_32*B_24");
    model.variable("var2").set("C_35", "B_22*B_35-B_32*B_25");
    model.variable("var2").set("C_43", "B_22*B_43-B_42*B_23");
    model.variable("var2").set("C_44", "B_22*B_44-B_42*B_24");
    model.variable("var2").set("C_45", "B_22*B_45-B_42*B_25");
    model.variable("var2").set("C_53", "B_22*B_53-B_52*B_23");
    model.variable("var2").set("C_54", "B_22*B_54-B_52*B_24");
    model.variable("var2").set("C_55", "B_22*B_55-B_52*B_25");
    model.variable("var2").set("CI_31", "B_22*BI_31-B_32*BI_21");
    model.variable("var2").set("CI_32", "-B_32*A_11");
    model.variable("var2").set("CI_33", "B_22*A_11");
    model.variable("var2").set("CI_41", "B_22*BI_41-B_42*BI_21");
    model.variable("var2").set("CI_42", "-B_42*A_11");
    model.variable("var2").set("CI_51", "B_22*BI_51-B_52*BI_21");
```

```
  model.variable("var2").set("CI_52", "-B_52*A_11");
  model.variable("var2").set("D_44", "C_33*C_44-C_43*C_34");
  model.variable("var2").set("D_45", "C_33*C_45-C_43*C_35");
  model.variable("var2").set("D_54", "C_33*C_54-C_53*C_34");
  model.variable("var2").set("D_55", "C_33*C_55-C_53*C_35");
  model.variable("var2").set("DI_41", "C_33*CI_41-C_43*CI_31");
  model.variable("var2").set("DI_42", "C_33*CI_42-C_43*CI_32");
  model.variable("var2").set("DI_43", "-C_43*CI_33");
  model.variable("var2").set("DI_44", "C_33*CI_33");
  model.variable("var2").set("DI_51", "C_33*CI_51-C_53*CI_31");
  model.variable("var2").set("DI_52", "C_33*CI_52-C_53*CI_32");
  model.variable("var2").set("DI_53", "-C_53*CI_33");
  model.variable("var2").set("E_55", "D_44*D_55-D_54*D_45");
  model.variable("var2").set("EI_51", "D_44*DI_51-D_54*DI_41");
  model.variable("var2").set("EI_52", "D_44*DI_52-D_54*DI_42");
  model.variable("var2").set("EI_53", "D_44*DI_53-D_54*DI_43");
  model.variable("var2").set("EI_54", "-D_54*DI_44");
  model.variable("var2").set("EI_55", "D_44*DI_44");
  model.variable("var2").set("F_11", "E_55*A_11");
  model.variable("var2").set("F_12", "E_55*A_12");
  model.variable("var2").set("F_13", "A_13*E_55");
  model.variable("var2").set("F_14", "A_14*E_55");
  model.variable("var2").set("F_22", "E_55*B_22");
  model.variable("var2").set("F_23", "B_23*E_55");
  model.variable("var2").set("F_24", "B_24*E_55");
  model.variable("var2").set("F_33", "C_33*E_55");
  model.variable("var2").set("F_34", "C_34*E_55");
  model.variable("var2").set("F_44", "D_44*E_55");
  model.variable("var2").set("FI_11", "E_55*AI_11-EI_51*A_15");
  model.variable("var2").set("FI_12", "-EI_52*A_15");
  model.variable("var2").set("FI_13", "-EI_53*A_15");
  model.variable("var2").set("FI_14", "-EI_54*A_15");
  model.variable("var2").set("FI_15", "-EI_55*A_15");
  model.variable("var2").set("FI_21", "E_55*BI_21-EI_51*B_25");
  model.variable("var2").set("FI_22", "E_55*BI_22-EI_52*B_25");
  model.variable("var2").set("FI_23", "-EI_53*B_25");
  model.variable("var2").set("FI_24", "-EI_54*B_25");
  model.variable("var2").set("FI_25", "-EI_55*B_25");
  model.variable("var2").set("FI_31", "E_55*CI_31-EI_51*C_35");
  model.variable("var2").set("FI_32", "E_55*CI_32-EI_52*C_35");
  model.variable("var2").set("FI_33", "E_55*CI_33-EI_53*C_35");
  model.variable("var2").set("FI_34", "-EI_54*C_35");
  model.variable("var2").set("FI_35", "-EI_55*C_35");
  model.variable("var2").set("FI_41", "E_55*DI_41-EI_51*D_45");
  model.variable("var2").set("FI_42", "E_55*DI_42-EI_52*D_45");
  model.variable("var2").set("FI_43", "E_55*DI_43-EI_53*D_45");
  model.variable("var2").set("FI_44", "E_55*DI_44-EI_54*D_45");
  model.variable("var2").set("FI_45", "-EI_55*D_45");
  model.variable("var2").set("G_11", "F_44*F_11");
  model.variable("var2").set("G_12", "F_44*F_12");
  model.variable("var2").set("G_13", "F_44*F_13");
  model.variable("var2").set("G_22", "F_44*F_22");
  model.variable("var2").set("G_23", "F_44*F_23");
  model.variable("var2").set("G_33", "F_44*F_33");
  model.variable("var2").set("GI_11", "F_44*FI_11-F_14*FI_41");
  model.variable("var2").set("GI_12", "F_44*FI_12-F_14*FI_42");
  model.variable("var2").set("GI_13", "F_44*FI_13-F_14*FI_43");
  model.variable("var2").set("GI_14", "F_44*FI_14-F_14*FI_44");
  model.variable("var2").set("GI_15", "F_44*FI_15-F_14*FI_45");
  model.variable("var2").set("GI_21", "F_44*FI_21-F_24*FI_41");
  model.variable("var2").set("GI_22", "F_44*FI_22-F_24*FI_42");
  model.variable("var2").set("GI_23", "F_44*FI_23-F_24*FI_43");
  model.variable("var2").set("GI_24", "F_44*FI_24-F_24*FI_44");
  model.variable("var2").set("GI_25", "F_44*FI_25-F_24*FI_45");
  model.variable("var2").set("GI_31", "F_44*FI_31-F_34*FI_41");
  model.variable("var2").set("GI_32", "F_44*FI_32-F_34*FI_42");
  model.variable("var2").set("GI_33", "F_44*FI_33-F_34*FI_43");
  model.variable("var2").set("GI_34", "F_44*FI_34-F_34*FI_44");

  return model;
}

public static Model run3(Model model) {
  model.variable("var2").set("GI_35", "F_44*FI_35-F_34*FI_45");
  model.variable("var2").set("H_11", "G_33*G_11");
  model.variable("var2").set("H_12", "G_33*G_12");
  model.variable("var2").set("H_22", "G_33*G_22");
  model.variable("var2").set("HI_11", "G_33*GI_11-G_13*GI_31");
  model.variable("var2").set("HI_12", "G_33*GI_12-G_13*GI_32");
  model.variable("var2").set("HI_13", "G_33*GI_13-G_13*GI_33");
  model.variable("var2").set("HI_14", "G_33*GI_14-G_13*GI_34");
```

```
model.variable("var2").set("HI_15", "G_33*GI_15-G_13*GI_35");
model.variable("var2").set("HI_21", "G_33*GI_21-G_23*GI_31");
model.variable("var2").set("HI_22", "G_33*GI_22-G_23*GI_32");
model.variable("var2").set("HI_23", "G_33*GI_23-G_23*GI_33");
model.variable("var2").set("HI_24", "G_33*GI_24-G_23*GI_34");
model.variable("var2").set("HI_25", "G_33*GI_25-G_23*GI_35");
model.variable("var2").set("HH_11", "H_22*H_11");
model.variable("var2").set("HHI_11", "H_22*HI_11-H_12*HI_21");
model.variable("var2").set("HHI_12", "H_22*HI_12-H_12*HI_22");
model.variable("var2").set("HHI_13", "H_22*HI_13-H_12*HI_23");
model.variable("var2").set("HHI_14", "H_22*HI_14-H_12*HI_24");
model.variable("var2").set("HHI_15", "H_22*HI_15-H_12*HI_25");
model.variable("var2").set("invA_11", "HHI_11/HH_11");
model.variable("var2").set("invA_12", "HHI_12/HH_11");
model.variable("var2").set("invA_13", "HHI_13/HH_11");
model.variable("var2").set("invA_14", "HHI_14/HH_11");
model.variable("var2").set("invA_15", "HHI_15/HH_11");
model.variable("var2").set("invA_21", "HI_21/H_22");
model.variable("var2").set("invA_22", "HI_22/H_22");
model.variable("var2").set("invA_23", "HI_23/H_22");
model.variable("var2").set("invA_24", "HI_24/H_22");
model.variable("var2").set("invA_25", "HI_25/H_22");
model.variable("var2").set("invA_31", "GI_31/G_33");
model.variable("var2").set("invA_32", "GI_32/G_33");
model.variable("var2").set("invA_33", "GI_33/G_33");
model.variable("var2").set("invA_34", "GI_34/G_33");
model.variable("var2").set("invA_35", "GI_35/G_33");
model.variable("var2").set("invA_41", "FI_41/F_44");
model.variable("var2").set("invA_42", "FI_42/F_44");
model.variable("var2").set("invA_43", "FI_43/F_44");
model.variable("var2").set("invA_44", "FI_44/F_44");
model.variable("var2").set("invA_45", "FI_45/F_44");
model.variable("var2").set("invA_51", "EI_51/E_55");
model.variable("var2").set("invA_52", "EI_52/E_55");
model.variable("var2").set("invA_53", "EI_53/E_55");
model.variable("var2").set("invA_54", "EI_54/E_55");
model.variable("var2").set("invA_55", "EI_55/E_55");
model.variable("var2").remove("A_11");
model.variable("var2").remove("A_12");
model.variable("var2").remove("A_13");
model.variable("var2").remove("A_14");
model.variable("var2").remove("A_15");
model.variable("var2").remove("A_21");
model.variable("var2").remove("A_22");
model.variable("var2").remove("A_23");
model.variable("var2").remove("A_24");
model.variable("var2").remove("A_25");
model.variable("var2").remove("A_31");
model.variable("var2").remove("A_32");
model.variable("var2").remove("A_33");
model.variable("var2").remove("A_34");
model.variable("var2").remove("A_35");
model.variable("var2").remove("A_41");
model.variable("var2").remove("A_42");
model.variable("var2").remove("A_43");
model.variable("var2").remove("A_44");
model.variable("var2").remove("A_45");
model.variable("var2").remove("A_51");
model.variable("var2").remove("A_52");
model.variable("var2").remove("A_53");
model.variable("var2").remove("A_54");
model.variable("var2").remove("A_55");
model.variable("var2").remove("AI_11");
model.variable("var2").remove("B_22");
model.variable("var2").remove("B_23");
model.variable("var2").remove("B_24");
model.variable("var2").remove("B_25");
model.variable("var2").remove("B_32");
model.variable("var2").remove("B_33");
model.variable("var2").remove("B_34");
model.variable("var2").remove("B_35");
model.variable("var2").remove("B_42");
model.variable("var2").remove("B_43");
model.variable("var2").remove("B_44");
model.variable("var2").remove("B_45");
model.variable("var2").remove("B_52");
model.variable("var2").remove("B_53");
model.variable("var2").remove("B_54");
model.variable("var2").remove("B_55");
model.variable("var2").remove("BI_21");
model.variable("var2").remove("BI_31");
```

```
model.variable("var2").remove("BI_41");
model.variable("var2").remove("BI_51");
model.variable("var2").remove("C_33");
model.variable("var2").remove("C_34");
model.variable("var2").remove("C_35");
model.variable("var2").remove("C_43");
model.variable("var2").remove("C_44");
model.variable("var2").remove("C_45");
model.variable("var2").remove("C_53");
model.variable("var2").remove("C_54");
model.variable("var2").remove("C_55");
model.variable("var2").remove("CI_31");
model.variable("var2").remove("CI_32");
model.variable("var2").remove("CI_33");
model.variable("var2").remove("CI_41");
model.variable("var2").remove("CI_42");
model.variable("var2").remove("CI_51");
model.variable("var2").remove("CI_52");
model.variable("var2").remove("D_44");
model.variable("var2").remove("D_45");
model.variable("var2").remove("D_54");
model.variable("var2").remove("D_55");
model.variable("var2").remove("DI_41");
model.variable("var2").remove("DI_42");
model.variable("var2").remove("DI_43");
model.variable("var2").remove("DI_44");
model.variable("var2").remove("DI_51");
model.variable("var2").remove("DI_52");
model.variable("var2").remove("DI_53");
model.variable("var2").remove("E_55");
model.variable("var2").remove("EI_51");
model.variable("var2").remove("EI_52");
model.variable("var2").remove("EI_53");
model.variable("var2").remove("EI_54");
model.variable("var2").remove("EI_55");
model.variable("var2").remove("F_11");
model.variable("var2").remove("F_12");
model.variable("var2").remove("F_13");
model.variable("var2").remove("F_14");
model.variable("var2").remove("F_22");
model.variable("var2").remove("F_23");
model.variable("var2").remove("F_24");
model.variable("var2").remove("F_33");
model.variable("var2").remove("F_34");
model.variable("var2").remove("F_44");
model.variable("var2").remove("FI_11");
model.variable("var2").remove("FI_12");
model.variable("var2").remove("FI_13");
model.variable("var2").remove("FI_14");
model.variable("var2").remove("FI_15");
model.variable("var2").remove("FI_21");
model.variable("var2").remove("FI_22");
model.variable("var2").remove("FI_23");
model.variable("var2").remove("FI_24");
model.variable("var2").remove("FI_25");
model.variable("var2").remove("FI_31");
model.variable("var2").remove("FI_32");
model.variable("var2").remove("FI_33");
model.variable("var2").remove("FI_34");
model.variable("var2").remove("FI_35");
model.variable("var2").remove("FI_41");
model.variable("var2").remove("FI_42");
model.variable("var2").remove("FI_43");
model.variable("var2").remove("FI_44");
model.variable("var2").remove("FI_45");
model.variable("var2").remove("G_11");
model.variable("var2").remove("G_12");
model.variable("var2").remove("G_13");
model.variable("var2").remove("G_22");
model.variable("var2").remove("G_23");
model.variable("var2").remove("G_33");
model.variable("var2").remove("GI_11");
model.variable("var2").remove("GI_12");
model.variable("var2").remove("GI_13");
model.variable("var2").remove("GI_14");
model.variable("var2").remove("GI_15");
model.variable("var2").remove("GI_21");
model.variable("var2").remove("GI_22");
model.variable("var2").remove("GI_23");
model.variable("var2").remove("GI_24");
model.variable("var2").remove("GI_25");
```

```
    model.variable("var2").remove("GI_31");
    model.variable("var2").remove("GI_32");
    model.variable("var2").remove("GI_33");
    model.variable("var2").remove("GI_34");
    model.variable("var2").remove("GI_35");
    model.variable("var2").remove("H_11");
    model.variable("var2").remove("H_12");
    model.variable("var2").remove("H_22");
    model.variable("var2").remove("HI_11");
    model.variable("var2").remove("HI_12");
    model.variable("var2").remove("HI_13");
    model.variable("var2").remove("HI_14");
    model.variable("var2").remove("HI_15");
    model.variable("var2").remove("HI_21");
    model.variable("var2").remove("HI_22");
    model.variable("var2").remove("HI_23");
    model.variable("var2").remove("HI_24");
    model.variable("var2").remove("HI_25");
    model.variable("var2").remove("HH_11");
    model.variable("var2").remove("HHI_11");
    model.variable("var2").remove("HHI_12");
    model.variable("var2").remove("HHI_13");
    model.variable("var2").remove("HHI_14");
    model.variable("var2").remove("HHI_15");
    model.variable("var2").remove("invA_11");
    model.variable("var2").remove("invA_12");
    model.variable("var2").remove("invA_13");
    model.variable("var2").remove("invA_14");
    model.variable("var2").remove("invA_15");
    model.variable("var2").remove("invA_21");
    model.variable("var2").remove("invA_22");
    model.variable("var2").remove("invA_23");
    model.variable("var2").remove("invA_24");
    model.variable("var2").remove("invA_25");
    model.variable("var2").remove("invA_31");
    model.variable("var2").remove("invA_32");
    model.variable("var2").remove("invA_33");
    model.variable("var2").remove("invA_34");
    model.variable("var2").remove("invA_35");
    model.variable("var2").remove("invA_41");
    model.variable("var2").remove("invA_42");
    model.variable("var2").remove("invA_43");
    model.variable("var2").remove("invA_44");
    model.variable("var2").remove("invA_45");
    model.variable("var2").remove("invA_51");
    model.variable("var2").remove("invA_52");
    model.variable("var2").remove("invA_53");
    model.variable("var2").remove("invA_54");
    model.variable("var2").remove("invA_55");
    model.variable("var2")
         .set("A_11", "-
(1/M_CH4)*((x_CO/Deff_CH4.CO)+(x_CO2/Deff_CH4.CO2)+(x_H2/Deff_CH4.H2)+(x_H2O/Deff_CH4.H2O)+(1/DK_CH4))", "(1)
CH4");
    model.variable("var2").set("A_12", "x_CH4/(M_CO*Deff_CH4.CO)", "(2) CO");
    model.variable("var2").set("A_13", "x_CH4/(M_CO2*Deff_CH4.CO2)", "(3) CO2");
    model.variable("var2").set("A_14", "x_CH4/(M_H2*Deff_CH4.H2)", "(4) H2");
    model.variable("var2").set("A_15", "x_CH4/(M_H2O*Deff_CH4.H2O)", "(5) H2O");
    model.variable("var2").set("A_21", "x_CO/(M_CH4*Deff_CH4.CO)");
    model.variable("var2")
         .set("A_22", "-
(1/M_CO)*((x_CH4/Deff_CH4.CO)+(x_CO2/Deff_CO.CO2)+(x_H2/Deff_CO.H2)+(x_H2O/Deff_CO.H2O)+(1/DK_CO))");
    model.variable("var2").set("A_23", "x_CO/(M_CO2*Deff_CO.CO2)");
    model.variable("var2").set("A_24", "x_CO/(M_H2*Deff_CO.H2)");
    model.variable("var2").set("A_25", "x_CO/(M_H2O*Deff_CO.H2O)");
    model.variable("var2").set("A_31", "x_CO2/(M_CH4*Deff_CH4.CO2)");
    model.variable("var2").set("A_32", "x_CO2/(M_CO*Deff_CO.CO2)");
    model.variable("var2")
         .set("A_33", "-
(1/M_CO2)*((x_CH4/Deff_CH4.CO2)+(x_CO/Deff_CO.CO2)+(x_H2/Deff_CO2.H2)+(x_H2O/Deff_CO2.H2O)+(1/DK_CO2))");
    model.variable("var2").set("A_34", "x_CO2/(M_H2*Deff_CO2.H2)");
    model.variable("var2").set("A_35", "x_CO2/(M_H2O*Deff_CO2.H2O)");
    model.variable("var2").set("A_41", "x_H2/(M_CH4*Deff_CH4.H2)");
    model.variable("var2").set("A_42", "x_H2/(M_CO*Deff_CO.H2)");
    model.variable("var2").set("A_43", "x_H2/(M_CO2*Deff_CO2.H2)");
    model.variable("var2")
         .set("A_44", "-
(1/M_H2)*((x_CH4/Deff_CH4.H2)+(x_CO/Deff_CO.H2)+(x_CO2/Deff_CO2.H2)+(x_H2O/Deff_H2.H2O)+(1/DK_H2))");
    model.variable("var2").set("A_45", "x_H2/(M_H2O*Deff_H2.H2O)");
    model.variable("var2").set("A_51", "x_H2O/(M_CH4*Deff_CH4.H2O)");
    model.variable("var2").set("A_52", "x_H2O/(M_CO*Deff_CO.H2O)");
    model.variable("var2").set("A_53", "x_H2O/(M_CO2*Deff_CO2.H2O)");
```

```
    model.variable("var2").set("A_54", "x_H2O/(M_H2*Deff_H2.H2O)");
    model.variable("var2")
         .set("A_55", "-
(1/M_H2O)*((x_CH4/Deff_CH4.H2O)+(x_CO/Deff_CO.H2O)+(x_CO2/Deff_CO2.H2O)+(x_H2/Deff_H2.H2O)+(1/DK_H2O))");
    model.variable("var2").set("AI_11", "1");
    model.variable("var2").set("B_22", "A_11*A_22-A_21*A_12");
    model.variable("var2").set("B_23", "A_11*A_23-A_21*A_13");
    model.variable("var2").set("B_24", "A_11*A_24-A_21*A_14");
    model.variable("var2").set("B_25", "A_11*A_25-A_21*A_15");
    model.variable("var2").set("B_32", "A_11*A_32-A_31*A_12");
    model.variable("var2").set("B_33", "A_11*A_33-A_31*A_13");
    model.variable("var2").set("B_34", "A_11*A_34-A_31*A_14");
    model.variable("var2").set("B_35", "A_11*A_35-A_31*A_15");
    model.variable("var2").set("B_42", "A_11*A_42-A_41*A_12");
    model.variable("var2").set("B_43", "A_11*A_43-A_41*A_13");
    model.variable("var2").set("B_44", "A_11*A_44-A_41*A_14");
    model.variable("var2").set("B_45", "A_11*A_45-A_41*A_15");
    model.variable("var2").set("B_52", "A_11*A_52-A_51*A_12");
    model.variable("var2").set("B_53", "A_11*A_53-A_51*A_13");
    model.variable("var2").set("B_54", "A_11*A_54-A_51*A_14");
    model.variable("var2").set("B_55", "A_11*A_55-A_51*A_15");
    model.variable("var2").set("BI_21", "-A_21*AI_11");
    model.variable("var2").set("BI_31", "-A_31*AI_11");
    model.variable("var2").set("BI_41", "-A_41*AI_11");
    model.variable("var2").set("BI_51", "-A_51*AI_11");
    model.variable("var2").set("C_33", "B_22*B_33-B_32*B_23");
    model.variable("var2").set("C_34", "B_22*B_34-B_32*B_24");
    model.variable("var2").set("C_35", "B_22*B_35-B_32*B_25");
    model.variable("var2").set("C_43", "B_22*B_43-B_42*B_23");
    model.variable("var2").set("C_44", "B_22*B_44-B_42*B_24");
    model.variable("var2").set("C_45", "B_22*B_45-B_42*B_25");
    model.variable("var2").set("C_53", "B_22*B_53-B_52*B_23");
    model.variable("var2").set("C_54", "B_22*B_54-B_52*B_24");
    model.variable("var2").set("C_55", "B_22*B_55-B_52*B_25");
    model.variable("var2").set("CI_31", "B_22*BI_31-B_32*BI_21");
    model.variable("var2").set("CI_32", "-B_32*A_11");
    model.variable("var2").set("CI_33", "B_22*A_11");
    model.variable("var2").set("CI_41", "B_22*BI_41-B_42*BI_21");
    model.variable("var2").set("CI_42", "-B_42*A_11");
    model.variable("var2").set("CI_51", "B_22*BI_51-B_52*BI_21");
    model.variable("var2").set("CI_52", "-B_52*A_11");
    model.variable("var2").set("D_44", "C_33*C_44-C_43*C_34");
    model.variable("var2").set("D_45", "C_33*C_45-C_43*C_35");
    model.variable("var2").set("D_54", "C_33*C_54-C_53*C_34");
    model.variable("var2").set("D_55", "C_33*C_55-C_53*C_35");
    model.variable("var2").set("DI_41", "C_33*CI_41-C_43*CI_31");
    model.variable("var2").set("DI_42", "C_33*CI_42-C_43*CI_32");
    model.variable("var2").set("DI_43", "-C_43*CI_33");
    model.variable("var2").set("DI_44", "C_33*CI_33");
    model.variable("var2").set("DI_51", "C_33*CI_51-C_53*CI_31");
    model.variable("var2").set("DI_52", "C_33*CI_52-C_53*CI_32");
    model.variable("var2").set("DI_53", "-C_53*CI_33");
    model.variable("var2").set("E_55", "D_44*D_55-D_54*D_45");
    model.variable("var2").set("EI_51", "D_44*DI_51-D_54*DI_41");
    model.variable("var2").set("EI_52", "D_44*DI_52-D_54*DI_42");
    model.variable("var2").set("EI_53", "D_44*DI_53-D_54*DI_43");
    model.variable("var2").set("EI_54", "-D_54*DI_44");
    model.variable("var2").set("EI_55", "D_44*DI_44");
    model.variable("var2").set("F_11", "E_55*A_11");
    model.variable("var2").set("F_12", "E_55*A_12");
    model.variable("var2").set("F_13", "A_13*E_55");
    model.variable("var2").set("F_14", "A_14*E_55");
    model.variable("var2").set("F_22", "E_55*B_22");
    model.variable("var2").set("F_23", "B_23*E_55");
    model.variable("var2").set("F_24", "B_24*E_55");
    model.variable("var2").set("F_33", "C_33*E_55");
    model.variable("var2").set("F_34", "C_34*E_55");
    model.variable("var2").set("F_44", "D_44*E_55");
    model.variable("var2").set("FI_11", "E_55*AI_11-EI_51*A_15");
    model.variable("var2").set("FI_12", "-EI_52*A_15");
    model.variable("var2").set("FI_13", "-EI_53*A_15");
    model.variable("var2").set("FI_14", "-EI_54*A_15");
    model.variable("var2").set("FI_15", "-EI_55*A_15");
    model.variable("var2").set("FI_21", "E_55*BI_21-EI_51*B_25");
    model.variable("var2").set("FI_22", "E_55*BI_22-EI_52*B_25");
    model.variable("var2").set("FI_23", "-EI_53*B_25");
    model.variable("var2").set("FI_24", "-EI_54*B_25");
    model.variable("var2").set("FI_25", "-EI_55*B_25");
    model.variable("var2").set("FI_31", "E_55*CI_31-EI_51*C_35");
    model.variable("var2").set("FI_32", "E_55*CI_32-EI_52*C_35");
    model.variable("var2").set("FI_33", "E_55*CI_33-EI_53*C_35");
```

```
model.variable("var2").set("FI_34", "-EI_54*C_35");
model.variable("var2").set("FI_35", "-EI_55*C_35");
model.variable("var2").set("FI_41", "E_55*DI_41-EI_51*D_45");
model.variable("var2").set("FI_42", "E_55*DI_42-EI_52*D_45");
model.variable("var2").set("FI_43", "E_55*DI_43-EI_53*D_45");
model.variable("var2").set("FI_44", "E_55*DI_44-EI_54*D_45");
model.variable("var2").set("FI_45", "-EI_55*D_45");
model.variable("var2").set("G_11", "F_44*F_11");
model.variable("var2").set("G_12", "F_44*F_12");
model.variable("var2").set("G_13", "F_44*F_13");
model.variable("var2").set("G_22", "F_44*F_22");
model.variable("var2").set("G_23", "F_44*F_23");
model.variable("var2").set("G_33", "F_44*F_33");
model.variable("var2").set("GI_11", "F_44*FI_11-F_14*FI_41");
model.variable("var2").set("GI_12", "F_44*FI_12-F_14*FI_42");
model.variable("var2").set("GI_13", "F_44*FI_13-F_14*FI_43");
model.variable("var2").set("GI_14", "F_44*FI_14-F_14*FI_44");
model.variable("var2").set("GI_15", "F_44*FI_15-F_14*FI_45");
model.variable("var2").set("GI_21", "F_44*FI_21-F_24*FI_41");
model.variable("var2").set("GI_22", "F_44*FI_22-F_24*FI_42");
model.variable("var2").set("GI_23", "F_44*FI_23-F_24*FI_43");
model.variable("var2").set("GI_24", "F_44*FI_24-F_24*FI_44");
model.variable("var2").set("GI_25", "F_44*FI_25-F_24*FI_45");
model.variable("var2").set("GI_31", "F_44*FI_31-F_34*FI_41");
model.variable("var2").set("GI_32", "F_44*FI_32-F_34*FI_42");
model.variable("var2").set("GI_33", "F_44*FI_33-F_34*FI_43");
model.variable("var2").set("GI_34", "F_44*FI_34-F_34*FI_44");
model.variable("var2").set("GI_35", "F_44*FI_35-F_34*FI_45");
model.variable("var2").set("H_11", "G_33*G_11");
model.variable("var2").set("H_12", "G_33*G_12");
model.variable("var2").set("H_22", "G_33*G_22");
model.variable("var2").set("HI_11", "G_33*GI_11-G_13*GI_31");
model.variable("var2").set("HI_12", "G_33*GI_12-G_13*GI_32");
model.variable("var2").set("HI_13", "G_33*GI_13-G_13*GI_33");
model.variable("var2").set("HI_14", "G_33*GI_14-G_13*GI_34");
model.variable("var2").set("HI_15", "G_33*GI_15-G_13*GI_35");
model.variable("var2").set("HI_21", "G_33*GI_21-G_23*GI_31");
model.variable("var2").set("HI_22", "G_33*GI_22-G_23*GI_32");
model.variable("var2").set("HI_23", "G_33*GI_23-G_23*GI_33");
model.variable("var2").set("HI_24", "G_33*GI_24-G_23*GI_34");
model.variable("var2").set("HI_25", "G_33*GI_25-G_23*GI_35");
model.variable("var2").set("HH_11", "H_22*H_11");
model.variable("var2").set("HHI_11", "H_22*HI_11-H_12*HI_21");
model.variable("var2").set("HHI_12", "H_22*HI_12-H_12*HI_22");
model.variable("var2").set("HHI_13", "H_22*HI_13-H_12*HI_23");
model.variable("var2").set("HHI_14", "H_22*HI_14-H_12*HI_24");
model.variable("var2").set("HHI_15", "H_22*HI_15-H_12*HI_25");
model.variable("var2").set("invA_11", "HHI_11/HH_11");
model.variable("var2").set("invA_12", "HHI_12/HH_11");
model.variable("var2").set("invA_13", "HHI_13/HH_11");
model.variable("var2").set("invA_14", "HHI_14/HH_11");
model.variable("var2").set("invA_15", "HHI_15/HH_11");
model.variable("var2").set("invA_21", "HI_21/H_22");
model.variable("var2").set("invA_22", "HI_22/H_22");
model.variable("var2").set("invA_23", "HI_23/H_22");
model.variable("var2").set("invA_24", "HI_24/H_22");
model.variable("var2").set("invA_25", "HI_25/H_22");
model.variable("var2").set("invA_31", "GI_31/G_33");
model.variable("var2").set("invA_32", "GI_32/G_33");
model.variable("var2").set("invA_33", "GI_33/G_33");
model.variable("var2").set("invA_34", "GI_34/G_33");
model.variable("var2").set("invA_35", "GI_35/G_33");
model.variable("var2").set("invA_41", "FI_41/F_44");
model.variable("var2").set("invA_42", "FI_42/F_44");
model.variable("var2").set("invA_43", "FI_43/F_44");
model.variable("var2").set("invA_44", "FI_44/F_44");
model.variable("var2").set("invA_45", "FI_45/F_44");
model.variable("var2").set("invA_51", "EI_51/E_55");
model.variable("var2").set("invA_52", "EI_52/E_55");
model.variable("var2").set("invA_53", "EI_53/E_55");
model.variable("var2").set("invA_54", "EI_54/E_55");
model.variable("var2").set("invA_55", "EI_55/E_55");
model.variable("var2").set("FI_22", "E_55*A_11-EI_52*B_25");

model.physics("c").field("dimensionless").field("X");

model.label("New Pellet Model 2016.mph");

model.physics("c").feature("cfeq1").setIndex("c", "x*c_TOT*1", 0);
model.physics("c").feature("cfeq1").setIndex("c", "x*cTOT*1", 0);
model.physics("c").feature("cfeq1").setIndex("c", "x*c_TOT*invA_11", 0);
```

```
model.variable().create("var5");
model.variable("var5").model("comp1");
model.variable("var5").label("General functions");
model.variable("var5").set("c_TOT", "c_CH4+c_CO+c_CO2+c_H2+c_H2O");
model.variable("var5").set("c_CH4", "x_CH4*p_Pellet/(R_const*T)");
model.variable("var5").set("p_Pellet", "(p_in+p_out)/2");
model.variable("var5").set("c_CO", "x_CO*p_Pellet/(R_const*T)");
model.variable("var5").set("c_CO2", "x_CO2*p_Pellet/(R_const*T)");
model.variable("var5").set("c_H2", "x_H2*p_Pellet/(R_const*T)");
model.variable("var5").set("c_H2O", "x_H2O*p_Pellet/(R_const*T)");
model.variable("var5").set("pelletpressure", "0000000000000000000000000000000000000000000000");

model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_11", 0);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_21", 1);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_31", 2);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_41", 3);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_51", 4);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_12", 5);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_12", 6);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_12", 7);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_12", 8);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_12", 9);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_22", 6);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_32", 7);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_42", 8);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_52", 9);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_13", 10);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_23", 11);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_33", 12);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_43", 13);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_53", 14);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_14", 15);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_24", 16);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_24", 17);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_34", 17);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_44", 18);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_54", 19);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_15", 20);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_25", 21);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_35", 22);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_45", 23);
model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*invA_55", 24);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_cat", 0);

model.variable("var5").set("r_CH4", "-(r1-r3)");
model.variable("var5").set("r_CO", "r1-r2");
model.variable("var5").set("r_CO2", "r2+r3");
model.variable("var5").set("r_H2", "3*r1+r2+4*r3");
model.variable("var5").set("r_H2O", "-r1-r2-2*r3");

model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst", 0);
model.physics("c").prop("Units").set("SourceTermQuantity", "none");
model.physics("c").prop("Units").set("CustomSourceTermUnit", "m^2*kg/(m^3*h)");
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst", 1);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst", 2);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst", 3);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst", 4);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CO*M_CH4*rho_catalyst", 1);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CO2*M_CH4*rho_catalyst", 2);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_H2*M_CH4*rho_catalyst", 3);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_H2O*M_H2O*rho_catalyst", 4);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_H2*M_H2*rho_catalyst", 3);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CO*M_CO*rho_catalyst", 1);
model.physics("c").feature("cfeq1").setIndex("f", "(x^2)*r_CO2*M_CO2*rho_catalyst", 2);
model.physics("c").feature("dir1").selection().set(new int[]{2});
model.physics("c").feature("dir1").setIndex("r", "x_CH4_0", 0);
model.physics("c").feature("dir1").setIndex("r", "x_CH4_0", 1);
model.physics("c").feature("dir1").setIndex("r", "x_CH4_0", 2);
model.physics("c").feature("dir1").setIndex("r", "x_CH4_0", 3);
model.physics("c").feature("dir1").setIndex("r", "x_H2O_0", 4);
model.physics("c").feature("dir1").setIndex("r", "x_H2_0", 3);
model.physics("c").feature("dir1").setIndex("r", "x_CO2_0", 2);
model.physics("c").feature("dir1").setIndex("r", "x_CO_0", 1);

model.param().set("kk", "0.208[W/(m*K)]");
model.param().descr("kk", "Pellet overall conductivity");

model.physics("g").feature("gfeq1").setIndex("f", "-(x^2)*((r1*dH1)+(r2*dH2)+(r3*dH3))*rho_catalyst", 0);
model.physics("g").feature("gfeq1").setIndex("Ga", "-kk*(x^2)*Tx", 0);
model.physics("g").feature("init1").set("T", "T_in");
```

```
model.physics("g").feature().create("dir1", "DirichletBoundary", 0);
model.physics("g").feature("dir1").setIndex("r", "T_in", 0);

model.mesh("mesh1").run();

model.sol().create("sol1");
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result().create("pg1", 1);
model.result("pg1").set("data", "dset1");
model.result("pg1").create("lngr1", "LineGraph");
model.result("pg1").feature("lngr1").set("xdata", "expr");
model.result("pg1").feature("lngr1").set("xdataexpr", "x");
model.result("pg1").feature("lngr1").selection().all();
model.result("pg1").feature("lngr1").set("expr", "x_CH4");
model.result().create("pg2", 1);
model.result("pg2").set("data", "dset1");
model.result("pg2").create("lngr1", "LineGraph");
model.result("pg2").feature("lngr1").set("xdata", "expr");
model.result("pg2").feature("lngr1").set("xdataexpr", "x");
model.result("pg2").feature("lngr1").selection().all();
model.result("pg2").feature("lngr1").set("expr", "T");

model.study("std1").feature("stat").set("physselection", "c");
model.study("std1").feature("stat").set("activate", new String[]{"c", "on", "g", "off"});

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.study("std1").feature("stat").set("physselection", "c");
model.study("std1").feature("stat").set("activate", new String[]{"c", "off", "g", "off"});
model.study("std1").feature("stat").set("physselection", "c");
model.study("std1").feature("stat").set("activate", new String[]{"c", "off", "g", "on"});

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
```

```
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");

    return model;
  }

  public static Model run4(Model model) {
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").attach("std1");

    model.study("std1").feature("stat").set("physselection", "c");
    model.study("std1").feature("stat").set("activate", new String[]{"c", "on", "g", "on"});

    model.mesh("mesh1").autoMeshSize(3);
    model.mesh("mesh1").run();

    model.sol("sol1").feature("s1").feature("aDef").set("matherr", "off");
    model.sol("sol1").feature("s1").feature("aDef").set("convinfo", "detailed");

    model.study("std1").feature("stat").set("useparam", "on");

    model.variable("var1")
         .set("r1", "k1*sqrt(p_pellet*x_H2/1[bar])*((p_pellet*(x_CH4*x_H2O)/x_H2)-
((p_pellet^3)*(x_H2^2)*x_CO/K1))/(DEN_star)^2");
    model.variable("var5").rename("p_Pellet", "p_pellet");
    model.variable("var5").set("p_pellet", "(p_in+p_out)/2");
    model.variable("var5").set("c_CH4", "x_CH4*p_pellet/(R_const*T)");
    model.variable("var5").set("c_CO", "x_CO*p_pellet/(R_const*T)");
    model.variable("var5").set("c_CO2", "x_CO2*p_pellet/(R_const*T)");
    model.variable("var5").set("c_H2", "x_H2*p_pellet/(R_const*T)");
    model.variable("var5").set("c_H2O", "x_H2O*p_pellet/(R_const*T)");
    model.variable("var1")
         .set("r2", "k2*p_pellet*x_H2*(((p_pellet^2)*x_CO*x_H2O)-((p_pellet^2)*x_H2*x_CO2/K2))/(DEN_star)^2");
    model.variable("var1")
         .set("r3", "k3*sqrt(p_pellet*x_H2/1[bar])*((p_pellet*x_CH4*x_H2O/(x_H2^2))-
((p_pellet^3)*(x_H2^2)*x_CO2/K3))/(DEN_star)^2");
    model.variable("var1")
         .set("DEN_star",
"(p_pellet*x_H2)+((p_pellet^2)*K_CO*x_H2*x_CO)+((p_pellet^2)*K_H2*(x_H2^2))+((p_pellet^2)*K_CH4*x_H2*x_CH4)+(p_
pellet*K_H2O*x_H2O)");
    model.variable("var1").remove("r3_A");
    model.variable("var1").remove("r2_A");
    model.variable("var1").remove("r1_A");

    model.sol().remove("sol1");

    model.study("std1").feature("stat").set("useparam", "off");

    model.sol().create("sol1");
    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.result().create("pg1", 1);
    model.result("pg1").set("data", "dset1");
    model.result("pg1").create("lngr1", "LineGraph");
    model.result("pg1").feature("lngr1").set("xdata", "expr");
    model.result("pg1").feature("lngr1").set("xdataexpr", "x");
    model.result("pg1").feature("lngr1").selection().all();
    model.result("pg1").feature("lngr1").set("expr", "x_CH4");
```

```
    model.result().create("pg2", 1);
    model.result("pg2").set("data", "dset1");
    model.result("pg2").create("lngr1", "LineGraph");
    model.result("pg2").feature("lngr1").set("xdata", "expr");
    model.result("pg2").feature("lngr1").set("xdataexpr", "x");
    model.result("pg2").feature("lngr1").selection().all();
    model.result("pg2").feature("lngr1").set("expr", "T");

    model.variable("var2").active(false);
    model.variable().create("var6");
    model.variable("var6").model("comp1");
    model.variable("var6").label("Inverse matrix A");
    model.variable("var6")
         .set("Det_M", "(a*(-h*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))+k*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-
u*v))+g*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))-l*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*w)))+c*(f*(-p*(t*w-
u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))+k*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-q*w))-g*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-
q*y))-l*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-q*w)))+d*(-f*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-u*v))-h*(m*(t*w-u*z)-
n*(s*t-q*z)+r*(s*u-q*w))+g*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))+l*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-
q*v)))+b*(h*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))-k*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))-f*(-p*(t*v-
jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))+l*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v)))+e*(f*(-o*(ii*w-u*y)+n*(ii*v-
jj*y)+p*(jj*w-u*v))+h*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-q*w))-g*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v))-
k*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v))))");
    model.variable("var6")
         .set("A_inv", "(-h*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))+k*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-
u*v))+g*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))-l*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v)))/Det_M");
    model.variable("var6")
         .set("a", "-
((x_CO/Deff_CH4.CO)+(x_CO2/Deff_CH4.CO2)+(x_H2O/Deff_CH4.H2O)+(x_H2/Deff_CH4.H2)+(1/DKeff_CH4))");
    model.variable("var6").set("b", "x_CH4/Deff_CH4.CO");
    model.variable("var6").set("c", "x_CH4/Deff_CH4.CO2");
    model.variable("var6").set("d", "x_CH4/Deff_CH4.H2O");
    model.variable("var6").set("e", "x_CH4/Deff_CH4.H2");
    model.variable("var6").set("f", "x_CO/Deff_CH4.CO");
    model.variable("var6")
         .set("g", "-
((x_CH4/Deff_CH4.CO)+(x_CO2/Deff_CO.CO2)+(x_H2O/Deff_CO.H2O)+(x_H2/Deff_CO.H2)+(1/DKeff_CO))");
    model.variable("var6").set("h", "x_CO/Deff_CO.CO2");
    model.variable("var6").set("k", "x_CO/Deff_CO.H2O");
    model.variable("var6").set("l", "x_CO/Deff_CO.H2");
    model.variable("var6").set("m", "x_CO2/Deff_CH4.CO2");
    model.variable("var6").set("n", "x_CO2/Deff_CO.CO2");
    model.variable("var6")
         .set("o", "-
((x_CH4/Deff_CH4.CO2)+(x_CO/Deff_CO.CO2)+(x_H2O/Deff_CO2.H2O)+(x_H2/Deff_CO2.H2)+(1/DKeff_CO2))");
    model.variable("var6").set("p", "x_CO2/Deff_CO2.H2O");
    model.variable("var6").set("r", "x_CO2/Deff_CO2.H2");
    model.variable("var6").set("s", "x_H2O/Deff_CH4.H2O");
    model.variable("var6").set("w", "x_H2O/Deff_CO.H2O");
    model.variable("var6").set("v", "x_H2O/Deff_CO2.H2O");
    model.variable("var6")
         .set("y", "-
((x_CH4/Deff_CH4.H2O)+(x_CO/Deff_CO.H2O)+(x_CO2/Deff_CO2.H2O)+(x_H2/Deff_H2.H2O)+(1/DKeff_H2O))");
    model.variable("var6").set("z", "x_H2O/Deff_H2.H2O");
    model.variable("var6").set("q", "x_H2/Deff_CH4.H2");
    model.variable("var6").set("u", "x_H2/Deff_CO.H2");
    model.variable("var6").set("jj", "x_H2/Deff_CO2.H2");
    model.variable("var6").set("ii", "x_H2/Deff_H2.H2O");
    model.variable("var6")
         .set("t", "-
((x_CH4/Deff_CH4.H2)+(x_CO/Deff_CO.H2)+(x_CO2/Deff_CO2.H2)+(x_H2/Deff_H2.H2O)+(1/DKeff_H2))");
    model.variable("var6")
         .set("B_inv", "(c*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))-d*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-u*v))-
b*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))+e*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v)))/Det_M");
    model.variable("var6")
         .set("C_inv", "(-c*(-k*(t*w-u*z)+g*(t*y-ii*z)+l*(ii*w-u*y))+d*(-h*(t*w-u*z)+g*(t*v-jj*z)+l*(jj*w-
u*v))+b*(-k*(t*v-jj*z)+h*(t*y-ii*z)+l*(ii*v-jj*y))-e*(-h*(ii*w-u*y)+g*(ii*v-jj*y)+k*(jj*w-u*v)))/Det_M");
    model.variable("var6")
         .set("D_inv", "(c*(-k*(n*t-r*u)+l*(ii*n-p*u)+g*(p*t-ii*r))-d*(-h*(n*t-r*u)+l*(jj*n-o*u)+g*(o*t-
jj*r))+e*(-h*(ii*n-p*u)+k*(jj*n-o*u)+g*(ii*o-jj*p))-b*(h*(p*t-ii*r)-k*(o*t-jj*r)+l*(ii*o-jj*p)))/Det_M");
    model.variable("var6")
         .set("E_inv", "(b*(h*(p*z-r*y)-k*(o*z-r*v)+l*(o*y-p*v))-c*(g*(p*z-r*y)-k*(n*z-r*w)+l*(n*y-
p*w))+d*(g*(o*z-r*v)-h*(n*z-r*w)+l*(n*v-o*w))-e*(g*(o*y-p*v)-h*(n*y-p*w)+k*(n*v-o*w)))/Det_M");
    model.variable("var6")
         .set("F_inv", "(h*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))-k*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))-
f*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))+l*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v)))/Det_M");
    model.variable("var6")
         .set("G_inv", "(-c*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))+d*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-
q*v))+a*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))-e*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v)))/Det_M");
    model.variable("var6")
         .set("H_inv", "(c*(-k*(s*t-q*z)+f*(t*y-ii*z)+l*(ii*s-q*y))-d*(-h*(s*t-q*z)+f*(t*v-jj*z)+l*(jj*s-q*v))-
a*(-k*(t*v-jj*z)+h*(t*y-ii*z)+l*(ii*v-jj*y))+e*(-h*(ii*s-q*y)+f*(ii*v-jj*y)+k*(jj*s-q*v)))/Det_M");
```

```
    model.variable("var6")
          .set("K_inv", "(a*(h*(p*t-ii*r)-k*(o*t-jj*r)+l*(ii*o-jj*p))-c*(f*(p*t-ii*r)-k*(m*t-q*r)+l*(ii*m-
p*q))+d*(f*(o*t-jj*r)-h*(m*t-q*r)+l*(jj*m-o*q))-e*(-h*(ii*m-p*q)+k*(jj*m-o*q)+f*(ii*o-jj*p)))/Det_M");
    model.variable("var6")
          .set("L_inv", "(-a*(h*(p*z-r*y)-k*(o*z-r*v)+l*(o*y-p*v))+c*(f*(p*z-r*y)-k*(m*z-r*s)+l*(m*y-p*s))-
d*(f*(o*z-r*v)-h*(m*z-r*s)+l*(m*v-o*s))+e*(f*(o*y-p*v)-h*(m*y-p*s)+k*(m*v-o*s)))/Det_M");
    model.variable("var6")
          .set("M_inv", "(f*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))+k*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-q*w))-
g*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))-l*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-q*w)))/Det_M");
    model.variable("var6")
          .set("N_inv", "(-a*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))-d*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-
q*w))+b*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))+e*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-q*w)))/Det_M");
    model.variable("var6")
          .set("O_inv", "(a*(-k*(t*w-u*z)+g*(t*y-ii*z)+l*(ii*w-u*y))+d*(f*(t*w-u*z)-g*(s*t-q*z)+l*(s*u-q*w))-
b*(-k*(s*t-q*z)+f*(t*y-ii*z)+l*(ii*s-q*y))-e*(f*(ii*w-u*y)-g*(ii*s-q*y)+k*(s*u-q*w)))/Det_M");
    model.variable("var6")
          .set("P_inv", "(-a*(-k*(n*t-r*u)+l*(ii*n-p*u)+g*(p*t-ii*r))-d*(f*(n*t-r*u)+l*(m*u-n*q)-g*(m*t-
q*r))+e*(f*(ii*n-p*u)+k*(m*u-n*q)-g*(ii*m-p*q))+b*(f*(p*t-ii*r)-k*(m*t-q*r)+l*(ii*m-p*q)))/Det_M");
    model.variable("var6")
          .set("R_inv", "(a*(g*(p*z-r*y)-k*(n*z-r*w)+l*(n*y-p*w))-b*(f*(p*z-r*y)-k*(m*z-r*s)+l*(m*y-
p*s))+d*(f*(n*z-r*w)-g*(m*z-r*s)+l*(m*w-n*s))-e*(f*(n*y-p*w)-g*(m*y-p*s)+k*(m*w-n*s)))/Det_M");
    model.variable("var6")
          .set("S_inv", "(-f*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-u*v))-h*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-
q*w))+g*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))+l*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6")
          .set("W_inv", "(a*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-u*v))+c*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-q*w))-
b*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))-e*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6")
          .set("V_inv", "(-a*(-h*(t*w-u*z)+g*(t*v-jj*z)+l*(jj*w-u*v))-c*(f*(t*w-u*z)-g*(s*t-q*z)+l*(s*u-
q*w))+b*(-h*(s*t-q*z)+f*(t*v-jj*z)+l*(jj*s-q*v))+e*(h*(s*u-q*w)+f*(jj*w-u*v)-g*(jj*s-q*v)))/Det_M");
    model.variable("var6")
          .set("Y_inv", "(a*(-h*(n*t-r*u)+l*(jj*n-o*u)+g*(o*t-jj*r))+c*(f*(n*t-r*u)+l*(m*u-n*q)-g*(m*t-q*r))-
e*(f*(jj*n-o*u)+h*(m*u-n*q)-g*(jj*m-o*q))-b*(f*(o*t-jj*r)-h*(m*t-q*r)+l*(jj*m-o*q)))/Det_M");
    model.variable("var6")
          .set("Z_inv", "(-a*(g*(o*z-r*v)-h*(n*z-r*w)+l*(n*v-o*w))+b*(f*(o*z-r*v)-h*(m*z-r*s)+l*(m*v-o*s))-
c*(f*(n*z-r*w)-g*(m*z-r*s)+l*(m*w-n*s))+e*(f*(n*v-o*w)+h*(m*w-n*s)-g*(m*v-o*s)))/Det_M");
    model.variable("var6")
          .set("Q_inv", "(f*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v))+h*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-
q*w))-g*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v))-k*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6")
          .set("U_inv", "(-a*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v))-c*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-
q*w))+b*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v))+d*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6")
          .set("JJ_inv", "(a*(-h*(ii*w-u*y)+g*(ii*v-jj*y)+k*(jj*w-u*v))+c*(f*(ii*w-u*y)-g*(ii*s-q*y)+k*(s*u-
q*w))-b*(-h*(ii*s-q*y)+f*(ii*v-jj*y)+k*(jj*s-q*v))-d*(h*(s*u-q*w)+f*(jj*w-u*v)-g*(jj*s-q*v)))/Det_M");
    model.variable("var6")
          .set("II_inv", "(-a*(-h*(ii*n-p*u)+k*(jj*n-o*u)+g*(ii*o-jj*p))-c*(f*(ii*n-p*u)+k*(m*u-n*q)-g*(ii*m-
p*q))+d*(f*(jj*n-o*u)+h*(m*u-n*q)-g*(jj*m-o*q))+b*(-h*(ii*m-p*q)+k*(jj*m-o*q)+f*(ii*o-jj*p)))/Det_M");
    model.variable("var6")
          .set("T_inv", "(a*(g*(o*y-p*v)-h*(n*y-p*w)+k*(n*v-o*w))-b*(f*(o*y-p*v)-h*(m*y-p*s)+k*(m*v-
o*s))+c*(f*(n*y-p*w)-g*(m*y-p*s)+k*(m*w-n*s))-d*(f*(n*v-o*w)+h*(m*w-n*s)-g*(m*v-o*s)))/Det_M");
    model.variable("var6")
          .set("a", "-
(1/M_CH4)*((x_CO/Deff_CH4.CO)+(x_CO2/Deff_CH4.CO2)+(x_H2/Deff_CH4.H2)+(x_H2O/Deff_CH4.H2O)+(1/DK_CH4))");
    model.variable("var6").set("b", "x_CH4/(M_CO*Deff_CH4.CO)");
    model.variable("var6").set("c", "x_CH4/(M_CO2*Deff_CH4.CO2)");
    model.variable("var6").set("d", "x_CH4/(M_CO2*Deff_CH4.CO2)");
    model.variable("var6").set("e", "x_CH4/(M_H2O*Deff_CH4.H2O)");
    model.variable("var6").set("d", "x_CH4/(M_H2*Deff_CH4.H2)");
    model.variable("var6").set("f", "x_CO/(M_CH4*Deff_CH4.CO)");
    model.variable("var6")
          .set("g", "-
(1/M_CO)*((x_CH4/Deff_CH4.CO)+(x_CO2/Deff_CO.CO2)+(x_H2/Deff_CO.H2)+(x_H2O/Deff_CO.H2O)+(1/DK_CO))");
    model.variable("var6").set("h", "x_CO/(M_CO2*Deff_CH4.CO2)");
    model.variable("var6").set("k", "x_CO/(M_H2*Deff_CH4.H2)");
    model.variable("var6").set("l", "x_CO/(M_H2O*Deff_CO.H2O)");
    model.variable("var6").set("k", "x_CO/(M_H2*Deff_CO.H2)");
    model.variable("var6").set("h", "x_CO/(M_CO2*Deff_CO.CO2)");
    model.variable().move("var6", 2);
    model.variable("var6").set("m", "x_CO2/(M_CH4*Deff_CH4.CO2)");
    model.variable("var6").set("n", "x_CO2/(M_CO*Deff_CO.CO2)");
    model.variable("var6")
          .set("o", "-
(1/M_CO2)*((x_CH4/Deff_CH4.CO2)+(x_CO/Deff_CO.CO2)+(x_H2/Deff_CO2.H2)+(x_H2O/Deff_CO2.H2O)+(1/DK_CO2))");
    model.variable("var6").set("p", "x_CO2/(M_H2*Deff_CO2.H2)");
    model.variable("var6").set("r", "x_CO2/(M_H2O*Deff_CO2.H2O)");
    model.variable("var6").set("s", "x_H2/(M_CH4*Deff_CH4.H2)");
    model.variable("var6").set("w", "x_H2/(M_CO*Deff_CO.H2)");
    model.variable("var6").set("v", "x_H2/(M_CO2*Deff_CO2.H2)");
    model.variable("var6")
```

```
        .set("y", "-
(1/M_H2)*((x_CH4/Deff_CH4.H2)+(x_CO/Deff_CO.H2)+(x_CO2/Deff_CO2.H2)+(x_H2O/Deff_H2.H2O)+(1/DK_H2))");
    model.variable("var6").set("z", "x_H2/(M_H2O*Deff_H2.H2O)");
    model.variable("var6").set("q", "x_H2O/(M_CH4*Deff_CH4.H2O)");
    model.variable("var6").set("u", "x_H2O/(M_CO*Deff_CO.H2O)");
    model.variable("var6").set("jj", "x_H2O/(M_CO2*Deff_CO2.H2O)");
    model.variable("var6").set("ii", "x_H2O/(M_H2*Deff_H2.H2O)");
    model.variable("var6")
        .set("t", "-
(1/M_H2O)*((x_CH4/Deff_CH4.H2O)+(x_CO/Deff_CO.H2O)+(x_CO2/Deff_CO2.H2O)+(x_H2/Deff_H2.H2O)+(1/DK_H2O))");

    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A_inv", 0);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A_inv", 5);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A_inv", 10);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A_inv", 15);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*E_inv", 20);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*D_inv", 15);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*C_inv", 10);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*B_inv", 5);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*F_inv", 1);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*G_inv", 6);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*H_inv", 11);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*K_inv", 16);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*L_inv", 21);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*M_inv", 2);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*N_inv", 7);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*O_inv", 12);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*P_inv", 17);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*S_inv", 22);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*R_inv", 22);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*S_inv", 3);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*W_inv", 8);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*V_inv", 13);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*Y_inv", 18);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*Z_inv", 23);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*Q_inv", 4);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*U_inv", 9);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*JJ_inv", 14);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*T_inv", 24);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*II_inv", 19);

    model.sol("sol1").clearSolution();
    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.mesh("mesh1").create("dis1", "Distribution");
    model.mesh("mesh1").feature("dis1").selection().all();
    model.mesh("mesh1").feature("dis1").set("type", "predefined");
    model.mesh("mesh1").feature("dis1").set("elemcount", "1000");
    model.mesh("mesh1").feature("dis1").set("elemratio", "20");
    model.mesh("mesh1").feature("dis1").set("reverse", "on");
    model.mesh("mesh1").run();
    model.mesh("mesh1").feature("dis1").set("reverse", "off");
    model.mesh("mesh1").feature("size").set("custom", "on");
    model.mesh("mesh1").feature("dis1").set("type", "number");
    model.mesh("mesh1").run();
    model.mesh("mesh1").feature().remove("dis1");
    model.mesh("mesh1").automatic(true);
    model.mesh("mesh1").automatic(false);
    model.mesh("mesh1").feature("edg1").create("dis1", "Distribution");
    model.mesh("mesh1").feature("edg1").feature("dis1").set("type", "predefined");
```

```
model.mesh("mesh1").feature("edg1").feature("dis1").set("elemcount", "100");
model.mesh("mesh1").feature("edg1").feature("dis1").set("elemratio", "20");
model.mesh("mesh1").run("edg1");
model.mesh("mesh1").feature("edg1").feature("dis1").set("reverse", "on");
model.mesh("mesh1").run("edg1");
model.mesh("mesh1").feature("edg1").feature("dis1").set("elemcount", "1000");
model.mesh("mesh1").run("edg1");
model.mesh("mesh1").run();

model.sol("sol1").clearSolution();
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.physics("c").feature("init1").set("x_CH4", "x_CH4_0");
model.physics("c").feature("init1").set("x_CO", "x_CO_0");
model.physics("c").feature("init1").set("x_CO2", "x_CO2_0");
model.physics("c").feature("init1").set("x_H2", "x_H2_0");
model.physics("c").feature("init1").set("x_H2O", "x_H2O_0");
model.physics("g").feature("dir1").selection().set(new int[]{2});

model.sol("sol1").clearSolution();
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").feature("s1").feature("aDef").set("matherr", "off");
model.sol("sol1").feature("s1").feature("aDef").set("complexfun", "off");
model.sol("sol1").feature("s1").feature("aDef").set("matherr", "on");
model.sol("sol1").clearSolution();
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
```

```
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.variable("var4")
         .set("Dh_CH4", "dh_CH4+(-0.703029*(T/1000)[kJ/(mol*K)]+((108.4773/2)*(T/1000)^2)[kJ/(mol*K^2)]+((-
42.52157/3)*(T/1000)^3)[kJ/(mol*K^3)]+((5.862788/4)*(T/1000)^4)[kJ/(mol*K^4)]+(0.678565/(T/1000))[kJ*K/mol])-
76.84376[kJ/mol]+74.87310 [kJ/mol]");
    model.variable("var4")
         .set("Dh_CO",
"dh_CO+(25.56759*(T/1000)[kJ/(mol*K)]+((6.096130/2)*(T/1000)^2)[kJ/(mol*K^2)]+((4.054656/3)*(T/1000)^3)[kJ/(mol
*K^3)]-((2.671301/4)*(T/1000)^4)[kJ/(mol*K^4)]-(0.310021/(T/1000))[kJ*K/mol])-
118.0089[kJ/mol]+110.5271[kJ/mol]");
    model.variable("var4")
         .set("Dh_CO2", "dh_CO2+(24.99735*(T/1000)[kJ/(mol*K)]+((55.18696/2)*(T/1000)^2)[kJ/(mol*K^2)]-
((33.69137/3)*(T/1000)^3)[kJ/(mol*K^3)]+((7.948387/4)*(T/1000)^4)[kJ/(mol*K^4)]+(0.136638/(T/1000))[kJ*K/mol])-
403.6075[kJ/mol]+393.5224[kJ/mol]");
    model.variable("var4")
         .set("Dh_H2", "dh_H2+(33.066178*(T/1000)[kJ/(mol*K)]-
((11.363417/2)*(T/1000)^2)[kJ/(mol*K^2)]+((11.432816/3)*(T/1000)^3)[kJ/(mol*K^3)]-
((2.772874/4)*(T/1000)^4)[kJ/(mol*K^4)]+(0.158558/(T/1000))[kJ*K/mol])-9.980797[kJ/mol]+0[kJ/mol]");
    model.variable("var4")
         .set("Dh_H2O",
"dh_H2O+(30.092*(T/1000)[kJ/(mol*K)]+((6.832514/2)*(T/1000)^2)[kJ/(mol*K^2)]+((6.793435/3)*(T/1000)^3)[kJ/(mol*
K^3)]-((2.534480/4)*(T/1000)^4)[kJ/(mol*K^4)]-(0.082139/(T/1000))[kJ*K/mol])-
250.8810[kJ/mol]+241.8264[kJ/mol]");
    model.variable("var4").set("dH1", "3*Dh_H2+Dh_CO-Dh_CH4-Dh_H2O");
    model.variable("var4").set("dH2", "Dh_CO2+Dh_H2-Dh_CO-Dh_H2O");
    model.variable("var4").set("dH3", "4*Dh_H2+Dh_CO2-Dh_CH4-2*Dh_H2O");

    model.param().set("dh_CH4", "-74.81[kJ/mol]");
    model.param().set("dh_CO", "-110.525[kJ/mol]");
    model.param().set("dh_CO2", "-393.509[kJ/mol]");
    model.param().set("dh_H2", "0[kJ/mol]");
    model.param().set("dh_H2O", "-285.830[kJ/mol]");

    model.sol("sol1").clearSolution();
    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");

    model.param().set("U_ave", "0.3[m/s]");
    model.param().set("CO.CH4_0", "0.1");
    model.param().set("CO2.CH4_0", "0.056");
    model.param().set("H2.CH4_0", "0.122");
    model.param().set("H2O.CH4_0", "3.358");
    model.param().set("A_pores", "9.3[m^2/g]");

    model.sol("sol1").clearSolution();
    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg1").run();
    model.result("pg1").run();
```

```
model.study("std1").feature("stat").set("useparam", "on");
model.study("std1").feature("stat").set("sweeptype", "sparse");
model.study("std1").feature("stat").setIndex("pname", "Bed_Parameters_Geometry", 0);
model.study("std1").feature("stat").setIndex("plistarr", "", 0);
model.study("std1").feature("stat").setIndex("punit", "", 0);
model.study("std1").feature("stat").setIndex("pname", "Bed_Parameters_Geometry", 0);
model.study("std1").feature("stat").setIndex("plistarr", "", 0);
model.study("std1").feature("stat").setIndex("punit", "", 0);
model.study("std1").feature("stat").setIndex("pname", "T_in", 0);
model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,1200)", 0);
model.study("std1").feature("stat").setIndex("punit", "K", 0);
model.study("std1").feature("stat").set("useadvanceddisable", "off");

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr2", "lngr1");
model.result("pg1").run();
model.result("pg1").feature("lngr2").set("expr", "x_CO");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").create("ptgr1", "PointGraph");
model.result("pg1").feature("ptgr1").set("expr", "x_CO2");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature().remove("ptgr1");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr3", "lngr1");
model.result("pg1").feature().duplicate("lngr4", "lngr2");
model.result("pg1").run();
model.result("pg1").feature("lngr3").set("expr", "x_H2");
model.result("pg1").run();
model.result("pg1").feature("lngr4").set("expr", "x_CO2");
model.result("pg1").feature().move("lngr4", 2);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").setIndex("looplevelinput", "manual", 0);
model.result("pg1").setIndex("looplevel", "1", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "5", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "6", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "7", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "8", 0);
model.result("pg1").run();
model.result("pg2").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr5", "lngr3");
model.result("pg1").run();
model.result("pg1").feature("lngr5").set("expr", "x_H2O");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "3", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "8", 0);
model.result("pg1").run();
model.result("pg2").run();
model.result("pg1").run();
model.result("pg1").label("Molar Fractions");
model.result("pg1").setIndex("looplevel", "2", 0);
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").label("Temperature");
model.result("pg2").run();
model.result("pg1").run();
```

```
    model.result("pg2").run();
    model.result().duplicate("pg3", "pg2");
    model.result("pg3").run();
    model.result("pg3").label("Check Molar Fraction Sum");
    model.result("pg3").run();
    model.result("pg3").feature("lngr1").set("expr", "x_CH4+x_CO+x_CO2+x_H2O++x_H2");
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg2").run();
    model.result("pg2").run();
    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg2").run();
    model.result("pg2").run();
    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg3").run();
    model.result("pg2").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg3").run();

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "2", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "4", 0);
    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg2").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").setIndex("looplevelinput", "manualindices", 0);

    return model;
  }

  public static Model run5(Model model) {
    model.result("pg3").setIndex("looplevelinput", "manual", 0);
    model.result("pg3").setIndex("looplevel", "4", 0);
    model.result("pg3").setIndex("looplevel", "8", 0);
    model.result("pg3").run();

    model.label("New Pellet Model 2016_otherMatrixINV.mph");

    model.result("pg3").run();

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg3").run();
    model.result("pg1").run();

    model.variable("var6").rename("A_inv", "A11_inv");
    model.variable("var6")
         .set("A11_inv", "(-h*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))+k*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-
u*v))+g*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))-l*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v)))/Det_M");
    model.variable("var6").rename("B_inv", "A12_inv");
    model.variable("var6")
         .set("A12_inv", "(c*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))-d*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-
u*v))-b*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))+e*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v)))/Det_M");
    model.variable("var6").rename("C_inv", "A13_inv");
    model.variable("var6")
```

```
        .set("A13_inv", "(-c*(-k*(t*w-u*z)+g*(t*y-ii*z)+l*(ii*w-u*y))+d*(-h*(t*w-u*z)+g*(t*v-jj*z)+l*(jj*w-
u*v))+b*(-k*(t*v-jj*z)+h*(t*y-ii*z)+l*(ii*v-jj*y))-e*(-h*(ii*w-u*y)+g*(ii*v-jj*y)+k*(jj*w-u*v)))/Det_M");
    model.variable("var6").rename("D_inv", "A_14_inv");
    model.variable("var6")
        .set("A_14_inv", "(c*(-k*(n*t-r*u)+l*(ii*n-p*u)+g*(p*t-ii*r))-d*(-h*(n*t-r*u)+l*(jj*n-o*u)+g*(o*t-
jj*r))+e*(-h*(ii*n-p*u)+k*(jj*n-o*u)+g*(ii*o-jj*p))-b*(h*(p*t-ii*r)-k*(o*t-jj*r)+l*(ii*o-jj*p)))/Det_M");
    model.variable("var6").rename("E_inv", "A_15_inv");
    model.variable("var6")
        .set("A_15_inv", "(b*(h*(p*z-r*y)-k*(o*z-r*v)+l*(o*y-p*v))-c*(g*(p*z-r*y)-k*(n*z-r*w)+l*(n*y-
p*w))+d*(g*(o*z-r*v)-h*(n*z-r*w)+l*(n*v-o*w))-e*(g*(o*y-p*v)-h*(n*y-p*w)+k*(n*v-o*w)))/Det_M");
    model.variable("var6").rename("A_14_inv", "A14_inv");
    model.variable("var6")
        .set("A14_inv", "(c*(-k*(n*t-r*u)+l*(ii*n-p*u)+g*(p*t-ii*r))-d*(-h*(n*t-r*u)+l*(jj*n-o*u)+g*(o*t-
jj*r))+e*(-h*(ii*n-p*u)+k*(jj*n-o*u)+g*(ii*o-jj*p))-b*(h*(p*t-ii*r)-k*(o*t-jj*r)+l*(ii*o-jj*p)))/Det_M");
    model.variable("var6").rename("A_15_inv", "A15_inv");
    model.variable("var6")
        .set("A15_inv", "(b*(h*(p*z-r*y)-k*(o*z-r*v)+l*(o*y-p*v))-c*(g*(p*z-r*y)-k*(n*z-r*w)+l*(n*y-
p*w))+d*(g*(o*z-r*v)-h*(n*z-r*w)+l*(n*v-o*w))-e*(g*(o*y-p*v)-h*(n*y-p*w)+k*(n*v-o*w)))/Det_M");
    model.variable("var6").rename("F_inv", "A21_inv");
    model.variable("var6")
        .set("A21_inv", "(h*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))-k*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-
q*v))-f*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))+l*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("G_inv", "A22_inv");
    model.variable("var6")
        .set("A22_inv", "(-c*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))+d*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-
q*v))+a*(-p*(t*v-jj*z)+o*(t*y-ii*z)+r*(ii*v-jj*y))-e*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("H_inv", "A23_inv");
    model.variable("var6")
        .set("A23_inv", "(c*(-k*(s*t-q*z)+f*(t*y-ii*z)+l*(ii*s-q*y))-d*(-h*(s*t-q*z)+f*(t*v-jj*z)+l*(jj*s-
q*v))-a*(-k*(t*v-jj*z)+h*(t*y-ii*z)+l*(ii*v-jj*y))+e*(-h*(ii*s-q*y)+f*(ii*v-jj*y)+k*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("K_inv", "A24_inv");
    model.variable("var6")
        .set("A24_inv", "(a*(h*(p*t-ii*r)-k*(o*t-jj*r)+l*(ii*o-jj*p))-c*(f*(p*t-ii*r)-k*(m*t-q*r)+l*(ii*m-
p*q))+d*(f*(o*t-jj*r)-h*(m*t-q*r)+l*(jj*m-o*q))-e*(-h*(ii*m-p*q)+k*(jj*m-o*q)+f*(ii*o-jj*p)))/Det_M");
    model.variable("var6").rename("L_inv", "A25_inv");
    model.variable("var6")
        .set("A25_inv", "(-a*(h*(p*z-r*y)-k*(o*z-r*v)+l*(o*y-p*v))+c*(f*(p*z-r*y)-k*(m*z-r*s)+l*(m*y-p*s))-
d*(f*(o*z-r*v)-h*(m*z-r*s)+l*(m*v-o*s))+e*(f*(o*y-p*v)-h*(m*y-p*s)+k*(m*v-o*s)))/Det_M");
    model.variable("var6").rename("M_inv", "A31_inv");
    model.variable("var6")
        .set("A31_inv", "(f*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))+k*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-q*w))-
g*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))-l*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-q*w)))/Det_M");
    model.variable("var6").rename("N_inv", "A32_inv");
    model.variable("var6")
        .set("A32_inv", "(-a*(-p*(t*w-u*z)+n*(t*y-ii*z)+r*(ii*w-u*y))-d*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-
q*w))+b*(-p*(s*t-q*z)+m*(t*y-ii*z)+r*(ii*s-q*y))+e*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-q*w)))/Det_M");
    model.variable("var6").rename("O_inv", "A33_inv");
    model.variable("var6")
        .set("A33_inv", "(a*(-k*(t*w-u*z)+g*(t*y-ii*z)+l*(ii*w-u*y))+d*(f*(t*w-u*z)-g*(s*t-q*z)+l*(s*u-q*w))-
b*(-k*(s*t-q*z)+f*(t*y-ii*z)+l*(ii*s-q*y))-e*(f*(ii*w-u*y)-g*(ii*s-q*y)+k*(s*u-q*w)))/Det_M");
    model.variable("var6").rename("P_inv", "A34_inv");
    model.variable("var6")
        .set("A34_inv", "(-a*(-k*(n*t-r*u)+l*(ii*n-p*u)+g*(p*t-ii*r))-d*(f*(n*t-r*u)+l*(m*u-n*q)-g*(m*t-
q*r))+e*(f*(ii*n-p*u)+k*(m*u-n*q)-g*(ii*m-p*q))+b*(f*(p*t-ii*r)-k*(m*t-q*r)+l*(ii*m-p*q)))/Det_M");
    model.variable("var6").rename("R_inv", "A35_inv");
    model.variable("var6")
        .set("A35_inv", "(a*(g*(p*z-r*y)-k*(n*z-r*w)+l*(n*y-p*w))-b*(f*(p*z-r*y)-k*(m*z-r*s)+l*(m*y-
p*s))+d*(f*(n*z-r*w)-g*(m*z-r*s)+l*(m*w-n*s))-e*(f*(n*y-p*w)-g*(m*y-p*s)+k*(m*w-n*s)))/Det_M");
    model.variable("var6").rename("S_inv", "A41_inv");
    model.variable("var6")
        .set("A41_inv", "(-f*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-u*v))-h*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-
q*w))+g*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))+l*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("W_inv", "A42_inv");
    model.variable("var6")
        .set("A42_inv", "(a*(-o*(t*w-u*z)+n*(t*v-jj*z)+r*(jj*w-u*v))+c*(m*(t*w-u*z)-n*(s*t-q*z)+r*(s*u-q*w))-
b*(-o*(s*t-q*z)+m*(t*v-jj*z)+r*(jj*s-q*v))-e*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("V_inv", "A43_inv");
    model.variable("var6")
        .set("A43_inv", "(-a*(-h*(t*w-u*z)+g*(t*v-jj*z)+l*(jj*w-u*v))-c*(f*(t*w-u*z)-g*(s*t-q*z)+l*(s*u-
q*w))+b*(-h*(s*t-q*z)+f*(t*v-jj*z)+l*(jj*s-q*v))+e*(h*(s*u-q*w)+f*(jj*w-u*v)-g*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("Y_inv", "A44_inv");
    model.variable("var6")
        .set("A44_inv", "(a*(-h*(n*t-r*u)+l*(jj*n-o*u)+g*(o*t-jj*r))+c*(f*(n*t-r*u)+l*(m*u-n*q)-g*(m*t-q*r))-
e*(f*(jj*n-o*u)+h*(m*u-n*q)-g*(jj*m-o*q))-b*(f*(o*t-jj*r)-h*(m*t-q*r)+l*(jj*m-o*q)))/Det_M");
    model.variable("var6").rename("Z_inv", "A45_inv");
    model.variable("var6")
        .set("A45_inv", "(-a*(g*(o*z-r*v)-h*(n*z-r*w)+l*(n*v-o*w))+b*(f*(o*z-r*v)-h*(m*z-r*s)+l*(m*v-o*s))-
c*(f*(n*z-r*w)-g*(m*z-r*s)+l*(m*w-n*s))+e*(f*(n*v-o*w)+h*(m*w-n*s)-g*(m*v-o*s)))/Det_M");
    model.variable("var6").rename("Q_inv", "A51_inv");
    model.variable("var6")
```

```
        .set("A51_inv", "(f*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v))+h*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-
q*w))-g*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v))-k*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("U_inv", "A52_inv");
    model.variable("var6")
        .set("A52_inv", "(-a*(-o*(ii*w-u*y)+n*(ii*v-jj*y)+p*(jj*w-u*v))-c*(m*(ii*w-u*y)-n*(ii*s-q*y)+p*(s*u-
q*w))+b*(-o*(ii*s-q*y)+m*(ii*v-jj*y)+p*(jj*s-q*v))+d*(o*(s*u-q*w)+m*(jj*w-u*v)-n*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("JJ_inv", "A53_inv");
    model.variable("var6")
        .set("A53_inv", "(a*(-h*(ii*w-u*y)+g*(ii*v-jj*y)+k*(jj*w-u*v))+c*(f*(ii*w-u*y)-g*(ii*s-q*y)+k*(s*u-
q*w))-b*(-h*(ii*s-q*y)+f*(ii*v-jj*y)+k*(jj*s-q*v))-d*(h*(s*u-q*w)+f*(jj*w-u*v)-g*(jj*s-q*v)))/Det_M");
    model.variable("var6").rename("II_inv", "A54_inv");
    model.variable("var6")
        .set("A54_inv", "(-a*(-h*(ii*n-p*u)+k*(jj*n-o*u)+g*(ii*o-jj*p))-c*(f*(ii*n-p*u)+k*(m*u-n*q)-g*(ii*m-
p*q))+d*(f*(jj*n-o*u)+h*(m*u-n*q)-g*(jj*m-o*q))+b*(-h*(ii*m-p*q)+k*(jj*m-o*q)+f*(ii*o-jj*p)))/Det_M");
    model.variable("var6").rename("T_inv", "A55_inv");
    model.variable("var6")
        .set("A55_inv", "(a*(g*(o*y-p*v)-h*(n*y-p*w)+k*(n*v-o*w))-b*(f*(o*y-p*v)-h*(m*y-p*s)+k*(m*v-
o*s))+c*(f*(n*y-p*w)-g*(m*y-p*s)+k*(m*w-n*s))-d*(f*(n*v-o*w)+h*(m*w-n*s)-g*(m*v-o*s)))/Det_M");

    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A11_inv", 0);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A12_inv", 5);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A13_inv", 10);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A14_inv", 15);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A15_inv", 20);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A21_inv", 1);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A22_inv", 6);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A23_inv", 11);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A24_inv", 16);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A25_inv", 21);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A31_inv", 2);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A32_inv", 7);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A33_inv", 12);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A34_inv", 17);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A35_inv", 22);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A41_inv", 3);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A42_inv", 8);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A43_inv", 13);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A44_inv", 18);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A45_inv", 23);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A51_inv", 4);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A52_inv", 9);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A53_inv", 14);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A54_inv", 19);
    model.physics("c").feature("cfeq1").setIndex("c", "-(x^2)*c_TOT*A55_inv", 24);

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg3").run();
    model.result("pg2").run();
    model.result("pg1").run();
    model.result("pg1").run();

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.physics().create("g2", "GeneralFormPDE", "geom1", new String[][]{{"u"}});

    model.study("std1").feature("stat").activate("g2", true);

    model.physics("g2").field("dimensionless").component(1, "X_CH4");
    model.physics("g2").field("dimensionless").component(2, "X_CO");
    model.physics("g2").field("dimensionless").component(3, "X_CO2");
    model.physics("g2").field("dimensionless").component(4, "X_H2");
    model.physics("g2").field("dimensionless").component(5, "X_H2O");
    model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst", 0);
    model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_CO*M_CO*rho_catalyst", 1);
    model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_CO2*M_CO2*rho_catalyst", 2);
    model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_H2*M_H2*rho_catalyst", 3);
    model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_H2O*M_H2O*rho_catalyst", 4);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*c_TOT*(A11_inv*X_CH4x+A12_inv*X_COx+A13_inv*X_CO2x+A14_inv*X_H2x+A15_inv*X_H2Ox)", 0);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*c_TOT*(A21_inv*X_CH4x+A22_inv*X_COx+A23_inv*X_CO2x+A24_inv*X_H2x+A25_inv*X_H2Ox)", 1);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*c_TOT*(A31_inv*X_CH4x+A32_inv*X_COx+A33_inv*X_CO2x+A34_inv*X_H2x+A35_inv*X_H2Ox)", 2);
```

```
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A41_inv*X_CH4x+A42_inv*X_COx+A43_inv*X_CO2x+A44_inv*X_H2x+A45_inv*X_H2Ox)", 3);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A51_inv*X_CH4x+A52_inv*X_COx+A53_inv*X_CO2x+A54_inv*X_H2x+A55_inv*X_H2Ox)", 4);
    model.physics("g2").feature().create("dir1", "DirichletBoundary", 0);
    model.physics("g2").feature("dir1").setIndex("r", "x_CH4_0", 0);
    model.physics("g2").feature("dir1").setIndex("r", "x_CH4_0", 1);
    model.physics("g2").feature("dir1").setIndex("r", "x_CH4_0", 2);
    model.physics("g2").feature("dir1").setIndex("r", "x_CH4_0", 3);
    model.physics("g2").feature("dir1").setIndex("r", "x_H2O_0", 4);
    model.physics("g2").feature("dir1").setIndex("r", "x_H2_0", 3);
    model.physics("g2").feature("dir1").setIndex("r", "x_CO2_0", 2);
    model.physics("g2").feature("dir1").setIndex("r", "x_CO_0", 1);
    model.physics("g2").feature("init1").set("X_CH4", "x_CH4_0");
    model.physics("g2").feature("init1").set("X_CO", "x_CO_0");
    model.physics("g2").feature("init1").set("X_CO2", "x_CO2_0");
    model.physics("g2").feature("init1").set("X_H2", "x_H2_0");
    model.physics("g2").feature("init1").set("X_H2O", "x_H2O_0");
    model.physics("g2").feature("dir1").selection().set(new int[]{2});
    model.physics().remove("c");
    model.physics("g2").field("dimensionless").component(1, "x_CH4");
    model.physics("g2").field("dimensionless").component(2, "x_CO");
    model.physics("g2").field("dimensionless").component(3, "x_CO2");
    model.physics("g2").field("dimensionless").component(4, "x_H2");
    model.physics("g2").field("dimensionless").component(5, "x_H2O");
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A11_inv*x_CH4x+A12_inv*x_COx+A13_inv*x_CO2x+A14_inv*x_H2x+A15_inv*x_H2Ox)", 0);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A21_inv*x_CH4x+A22_inv*x_COx+A23_inv*x_CO2x+A24_inv*x_H2x+A25_inv*x_H2Ox)", 1);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A31_inv*x_CH4x+A32_inv*x_COx+A33_inv*x_CO2x+A34_inv*x_H2x+A35_inv*x_H2Ox)", 2);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A41_inv*x_CH4x+A42_inv*x_COx+A43_inv*x_CO2x+A44_inv*x_H2x+A45_inv*x_H2Ox)", 3);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*c_TOT*(A51_inv*x_CH4x+A52_inv*x_COx+A53_inv*x_CO2x+A54_inv*x_H2x+A55_inv*x_H2Ox)", 4);
    model.physics().move("g2", 0);

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg3").run();
    model.result("pg2").run();
    model.result("pg1").run();

    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 0);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 1);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 2);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 4);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A21_inv*((c_CH4/T)*Tx+c_CH4x)+A22_inv*((c_CO/T)*Tx+c_COx)+A23_inv*((c_CO2/T)*Tx+c_CO2x)+A24_inv*((c_H2/
T)*Tx+c_H2x)+A25_inv*((c_H2O/T)*Tx+c_H2Ox))", 1);
    model.physics("g2").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A35_inv*((c_H2O/T)*Tx+c_H2Ox))", 2);
    model.physics("g2").feature("gfeq1")
```

```
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A45_inv*((c_H2O/T)*Tx+c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A54_inv*((c_H2/
T)*Tx+c_H2x)+A55_inv*((c_H2O/T)*Tx+c_H2Ox))", 4);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A44_inv*((c_H2/
T)*Tx+c_H2x)+A45_inv*((c_H2O/T)*Tx+c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A33_inv*((c_CO2/T)*Tx+c_CO2x)+A34_inv*((c_H2/
T)*Tx+c_H2x)+A35_inv*((c_H2O/T)*Tx+c_H2Ox))", 2);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A43_inv*((c_CO2/T)*Tx+c_CO2x)+A44_inv*((c_H2/
T)*Tx+c_H2x)+A45_inv*((c_H2O/T)*Tx+c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A52_inv*((c_CO/T)*Tx+c_COx)+A53_inv*((c_CO2/T)*Tx+c_CO2x)+A54_inv*((c_H2/
T)*Tx+c_H2x)+A55_inv*((c_H2O/T)*Tx+c_H2Ox))", 4);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A42_inv*((c_CO/T)*Tx+c_COx)+A43_inv*((c_CO2/T)*Tx+c_CO2x)+A44_inv*((c_H2/
T)*Tx+c_H2x)+A45_inv*((c_H2O/T)*Tx+c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A32_inv*((c_CO/T)*Tx+c_COx)+A33_inv*((c_CO2/T)*Tx+c_CO2x)+A34_inv*((c_H2/
T)*Tx+c_H2x)+A35_inv*((c_H2O/T)*Tx+c_H2Ox))", 2);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A31_inv*((c_CH4/T)*Tx+c_CH4x)+A32_inv*((c_CO/T)*Tx+c_COx)+A33_inv*((c_CO2/T)*Tx+c_CO2x)+A34_inv*((c_H2/
T)*Tx+c_H2x)+A35_inv*((c_H2O/T)*Tx+c_H2Ox))", 2);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A41_inv*((c_CH4/T)*Tx+c_CH4x)+A42_inv*((c_CO/T)*Tx+c_COx)+A43_inv*((c_CO2/T)*Tx+c_CO2x)+A44_inv*((c_H2/
T)*Tx+c_H2x)+A45_inv*((c_H2O/T)*Tx+c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
        .setIndex("Ga",
"(x^2)*(A51_inv*((c_CH4/T)*Tx+c_CH4x)+A52_inv*((c_CO/T)*Tx+c_COx)+A53_inv*((c_CO2/T)*Tx+c_CO2x)+A54_inv*((c_H2/
T)*Tx+c_H2x)+A55_inv*((c_H2O/T)*Tx+c_H2Ox))", 4);
    model.physics().create("g3", "GeneralFormPDE", "geom1", new String[][]{{"u2"}});

    model.study("std1").feature("stat").activate("g3", true);

    model.physics("g3").field("dimensionless").field("c");
    model.physics("g3").field("dimensionless").component(new String[]{"u2", "c2", "c3", "c4", "c5"});

    model.variable("var5").remove("c_CH4");

    model.physics("g3").field("dimensionless").component(1, "c_CH4");
    model.physics("g3").feature("gfeq1").setIndex("f", "x_CH4*p_pellet/(R_const*T)", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 0);
    model.physics("g3").prop("Units").set("DependentVariableQuantity", "concentration");
    model.physics("g3").prop("Units").set("SourceTermQuantity", "concentration");

    model.variable("var5").remove("c_CO");
    model.variable("var5").remove("c_CO2");
    model.variable("var5").remove("c_H2");
    model.variable("var5").remove("c_H2O");

    model.physics("g3").feature("gfeq1").setIndex("f", "x_CO*p_pellet/(R_const*T)", 1);
    model.physics("g3").field("dimensionless").component(2, "c_CO");
    model.physics("g3").field("dimensionless").component(3, "c_CO2");
    model.physics("g3").field("dimensionless").component(4, "c_H2");
    model.physics("g3").field("dimensionless").component(5, "c_H2O");
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 4);
    model.physics("g3").feature("gfeq1").setIndex("f", "x_CO2*p_pellet/(R_const*T)", 2);
    model.physics("g3").feature("gfeq1").setIndex("f", "x_H2*p_pellet/(R_const*T)", 3);
    model.physics("g3").feature("gfeq1").setIndex("f", "x_H2O*p_pellet/(R_const*T)", 4);
    model.physics("g3").feature("init1").set("c_CH4", "c_CH4_0");
    model.physics("g3").feature("init1").set("c_CO", "c_CO_0");
    model.physics("g3").feature("init1").set("c_CO2", "c_CO2_0");
    model.physics("g3").feature("init1").set("c_H2", "c_H2_0");
    model.physics("g3").feature("init1").set("c_H2O", "c_H2O_0");
    model.physics("g3").feature().create("dir1", "DirichletBoundary", 0);
```

```
    model.physics("g3").feature("dir1").selection().set(new int[]{2});
    model.physics("g3").feature("dir1").setIndex("r", "c_H2_0", 0);
    model.physics("g3").feature("dir1").setIndex("r", "c_H2_0", 1);
    model.physics("g3").feature("dir1").setIndex("r", "c_H2_0", 2);
    model.physics("g3").feature("dir1").setIndex("r", "c_H2_0", 3);
    model.physics("g3").feature("dir1").setIndex("r", "c_H2O_0", 4);
    model.physics("g3").feature("dir1").setIndex("r", "c_CO2_0", 2);
    model.physics("g3").feature("dir1").setIndex("r", "c_CO_0", 1);
    model.physics("g3").feature("dir1").setIndex("r", "c_CH4_0", 0);

    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "hnlin");

    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A11_inv*(c_CH4*ln(T)x+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 0);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A11_inv*(c_CH4*log(T)x+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2
/T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 0);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A11_inv*(c_CH4*(log(T))x+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_
H2/T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 0);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A11_inv*((c_CH4/T)*Tx+c_CH4x)+A12_inv*((c_CO/T)*Tx+c_COx)+A13_inv*((c_CO2/T)*Tx+c_CO2x)+A14_inv*((c_H2/
T)*Tx+c_H2x)+A15_inv*((c_H2O/T)*Tx+c_H2Ox))", 0);

    model.param().set("CO.CH4_0", "1.1");
    model.param().set("CO2.CH4_0", "1.056");
    model.param().set("H2.CH4_0", "1.122");

    model.study("std1").feature("stat").set("useparam", "off");

    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "2500");

    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A11_inv*(c_CH4x)+A12_inv*(c_COx)+A13_inv*(c_CO2x)+A14_inv*(c_H2x)+A15_inv*(c_H2Ox))", 0);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A21_inv*(c_CH4x)+A22_inv*(c_COx)+A23_inv*(c_CO2x)+A24_inv*(c_H2x)+A25_inv*(c_H2Ox))", 1);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A31_inv*(c_CH4x)+A32_inv*(c_COx)+A33_inv*(c_CO2x)+A34_inv*(c_H2x)+A35_inv*(c_H2Ox))", 2);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A41_inv*(c_CH4x)+A42_inv*(c_COx)+A43_inv*(c_CO2x)+A44_inv*(c_H2x)+A45_inv*(c_H2Ox))", 3);
    model.physics("g2").feature("gfeq1")
         .setIndex("Ga",
"(x^2)*(A51_inv*(c_CH4x)+A52_inv*(c_COx)+A53_inv*(c_CO2x)+A54_inv*(c_H2x)+A55_inv*(c_H2Ox))", 4);
    model.physics().remove("g3");
    model.physics("g2").field("dimensionless").field("c");
    model.physics("g2").field("dimensionless").component(1, "c_CH4");
    model.physics("g2").field("dimensionless").component(2, "c_CO");
    model.physics("g2").field("dimensionless").component(3, "c_CO2");
    model.physics("g2").field("dimensionless").component(4, "c_H2");
    model.physics("g2").field("dimensionless").component(5, "c_H2O");
    model.physics("g2").prop("Units").set("DependentVariableQuantity", "concentration");
    model.physics("g2").prop("Units").set("SourceTermQuantity", "none");
    model.physics("g2").feature("init1").set("c_CH4", "c_CH4_0");
    model.physics("g2").feature("init1").set("c_CO", "c_CO_0");
    model.physics("g2").feature("init1").set("c_CO2", "c_CO2_0");
    model.physics("g2").feature("init1").set("c_H2", "c_H2_0");
    model.physics("g2").feature("init1").set("c_H2O", "c_H2O_0");
    model.physics("g2").feature("dir1").setIndex("r", "c_CH4_0", 0);
    model.physics("g2").feature("dir1").setIndex("r", "c_CO_0", 1);
    model.physics("g2").feature("dir1").setIndex("r", "c_CO2_0", 2);
    model.physics("g2").feature("dir1").setIndex("r", "c_H2_0", 3);
    model.physics("g2").feature("dir1").setIndex("r", "c_H2O_0", 4);

    model.variable("var5").set("x_CH4", "c_CH4/c_TOT");
    model.variable("var5").set("x_CO", "c_CO/c_TOT");
    model.variable("var5").set("xCO2", "c_CO2/c_TOT");
    model.variable("var5").set("x_H2", "c_H2/c_TOT");
    model.variable("var5").set("x_H2O", "c_H2O/c_TOT");
    model.variable("var5").rename("xCO2", "x_CO2");
    model.variable("var5").set("x_CO2", "c_CO2/c_TOT");

    model.sol("sol1").runAll();
```

```
    model.result("pg1").run();

    model.study("std1").feature("stat").set("useparam", "on");

    model.param().set("CO.CH4_0", "0.1");
    model.param().set("CO2.CH4_0", "0.056");
    model.param().set("H2.CH4_0", "0.122");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevelinput", "manual", 0);
    model.result("pg1").setIndex("looplevel", "3", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "4", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "5", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "6", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "7", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "8", 0);
    model.result("pg1").run();
    model.result("pg3").run();
    model.result("pg2").run();
    model.result("pg1").run();
    model.result().duplicate("pg4", "pg1");
    model.result("pg4").run();
    model.result("pg4").label("Molar Concentrations");
    model.result().move("pg4", 1);
    model.result("pg1").run();
    model.result("pg4").run();
    model.result("pg2").run();
    model.result("pg2").run();
    model.result("pg2").run();
    model.result().duplicate("pg5", "pg2");
    model.result("pg5").run();
    model.result("pg5").label("Pressure");
    model.result("pg5").run();
    model.result("pg5").feature("lngr1").set("expr", "c_TOT*R_const*T");
    model.result("pg5").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevelinput", "manual", 0);
    model.result("pg5").setIndex("looplevel", "2", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "3", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "4", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "8", 0);
    model.result("pg5").run();

    model.label("New Pellet Model 2016_constant-Pressure_constant-Temperature(grad_ci).mph");

    model.result("pg5").run();

    model.physics().create("g3", "GeneralFormPDE", "geom1", new String[][]{{"u"}});

    model.study("std1").feature("stat").activate("g3", true);

    model.physics("g3").prop("Units").set("DependentVariableQuantity", "dimensionless");
    model.physics("g3").field("dimensionless").field("n");
    model.physics("g3").field("dimensionless").component(new String[]{"u", "n2", "n3", "n4", "n5"});
    model.physics("g3").field("dimensionless").component(1, "n_CH4");
    model.physics("g3").field("dimensionless").component(2, "n_H2");
    model.physics("g3").field("dimensionless").component(2, "n_CO");
    model.physics("g3").field("dimensionless").component(3, "n_CO2");
    model.physics("g3").field("dimensionless").component(4, "n_H2");
    model.physics("g3").field("dimensionless").component(5, "n_H2O");
    model.physics("g3").feature("gfeq1")
          .setIndex("Ga",
"(x^2)*(A11_inv*(c_CH4x)+A12_inv*(c_COx)+A13_inv*(c_CO2x)+A14_inv*(c_H2x)+A15_inv*(c_H2Ox))", 0);
    model.physics("g3").feature("gfeq1")
          .setIndex("Ga",
"a(A11_inv*(c_CH4x)+A12_inv*(c_COx)+A13_inv*(c_CO2x)+A14_inv*(c_H2x)+A15_inv*(c_H2Ox))", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "a*n_CH4+b*n", 0);
```

```
    model.variable("var6").rename("a", "A_11");
    model.variable("var6")
         .set("A_11", "-
(1/M_CH4)*((x_CO/Deff_CH4.CO)+(x_CO2/Deff_CH4.CO2)+(x_H2/Deff_CH4.H2)+(x_H2O/Deff_CH4.H2O)+(1/DK_CH4))");
    model.variable("var6").rename("b", "A_12");
    model.variable("var6").set("A_12", "x_CH4/(M_CO*Deff_CH4.CO)");
    model.variable("var6").rename("c", "A_13");
    model.variable("var6").set("A_13", "x_CH4/(M_CO2*Deff_CH4.CO2)");
    model.variable("var6").remove("d");
    model.variable("var6").set("A_14", "x_CH4/(M_H2*Deff_CH4.H2)");
    model.variable("var6").rename("e", "A_15");
    model.variable("var6").set("A_15", "x_CH4/(M_H2O*Deff_CH4.H2O)");
    model.variable("var6").rename("f", "A_21");
    model.variable("var6").set("A_21", "x_CO/(M_CH4*Deff_CH4.CO)");
    model.variable("var6").rename("g", "A_22");
    model.variable("var6")
         .set("A_22", "-
(1/M_CO)*((x_CH4/Deff_CH4.CO)+(x_CO2/Deff_CO.CO2)+(x_H2/Deff_CO.H2)+(x_H2O/Deff_CO.H2O)+(1/DK_CO))");
    model.variable("var6").rename("h", "A_23");
    model.variable("var6").set("A_23", "x_CO/(M_CO2*Deff_CO.CO2)");
    model.variable("var6").rename("k", "A_24");
    model.variable("var6").set("A_24", "x_CO/(M_H2*Deff_CO.H2)");
    model.variable("var6").rename("l", "A_35");
    model.variable("var6").set("A_35", "x_CO/(M_H2O*Deff_CO.H2O)");
    model.variable("var6").rename("A_35", "A_25");
    model.variable("var6").set("A_25", "x_CO/(M_H2O*Deff_CO.H2O)");
    model.variable("var6").rename("m", "A_31");
    model.variable("var6").set("A_31", "x_CO2/(M_CH4*Deff_CH4.CO2)");
    model.variable("var6").rename("n", "A_32");
    model.variable("var6").set("A_32", "x_CO2/(M_CO*Deff_CO.CO2)");
    model.variable("var6").rename("o", "A_33");
    model.variable("var6")
         .set("A_33", "-
(1/M_CO2)*((x_CH4/Deff_CH4.CO2)+(x_CO/Deff_CO.CO2)+(x_H2/Deff_CO2.H2)+(x_H2O/Deff_CO2.H2O)+(1/DK_CO2))");
    model.variable("var6").rename("p", "A_34");
    model.variable("var6").set("A_34", "x_CO2/(M_H2*Deff_CO2.H2)");
    model.variable("var6").rename("r", "A_35");
    model.variable("var6").set("A_35", "x_CO2/(M_H2O*Deff_CO2.H2O)");
    model.variable("var6").rename("s", "A_41");
    model.variable("var6").set("A_41", "x_H2/(M_CH4*Deff_CH4.H2)");
    model.variable("var6").rename("w", "A_42");
    model.variable("var6").set("A_42", "x_H2/(M_CO*Deff_CO.H2)");
    model.variable("var6").rename("v", "A_43");
    model.variable("var6").set("A_43", "x_H2/(M_CO2*Deff_CO2.H2)");
    model.variable("var6").rename("y", "A_44");
    model.variable("var6")
         .set("A_44", "-
(1/M_H2)*((x_CH4/Deff_CH4.H2)+(x_CO/Deff_CO.H2)+(x_CO2/Deff_CO2.H2)+(x_H2O/Deff_H2.H2O)+(1/DK_H2))");
    model.variable("var6").rename("z", "A_45");
    model.variable("var6").set("A_45", "x_H2/(M_H2O*Deff_H2.H2O)");
    model.variable("var6").rename("q", "A_51");
    model.variable("var6").set("A_51", "x_H2O/(M_CH4*Deff_CH4.H2O)");
    model.variable("var6").rename("u", "A_52");
    model.variable("var6").set("A_52", "x_H2O/(M_CO*Deff_CO.H2O)");
    model.variable("var6").rename("jj", "A_53");
    model.variable("var6").set("A_53", "x_H2O/(M_CO2*Deff_CO2.H2O)");
    model.variable("var6").rename("ii", "A_54");
    model.variable("var6").set("A_54", "x_H2O/(M_H2*Deff_H2.H2O)");
    model.variable("var6").rename("t", "A_55");
    model.variable("var6")
         .set("A_55", "-
(1/M_H2O)*((x_CH4/Deff_CH4.H2O)+(x_CO/Deff_CO.H2O)+(x_CO2/Deff_CO2.H2O)+(x_H2/Deff_H2.H2O)+(1/DK_H2O))");
    model.variable("var6").remove("A55_inv");
    model.variable("var6").remove("A11_inv");
    model.variable("var6").remove("A12_inv");
    model.variable("var6").remove("A13_inv");
    model.variable("var6").remove("A14_inv");
    model.variable("var6").remove("A15_inv");
    model.variable("var6").remove("A21_inv");
    model.variable("var6").remove("A22_inv");

    return model;
  }

  public static Model run6(Model model) {
    model.variable("var6").remove("A23_inv");
    model.variable("var6").remove("A24_inv");
    model.variable("var6").remove("A25_inv");
    model.variable("var6").remove("A31_inv");
    model.variable("var6").remove("A32_inv");
    model.variable("var6").remove("A33_inv");
```

```
    model.variable("var6").remove("A34_inv");
    model.variable("var6").remove("A35_inv");
    model.variable("var6").remove("A41_inv");
    model.variable("var6").remove("A42_inv");
    model.variable("var6").remove("A43_inv");
    model.variable("var6").remove("A44_inv");
    model.variable("var6").remove("A45_inv");
    model.variable("var6").remove("A51_inv");
    model.variable("var6").remove("A52_inv");
    model.variable("var6").remove("A53_inv");
    model.variable("var6").remove("A54_inv");
    model.variable("var6").remove("Det_M");

    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
4);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O",
1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O",
2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O",
3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O",
4);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 0);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 1);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O",
1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 2);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O",
2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 3);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O",
3);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O",
4);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 4);
    model.physics("g3").prop("Units").set("SourceTermQuantity", "massflux");
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CH4x", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_COx", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CO2x", 2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_H2x", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_H2Ox", 4);
    model.physics("g2").active(false);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 4);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "00", 0);
    model.physics("g3").prop("Units").set("SourceTermQuantity", "massfraction");
    model.physics("g3").prop("Units").set("DependentVariableQuantity", "massflux");
    model.physics("g3").prop("Units").set("SourceTermQuantity", "none");
    model.physics("g3").prop("Units").set("CustomSourceTermUnit", "kmol/(m^2*m^2)");
    model.physics("g2").active(true);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "n_CH4", 0);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "n_CH4x", 0);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CH4x", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O-c_CH4x", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O-c_CH4x", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O-c_CH4x", 2);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O-c_CH4x", 3);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O-c_H2Ox", 4);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O-c_H2x", 3);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O-c_CO2x", 2);
    model.physics("g3").feature("gfeq1")
```

```
        .setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O-c_COx", 1);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CH4", 0);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CO", 1);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CO2", 2);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_H2", 3);
    model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_H2O", 4);
    model.physics("g3").feature().create("cons1", "Constraint", 0);
    model.physics("g3").feature("cons1").selection().set(new int[]{2});
    model.physics("g3").feature().remove("cons1");
    model.physics("g3").feature().create("dir1", "PairDirichletBoundary", 0);
    model.physics("g3").feature().remove("dir1");
    model.physics("g3").feature().create("flux1", "FluxBoundary", 0);
    model.physics("g3").feature("flux1").setIndex("g", "-c_CH4x", 0);
    model.physics("g3").feature("flux1").setIndex("g", "-c_CH4", 0);
    model.physics("g3").feature("flux1").setIndex("g", "c_CH4", 0);
    model.physics("g3").feature().create("cons1", "Constraint", 0);
    model.physics("g3").feature("cons1").setIndex("R", "c_H2O_0", 0);
    model.physics("g3").feature("cons1").selection().set(new int[]{2});
    model.physics("g3").feature("flux1").active(false);
    model.physics("g3").feature("cons1").setIndex("R", "c_H2O_0", 1);
    model.physics("g3").feature("cons1").setIndex("R", "c_H2O_0", 4);
    model.physics("g3").feature("cons1").setIndex("R", "c_H2_0", 3);
    model.physics("g3").feature("cons1").setIndex("R", "c_CO2_0", 2);
    model.physics("g3").feature("cons1").setIndex("R", "c_CO_0", 1);
    model.physics("g3").feature("cons1").setIndex("R", "c_CH4_0", 0);

    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "4500");

    model.physics("g3").feature("init1").set("n_CH4", "M_CH4*D_CH4.H2O*c_CH4_0/r_pellet");
    model.physics("g3").feature("init1").set("n_H2", "M_H2*D_CH4.H2*c_H2_0/r_pellet");
    model.physics("g3").feature("init1").set("n_H2O", "M_H2O*D_CH4.H2O*c_H2O_0/r_pellet");
    model.physics("g3").feature("init1").set("n_CO2", "M_CO2*D_CH4.CO2*c_CO2_0/r_pellet");
    model.physics("g3").feature("init1").set("n_CO", "M_CO*D_CH4.CO*c_CO_0/r_pellet");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "3", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "4", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "6", 0);
    model.result("pg1").run();
    model.result("pg4").run();
    model.result("pg4").setIndex("looplevel", "4", 0);
    model.result("pg4").setIndex("looplevel", "8", 0);
    model.result("pg4").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "4", 0);
    model.result("pg5").setIndex("looplevel", "7", 0);
    model.result("pg5").run();
    model.result("pg2").run();
    model.result("pg1").run();
    model.result("pg4").run();
    model.result("pg4").setIndex("looplevel", "4", 0);
    model.result("pg4").run();
    model.result("pg4").setIndex("looplevel", "7", 0);
    model.result("pg4").run();
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "3", 0);
    model.result("pg1").setIndex("looplevel", "5", 0);
    model.result("pg1").run();
    model.result("pg3").run();
    model.result("pg5").run();
    model.result("pg3").run();

    model.physics("g3").feature("gfeq1").setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CH4x", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_COx", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CO2", 2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CO", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CH4", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_H2", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_H2O", 4);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O",
1);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O",
2);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O",
3);
```

```
    model.physics("g3").feature("gfeq1").setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O",
4);

    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.7");

    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initmethod", "sol");
    model.study("std1").feature("stat").set("initstudy", "std1");
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "hnlin");

    model.result("pg5").run();
    model.result("pg3").run();
    model.result("pg4").run();
    model.result("pg1").run();

    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O-c_CH4x", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O-c_COx", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O-c_CO2x", 2);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O-c_H2x", 3);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O-c_H2Ox", 4);

    model.result("pg1").run();
    model.result("pg1").set("solnum", new String[]{"6"});
    model.result("pg1").run();

    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.6");

    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initmethod", "init");
    model.study("std1").feature("stat").set("initstudy", "zero");

    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "hnlin");

    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initmethod", "sol");
    model.study("std1").feature("stat").set("initstudy", "std1");

    model.sol().create("sol2");
    model.sol("sol2").study("std1");
    model.sol("sol2").create("copy1", "CopySolution");
    model.sol().remove("sol2");

    model.study("std1").feature("stat").set("initmethod", "init");
    model.study("std1").feature("stat").set("initstudy", "zero");
    model.study("std1").feature("stat").set("useinitsol", "off");

    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");

    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 4);

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.sol("sol1").copySolution("sol2");

    model.result().dataset("dset2").label("Study 1/Solution 1 - Before change n");

    model.sol("sol2").label("Solution 1 - Before change n");

    model.physics("g3").feature("gfeq1").setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O-",
0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CH4", 0);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O",
0);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O",
1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CO", 1);
```

```
    model.physics("g3").feature("gfeq1").setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O",
2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_CO2", 2);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O",
3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_H2", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "c_H2O", 4);
    model.physics("g3").feature("gfeq1").setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O",
4);

    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initmethod", "sol");
    model.study("std1").feature("stat").set("initstudy", "std1");
    model.study("std1").feature("stat").set("initsol", "sol2");
    model.study("std1").feature("stat").set("solnum", "1");

    model.result("pg1").run();
    model.result("pg4").run();
    model.result("pg2").run();
    model.result("pg4").run();
    model.result("pg1").run();
    model.result("pg3").run();
    model.result("pg5").run();

    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O-c_CH4x", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O-c_COx", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O-c_CO2x", 2);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O-c_H2x", 3);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O-c_H2Ox", 4);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 4);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 3);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 2);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 0);

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.sol("sol1").copySolution("sol3");

    model.result().dataset("dset3").label("Study 1/Solution 1 - grad_ci");

    model.sol("sol3").label("Solution 1 - - grad_ci");
    model.sol("sol3").label("Solution 1 - grad_ci");

    model.physics().create("g4", "GeneralFormPDE", "geom1", new String[][]{{"u"}});

    model.study("std1").feature("stat").activate("g4", true);

    model.physics("g4").prop("Units").set("DependentVariableQuantity", "pressure");
    model.physics("g4").prop("Units").set("SourceTermQuantity", "pressure");

    model.variable("var5").remove("p_pellet");

    model.physics("g4").field("dimensionless").field("p");
    model.physics("g4").field("dimensionless").component(1, "p_pellet");
    model.physics("g4").feature("gfeq1").setIndex("Ga", "0", 0);
    model.physics("g4").feature("gfeq1").setIndex("f", "(c_TOT*R_const*T)-p_pellet", 0);
    model.physics("g4").feature().create("dir1", "DirichletBoundary", 0);
    model.physics("g4").feature("dir1").setIndex("r", "p_in", 0);
    model.physics("g4").feature("init1").set("p_pellet", "p_in");
    model.physics("g4").feature("dir1").selection().set(new int[]{2});
    model.physics().move("g", 2);

    model.result("pg2").run();

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg5").run();
    model.result("pg5").run();
    model.result("pg5").feature("lngr1").set("expr", "p_pellet");
    model.result("pg5").run();

    model.sol("sol1").copySolution("sol4");
```

```
    model.result().dataset("dset4").label("Study 1/Solution 1 - grad_ci with pressure");

    model.sol("sol4").label("Solution 1 - grad_ci with pressure");

    model.physics("g3").feature("gfeq1")
         .setIndex("Ga", "-
c_CH4x+x_CH4*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)+((1/p_pellet)+((1/DK_CH4)*(Bo/mu_gas)))*c_CH4*p_pelletx", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("Ga", "c_CH4x-
on*x_CH4*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)+on1*((1/p_pellet)+((1/DK_CH4)*(Bo/mu_gas)))*c_CH4*p_pelletx", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("Ga", "-c_CH4x+x_CH4*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)-
((1/p_pellet)+((1/DK_CH4)*(Bo/mu_gas)))*c_CH4*p_pelletx", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O-
c_CH4x+x_CH4*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)-((1/p_pellet)+((1/DK_CH4)*(Bo/mu_gas)))*c_CH4*p_pelletx", 0);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 1);
    model.physics("g3").feature("gfeq1").setIndex("Ga", "0", 0);

    model.variable("var5").set("Bo", "((2*r_pore)^2)/32");
    model.variable("var5").set("mu_H2O", "3.26*1e-5[Pa*s]");
    model.variable("var5").set("mu_H2", "3.26*1e-5[Pa*s]");
    model.variable("var5").set("mu_CO2", "3.26*1e-5[Pa*s]");
    model.variable("var5").set("mu_CO", "3.26*1e-5[Pa*s]");
    model.variable("var5").set("mu_CH4", "2.53*1e-5[Pa*s]");
    model.variable("var5").set("mu_CO", "3.86*1e-5[Pa*s]");
    model.variable("var5").set("mu_CO2", "3.74*1e-5[Pa*s]");
    model.variable("var5").set("mu_H2", "1.84*1e-5[Pa*s]");
    model.variable("var5")
         .set("mu_gas",
"(x_CH4*mu_CH4*sqrt(M_CH4/1[kg/kmol])+x_CO*mu_CO*sqrt(M_CO/1[kg/kmol])+x_CO2*mu_CO2*sqrt(M_CO2/1[kg/kmol])+x_H2
*mu_H2*sqrt(M_H2/1[kg/kmol])+x_H2O*mu_H2O*sqrt(M_H2O/1[kg/kmol]))/(x_CH4*sqrt(M_CH4/1[kg/kmol])+x_CO*sqrt(M_CO/
1[kg/kmol])+x_CO2*sqrt(M_CO2/1[kg/kmol])+x_H2*sqrt(M_H2/1[kg/kmol])+x_H2O*sqrt(M_H2O/1[kg/kmol]))");
    model.variable("var5").descr("mu_CH4", "Dinamic viscosity of CH4 @873K");
    model.variable("var5").descr("mu_CO", "Dinamic viscosity of CO @873K");
    model.variable("var5").descr("mu_CO2", "Dinamic viscosity of CO2 @873K");
    model.variable("var5").descr("mu_H2", "Dinamic viscosity of H2 @873K");
    model.variable("var5").descr("mu_H2O", "Dinamic viscosity of H2O @873K");
    model.variable("var5").descr("mu_gas", "Mean dinamic viscosity of the mixture @873K");

    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O-
c_COx+x_CO*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)-((1/p_pellet)+((1/DK_CO)*(Bo/mu_gas)))*c_CO*p_pelletx", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O-
c_CO2x+x_CO2*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)-((1/p_pellet)+((1/DK_CO2)*(Bo/mu_gas)))*c_CO2*p_pelletx", 2);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O-
c_H2x+x_H2*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)-((1/p_pellet)+((1/DK_H2)*(Bo/mu_gas)))*c_H2*p_pelletx", 3);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O-
c_H2Ox+x_H2O*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox)-((1/p_pellet)+((1/DK_H2O)*(Bo/mu_gas)))*c_H2O*p_pelletx", 4);

    model.label("New Pellet Model 2016_constant-Pressure_constant-Temperature(grad_ci)-automatic matrix
inv.mph");

    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_11*n_CH4+A_12*n_CO+A_13*n_CO2+A_14*n_H2+A_15*n_H2O-
c_CH4x+((on1)*x_CH4*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox))-
((on2)*x_CH4*((1/p_pellet)+((1/DK_CH4)*(Bo/mu_gas)))*c_CH4*p_pelletx)", 0);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_21*n_CH4+A_22*n_CO+A_23*n_CO2+A_24*n_H2+A_25*n_H2O-
c_COx+((on1)*x_CO*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox))-
((on2)*x_CO*((1/p_pellet)+((1/DK_CO)*(Bo/mu_gas)))*c_CO*p_pelletx)", 1);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_31*n_CH4+A_32*n_CO+A_33*n_CO2+A_34*n_H2+A_35*n_H2O-
c_CO2x+((on1)*x_CO2*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox))-
((on2)*x_CO2*((1/p_pellet)+((1/DK_CO2)*(Bo/mu_gas)))*c_CO2*p_pelletx)", 2);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_41*n_CH4+A_42*n_CO+A_43*n_CO2+A_44*n_H2+A_45*n_H2O-
c_H2x+((on1)*x_H2*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox))-
((on2)*x_H2*((1/p_pellet)+((1/DK_H2)*(Bo/mu_gas)))*c_H2*p_pelletx)", 3);
    model.physics("g3").feature("gfeq1")
         .setIndex("f", "A_51*n_CH4+A_52*n_CO+A_53*n_CO2+A_54*n_H2+A_55*n_H2O-
c_H2Ox+((on1)*x_H2O*(c_CH4x+c_COx+c_CO2x+c_H2x+c_H2Ox))-
((on2)*x_H2O*((1/p_pellet)+((1/DK_H2O)*(Bo/mu_gas)))*c_H2O*p_pelletx)", 4);

    model.param().set("on1", "0");
    model.param().set("on2", "0");
```

```
    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "4", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "8", 0);
    model.result("pg1").run();

    model.sol("sol1").copySolution("sol5");

    model.result().dataset("dset5").label("Study 1/Solution 1 - on1 =on2 = 0");

    model.sol("sol5").label("Solution 1 - on1= on2= 0");

    model.param().set("on1", "1");
    model.param().set("on2", "1");

    model.study("std1").feature("stat").set("initsol", "sol5");
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "3", 0);
    model.result("pg4").run();
    model.result("pg3").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "3", 0);
    model.result("pg5").run();

    model.sol("sol1").copySolution("sol6");

    model.result().dataset("dset6").label("Study 1/Solution 1 - on1= on2= 1");

    model.sol("sol6").label("Solution 1 - on1= on2= 1");

    model.label("New Pellet Model 2016_constant-Pressure_constant-Temperature(grad_ci)-automatic matrix
inv.mph");

    model.study("std1").feature("stat").set("useinitsol", "off");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "1", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "4", 0);
    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "6", 0);
    model.result("pg1").run();

    model.sol().remove("sol6");
    model.sol().remove("sol5");
    model.sol().remove("sol4");
    model.sol().remove("sol3");
    model.sol().remove("sol2");

    model.result("pg1").run();
    model.result("pg4").run();
    model.result("pg1").run();

    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("solnum", "auto");

    model.param().set("on1", "0");
    model.param().set("on2", "0");

    model.study("std1").feature("stat").set("useinitsol", "off");

    model.sol().remove("sol1");
    model.sol().create("sol1");
    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "auto");

    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
```

```
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result().create("pg1", 1);
model.result("pg1").set("data", "dset1");
model.result("pg1").create("lngr1", "LineGraph");
model.result("pg1").feature("lngr1").set("xdata", "expr");
model.result("pg1").feature("lngr1").set("xdataexpr", "x");
model.result("pg1").feature("lngr1").selection().all();
model.result("pg1").feature("lngr1").set("expr", "c_CH4");
model.result().create("pg2", 1);
model.result("pg2").set("data", "dset1");
model.result("pg2").create("lngr1", "LineGraph");
model.result("pg2").feature("lngr1").set("xdata", "expr");
model.result("pg2").feature("lngr1").set("xdataexpr", "x");
model.result("pg2").feature("lngr1").selection().all();
model.result("pg2").feature("lngr1").set("expr", "n_CH4");
model.result().create("pg3", 1);
model.result("pg3").set("data", "dset1");
model.result("pg3").create("lngr1", "LineGraph");
model.result("pg3").feature("lngr1").set("xdata", "expr");
model.result("pg3").feature("lngr1").set("xdataexpr", "x");
model.result("pg3").feature("lngr1").selection().all();
model.result("pg3").feature("lngr1").set("expr", "T");
model.result().create("pg4", 1);
model.result("pg4").set("data", "dset1");
model.result("pg4").create("lngr1", "LineGraph");
model.result("pg4").feature("lngr1").set("xdata", "expr");
model.result("pg4").feature("lngr1").set("xdataexpr", "x");
model.result("pg4").feature("lngr1").selection().all();
model.result("pg4").feature("lngr1").set("expr", "p_pellet");

model.param().set("on1", "1");
model.param().set("on2", "1");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "auto");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").run();
model.result("pg3").run();
model.result("pg4").run();
model.result("pg4").run();
model.result("pg4").run();
```

```
model.result("pg3").run();
model.result("pg3").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg1").run();
model.result("pg1").run();

model.physics("g2").label("Species balance (Concentrations)");
model.physics("g3").label("Momentum balance (species mass fluxes)");
model.physics("g").label("Energy Balance (Temperature)");
model.physics("g3").label("Momentum balance (Species Mass Fluxes, ni)");
model.physics("g2").label("Species balance (Concentrations, ci)");
model.physics("g").label("Energy Balance (Temperature, T)");
model.physics("g4").label("Equation of state, Ideal Gas (Pressure, p)");

model.param().remove("on2");
model.param().remove("on1");

model.variable("var5").set("on1", "1");
model.variable("var5").set("on2", "1");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "auto");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").run();
model.result("pg2").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg4").run();

model.variable("var6").label("Matrix A");
model.variable("var3").set("DK_H2O", "(eps_V/Tau)*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2O))");
model.variable("var3").set("DK_H2", "(eps_V/Tau)*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2))");
model.variable("var3").set("DK_CO2", "(eps_V/Tau)*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO2))");
model.variable("var3").set("DK_CO", "(eps_V/Tau)*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO))");
model.variable("var3").set("DK_CH4", "(eps_V/Tau)*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CH4))");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "auto");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
```

```
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");

  return model;
}

public static Model run7(Model model) {
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.6");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-5");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "hnlin");
  model.sol().remove("sol1");

  model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);

  model.sol().create("sol1");
  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "auto");

  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");

  model.result().create("pg1", 1);
  model.result("pg1").set("data", "dset1");
  model.result("pg1").create("lngr1", "LineGraph");
  model.result("pg1").feature("lngr1").set("xdata", "expr");
  model.result("pg1").feature("lngr1").set("xdataexpr", "x");
  model.result("pg1").feature("lngr1").selection().all();
  model.result("pg1").feature("lngr1").set("expr", "c_CH4");
  model.result().create("pg2", 1);
  model.result("pg2").set("data", "dset1");
  model.result("pg2").create("lngr1", "LineGraph");
  model.result("pg2").feature("lngr1").set("xdata", "expr");
  model.result("pg2").feature("lngr1").set("xdataexpr", "x");
  model.result("pg2").feature("lngr1").selection().all();
  model.result("pg2").feature("lngr1").set("expr", "n_CH4");
  model.result().create("pg3", 1);
  model.result("pg3").set("data", "dset1");
  model.result("pg3").create("lngr1", "LineGraph");
  model.result("pg3").feature("lngr1").set("xdata", "expr");
  model.result("pg3").feature("lngr1").set("xdataexpr", "x");
  model.result("pg3").feature("lngr1").selection().all();
  model.result("pg3").feature("lngr1").set("expr", "T");
  model.result().create("pg4", 1);
  model.result("pg4").set("data", "dset1");
  model.result("pg4").create("lngr1", "LineGraph");
  model.result("pg4").feature("lngr1").set("xdata", "expr");
  model.result("pg4").feature("lngr1").set("xdataexpr", "x");
  model.result("pg4").feature("lngr1").selection().all();
  model.result("pg4").feature("lngr1").set("expr", "p_pellet");

  model.sol().remove("sol1");

  model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,1200)", 0);

  model.sol().create("sol1");
```

```
    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "auto");

    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.result().create("pg1", 1);
    model.result("pg1").set("data", "dset1");
    model.result("pg1").create("lngr1", "LineGraph");
    model.result("pg1").feature("lngr1").set("xdata", "expr");
    model.result("pg1").feature("lngr1").set("xdataexpr", "x");
    model.result("pg1").feature("lngr1").selection().all();
    model.result("pg1").feature("lngr1").set("expr", "c_CH4");
    model.result().create("pg2", 1);
    model.result("pg2").set("data", "dset1");
    model.result("pg2").create("lngr1", "LineGraph");
    model.result("pg2").feature("lngr1").set("xdata", "expr");
    model.result("pg2").feature("lngr1").set("xdataexpr", "x");
    model.result("pg2").feature("lngr1").selection().all();
    model.result("pg2").feature("lngr1").set("expr", "n_CH4");
    model.result().create("pg3", 1);
    model.result("pg3").set("data", "dset1");
    model.result("pg3").create("lngr1", "LineGraph");
    model.result("pg3").feature("lngr1").set("xdata", "expr");
    model.result("pg3").feature("lngr1").set("xdataexpr", "x");
    model.result("pg3").feature("lngr1").selection().all();
    model.result("pg3").feature("lngr1").set("expr", "T");
    model.result().create("pg4", 1);
    model.result("pg4").set("data", "dset1");
    model.result("pg4").create("lngr1", "LineGraph");
    model.result("pg4").feature("lngr1").set("xdata", "expr");
    model.result("pg4").feature("lngr1").set("xdataexpr", "x");
    model.result("pg4").feature("lngr1").selection().all();
    model.result("pg4").feature("lngr1").set("expr", "p_pellet");

    model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.6");
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-5");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");

    model.variable("var3").set("DK_H2O", "((eps_V/Tau)*on3+1*(1-
on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2O))");
    model.variable("var5").set("on3", "0");
    model.variable("var3").set("DK_H2", "((eps_V/Tau)*on3+1*(1-
on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_H2))");
    model.variable("var3").set("DK_CO2", "((eps_V/Tau)*on3+1*(1-
on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO2))");
    model.variable("var3").set("DK_CO", "((eps_V/Tau)*on3+1*(1-
on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CO))");
    model.variable("var3").set("DK_CH4", "((eps_V/Tau)*on3+1*(1-
on3))*(4/3)*(r_pore)*sqrt((R_const*T)/(2*pi*M_CH4))");

    model.sol().remove("sol1");

    model.study("std1").feature("stat").setIndex("pname", "Bed_Parameters_Geometry", 1);
    model.study("std1").feature("stat").setIndex("plistarr", "", 1);
    model.study("std1").feature("stat").setIndex("punit", "", 1);
    model.study("std1").feature("stat").setIndex("pname", "Bed_Parameters_Geometry", 1);
    model.study("std1").feature("stat").setIndex("plistarr", "", 1);
    model.study("std1").feature("stat").setIndex("punit", "", 1);
    model.study("std1").feature("stat").remove("pname", 1);
    model.study("std1").feature("stat").remove("punit", 1);
    model.study("std1").feature("stat").remove("plistarr", 1);
```

```
model.sol().create("sol1");
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "auto");

model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result().create("pg1", 1);
model.result("pg1").set("data", "dset1");
model.result("pg1").create("lngr1", "LineGraph");
model.result("pg1").feature("lngr1").set("xdata", "expr");
model.result("pg1").feature("lngr1").set("xdataexpr", "x");
model.result("pg1").feature("lngr1").selection().all();
model.result("pg1").feature("lngr1").set("expr", "c_CH4");
model.result().create("pg2", 1);
model.result("pg2").set("data", "dset1");
model.result("pg2").create("lngr1", "LineGraph");
model.result("pg2").feature("lngr1").set("xdata", "expr");
model.result("pg2").feature("lngr1").set("xdataexpr", "x");
model.result("pg2").feature("lngr1").selection().all();
model.result("pg2").feature("lngr1").set("expr", "n_CH4");
model.result().create("pg3", 1);
model.result("pg3").set("data", "dset1");
model.result("pg3").create("lngr1", "LineGraph");
model.result("pg3").feature("lngr1").set("xdata", "expr");
model.result("pg3").feature("lngr1").set("xdataexpr", "x");
model.result("pg3").feature("lngr1").selection().all();
model.result("pg3").feature("lngr1").set("expr", "T");
model.result().create("pg4", 1);
model.result("pg4").set("data", "dset1");
model.result("pg4").create("lngr1", "LineGraph");
model.result("pg4").feature("lngr1").set("xdata", "expr");
model.result("pg4").feature("lngr1").set("xdataexpr", "x");
model.result("pg4").feature("lngr1").selection().all();
model.result("pg4").feature("lngr1").set("expr", "p_pellet");

model.sol("sol1").runAll();

model.result("pg1").run();

model.study("std1").feature("stat").set("useinitsol", "on");
model.study("std1").feature("stat").set("solnum", "1");

model.variable("var5").set("on3", "0.2");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
```

```
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").feature("v1").set("solnum", "auto");
model.sol("sol1").feature("v1").set("solvertype", "solnum");
model.sol("sol1").feature("v1").set("listsolnum", new String[]{"1"});
model.sol("sol1").feature("v1").set("solnum", 8);
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();

model.variable("var5").set("on3", "0.4");

model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").feature("v1").set("solnum", "auto");
model.sol("sol1").feature("v1").set("solvertype", "solnum");
model.sol("sol1").feature("v1").set("listsolnum", new String[]{"1"});
model.sol("sol1").feature("v1").set("solnum", 1);
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").attach("std1");

model.variable("var5").set("on3", "0.2");

model.sol("sol1").runAll();

model.result("pg1").run();

model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");
model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.6");
model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-5");

model.variable("var5").set("on3", "0.3");

model.study("std1").feature("stat").set("solnum", "auto");

model.sol("sol1").runAll();

model.result("pg1").run();

model.variable("var5").set("on3", "0.4");

model.sol("sol1").runAll();

model.result("pg1").run();

model.variable("var5").set("on3", "0.4");

model.sol("sol1").runAll();
```

```
model.result("pg1").run();

model.variable("var5").set("on3", "0.45");

model.sol("sol1").runAll();

model.result("pg1").run();

model.variable("var5").set("on3", "0.45");

model.study("std1").feature("stat").set("useinitsol", "off");

model.variable("var5").set("on3", "0");

model.study("std1").feature("stat").set("useinitsol", "off");

model.sol().remove("sol1");
model.sol().create("sol1");
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "auto");

model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result().create("pg1", 1);
model.result("pg1").set("data", "dset1");
model.result("pg1").create("lngr1", "LineGraph");
model.result("pg1").feature("lngr1").set("xdata", "expr");
model.result("pg1").feature("lngr1").set("xdataexpr", "x");
model.result("pg1").feature("lngr1").selection().all();
model.result("pg1").feature("lngr1").set("expr", "c_CH4");
model.result().create("pg2", 1);
model.result("pg2").set("data", "dset1");
model.result("pg2").create("lngr1", "LineGraph");
model.result("pg2").feature("lngr1").set("xdata", "expr");
model.result("pg2").feature("lngr1").set("xdataexpr", "x");
model.result("pg2").feature("lngr1").selection().all();
model.result("pg2").feature("lngr1").set("expr", "n_CH4");
model.result().create("pg3", 1);
model.result("pg3").set("data", "dset1");
model.result("pg3").create("lngr1", "LineGraph");
model.result("pg3").feature("lngr1").set("xdata", "expr");
model.result("pg3").feature("lngr1").set("xdataexpr", "x");
model.result("pg3").feature("lngr1").selection().all();
model.result("pg3").feature("lngr1").set("expr", "T");
model.result().create("pg4", 1);
model.result("pg4").set("data", "dset1");
model.result("pg4").create("lngr1", "LineGraph");
model.result("pg4").feature("lngr1").set("xdata", "expr");
model.result("pg4").feature("lngr1").set("xdataexpr", "x");
model.result("pg4").feature("lngr1").selection().all();
model.result("pg4").feature("lngr1").set("expr", "p_pellet");

model.sol("sol1").runAll();

model.result("pg1").run();

model.sol("sol1").copySolution("sol2");

model.result().dataset("dset2").label("Study 1/Solution 1 - on3=0");

model.sol("sol2").label("Solution 1 - on3=0");
```

```
model.variable("var5").set("on3", "0.45");

model.study("std1").feature("stat").set("useinitsol", "on");
model.study("std1").feature("stat").set("initsol", "sol2");
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();

model.sol("sol1").copySolution("sol3");

model.result().dataset("dset3").label("Study 1/Solution 1 - on3=0.45");

model.sol("sol3").label("Solution 1 - on3=0.45");

model.variable("var5").set("on3", "0.5");

model.study("std1").feature("stat").set("initsol", "sol3");
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();

model.variable("var5").set("on3", "0.8");

model.sol("sol1").study("std1");
```

```
model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();

model.variable("var5").set("on3", "1");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-5");
model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.7");

model.variable("var5").set("on3", "0.9");

model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");
model.sol("sol1").runAll();

model.result("pg1").run();

model.sol("sol1").copySolution("sol4");

model.result().dataset("dset4").label("Study 1/Solution 1 - on3=0.9");

model.sol("sol4").label("Solution 1 - on3=0.9");

model.variable("var5").set("on3", "1");

model.study("std1").feature("stat").set("initsol", "sol4");
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "2500");
model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.6");

model.study("std1").feature("stat").set("solnum", "1");
```

```
model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-6");

model.study("std1").feature("stat").set("solnum", "3");
model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);

model.sol("sol1").runAll();

model.result("pg1").run();

model.study("std1").feature("stat").set("initsol", "current");
model.study("std1").feature("stat").set("solnum", "1");
model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);

model.sol("sol1").runAll();

model.result("pg1").run();

model.study("std1").feature("stat").set("solnum", "1");
model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,1200)", 0);
model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);
model.study("std1").feature("stat").set("initsol", "sol4");
model.study("std1").feature("stat").set("solnum", "2");

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg2").run();
model.result("pg3").run();
model.result("pg4").run();
model.result("pg3").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg1").run();

model.sol("sol1").copySolution("sol5");

model.result().dataset("dset5").label("Study 1/Solution 1 - on3=1 (600K ahead)");

model.sol("sol5").label("Solution 1 - on3=1 (600K ahead)");

model.label("New Pellet Model 2016 - Full version (FINAL).mph");

model.result("pg4").run();
model.result("pg3").run();
model.result("pg2").run();
model.result("pg1").run();

model.variable("var5").set("r_CH4", "-(r1+r3)");

model.result("pg1").run();
model.result("pg1").set("data", "dset5");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").set("data", "dset1");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature("lngr1").selection().all();
model.result("pg1").run();
model.result("pg1").run();

model.study("std1").feature("stat").set("useinitsol", "off");
model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,1200)", 0);
model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg2").run();
model.result("pg3").run();
model.result("pg4").run();
model.result("pg4").label("Pellet pressure");
model.result("pg3").run();
model.result("pg3").label("Pellet Temperature");
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").label("Pellet Total Mass Flux - ni");
```

```
  model.result("pg3").run();
  model.result("pg3").label("Pellet Temperature - T");
  model.result("pg4").run();
  model.result("pg4").label("Pellet pressure - p_pressure");
  model.result("pg1").run();
  model.result("pg1").label("Pellet Concentration - ci");
  model.result("pg2").run();
  model.result("pg1").run();
  model.result("pg1").run();
  model.result("pg1").feature().duplicate("lngr2", "lngr1");
  model.result("pg1").run();
  model.result("pg1").run();
  model.result("pg1").feature("lngr1").set("legend", "on");
  model.result("pg1").feature("lngr1").set("legendmethod", "manual");
  model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_700 K", 0);
  model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_800 K", 1);
  model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_900 K", 2);
  model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_1000 K", 3);
  model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_1100 K", 4);
  model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_1200 K", 5);
  model.result("pg1").run();
  model.result("pg1").feature("lngr2").set("expr", "c_CO");
  model.result("pg1").feature("lngr2").set("legend", "on");
  model.result("pg1").feature("lngr2").set("legendmethod", "manual");
  model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_700 K", 0);
  model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_800 K", 1);
  model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_900 K", 2);
  model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_1000 K", 3);
  model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_1100 K", 4);
  model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_1200 K", 5);
  model.result("pg1").feature("lngr2").set("legendmethod", "automatic");
  model.result("pg1").feature("lngr2").set("legend", "on");
  model.result("pg1").feature("lngr2").set("legendmethod", "automatic");

  model.study("std1").feature("stat").set("useinitsol", "on");
  model.study("std1").feature("stat").set("initsol", "current");
  model.study("std1").feature("stat").set("solnum", "1");
  model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);

  model.sol("sol1").runAll();

  model.result("pg1").run();
  model.result("pg1").setIndex("looplevelinput", "manual", 0);
  model.result("pg1").setIndex("looplevel", "1", 0);
  model.result("pg1").run();

  model.study("std1").feature("stat").setIndex("plistarr", "550, range(600,100,1200)", 0);
  model.study("std1").feature("stat").set("solnum", "1");
  model.study("std1").feature("stat").setIndex("plistarr", "580, range(600,100,1200)", 0);
  model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);
  model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);
  model.study("std1").feature("stat").set("useinitsol", "off");

  model.sol().remove("sol1");
  model.sol().remove("sol2");
  model.sol().remove("sol3");
  model.sol().remove("sol4");

  return model;
}

public static Model run8(Model model) {
  model.sol().remove("sol5");

  model.study("std1").feature("stat").setIndex("plistarr", "range(800,100,1200)", 0);

  model.sol().create("sol1");
  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "auto");

  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
```

```
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.result().create("pg1", 1);
    model.result("pg1").set("data", "dset1");
    model.result("pg1").create("lngr1", "LineGraph");
    model.result("pg1").feature("lngr1").set("xdata", "expr");
    model.result("pg1").feature("lngr1").set("xdataexpr", "x");
    model.result("pg1").feature("lngr1").selection().all();
    model.result("pg1").feature("lngr1").set("expr", "c_CH4");
    model.result().create("pg2", 1);
    model.result("pg2").set("data", "dset1");
    model.result("pg2").create("lngr1", "LineGraph");
    model.result("pg2").feature("lngr1").set("xdata", "expr");
    model.result("pg2").feature("lngr1").set("xdataexpr", "x");
    model.result("pg2").feature("lngr1").selection().all();
    model.result("pg2").feature("lngr1").set("expr", "n_CH4");
    model.result().create("pg3", 1);
    model.result("pg3").set("data", "dset1");
    model.result("pg3").create("lngr1", "LineGraph");
    model.result("pg3").feature("lngr1").set("xdata", "expr");
    model.result("pg3").feature("lngr1").set("xdataexpr", "x");
    model.result("pg3").feature("lngr1").selection().all();
    model.result("pg3").feature("lngr1").set("expr", "T");
    model.result().create("pg4", 1);
    model.result("pg4").set("data", "dset1");
    model.result("pg4").create("lngr1", "LineGraph");
    model.result("pg4").feature("lngr1").set("xdata", "expr");
    model.result("pg4").feature("lngr1").set("xdataexpr", "x");
    model.result("pg4").feature("lngr1").selection().all();
    model.result("pg4").feature("lngr1").set("expr", "p_pellet");

    model.sol("sol1").feature("s1").feature("fc1").set("initstep", "0.6");
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", "1.0E-5");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");

    model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);
    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initmethod", "init");
    model.study("std1").feature("stat").set("initstudy", "zero");
    model.study("std1").feature("stat").set("useinitsol", "off");

    model.variable("var5").set("on3", "0");

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "off", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "on", "g3", "off", "g", "off", "g4", "on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "off", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "off", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "off", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("useinitsol", "on");
```

```
    model.study("std1").feature("stat").set("initmethod", "sol");
    model.study("std1").feature("stat").set("initstudy", "std1");
    model.study("std1").feature("stat").set("solnum", "1");
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "off", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});
    model.study("std1").feature("stat").set("useinitsol", "off");

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "off", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "off", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "on", "g3", "on", "g", "off", "g4", "off"});
    model.study("std1").feature("stat").set("useinitsol", "off");
    model.study("std1").feature("stat").set("usesol", "on");
    model.study("std1").feature("stat").set("notsolmethod", "sol");
    model.study("std1").feature("stat").set("notstudy", "std1");
    model.study("std1").feature("stat").set("notsolnum", "1");

    model.result("pg2").run();
    model.result("pg1").run();
    model.result("pg1").set("innerinput", "manual");
    model.result("pg1").set("solnum", new String[]{"1", "2", "3", "4"});
    model.result("pg1").run();
    model.result("pg1").set("solnum", new String[]{"1"});
    model.result("pg1").run();

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"off"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "off", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});
    model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);
    model.study("std1").feature("stat").set("usesol", "off");
    model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "off", "g4", "on"});

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.variable("var5").set("on3", "1");

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});
    model.study("std1").feature("stat").set("useinitsol", "off");
    model.study("std1").feature("stat").set("usesol", "off");
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "off", "g4", "on"});

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "2", 0);
    model.result("pg1").run();
```

```
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "off", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").setIndex("plistarr", "800", 0);
    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initmethod", "init");
    model.study("std1").feature("stat").set("initstudy", "zero");
    model.study("std1").feature("stat").set("usesol", "on");
    model.study("std1").feature("stat").set("notsolmethod", "init");
    model.study("std1").feature("stat").set("notstudy", "zero");
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "off", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "on", "g3", "off", "g", "off", "g4", "on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "off", "g4", "on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "1", 0);

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "off", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "off", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat")
         .set("activate", new String[]{"g2", "off", "g3", "off", "g", "on", "g4", "on"});

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "off", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("physselection", "g2");
    model.study("std1").feature("stat").set("activate", new String[]{"g2", "on", "g3", "on", "g", "on", "g4",
"on"});
    model.study("std1").feature("stat").set("useinitsol", "off");
    model.study("std1").feature("stat").set("usesol", "off");
    model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);
    model.study("std1").feature("stat").set("useinitsol", "on");
    model.study("std1").feature("stat").set("initstudy", "std1");
    model.study("std1").feature("stat").set("solnum", "2");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "1", 0);

    model.sol("sol1").copySolution("sol2");
    model.sol("sol2").label("Solution 1 - 700K to 1200K");

    model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);
    model.study("std1").feature("stat").set("initsol", "sol2");
    model.study("std1").feature("stat").set("solnum", "1");
```

```
model.study("std1").feature("stat").setIndex("plistarr", "650, range(700,100,1200)", 0);
model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,1200)", 0);
model.study("std1").feature("stat").set("solnum", "auto");
model.study("std1").feature("stat").set("useinitsol", "on");

model.sol("sol1").runAll();

model.result("pg1").run();

model.study("std1").feature("stat").setIndex("plistarr", "range(600,100,1200)", 0);
model.study("std1").feature("stat").set("initmethod", "sol");
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").runAll();

model.result("pg1").run();

model.sol("sol1").copySolution("sol3");

model.result().dataset("dset3").label("Study 1/Solution 1 - 600K to 1200K");
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "1", 0);
model.result("pg1").setIndex("looplevel", "1,2,3,4,5,6,7", 0);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature("lngr1").set("legend", "on");
model.result("pg1").feature("lngr1").set("legendmethod", "manual");
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_600 K", 0);
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_700 K", 1);
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_800 K", 2);
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_900 K", 3);
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_1000 K", 4);
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_1100 K", 5);
model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4_1200 K", 6);
model.result("pg1").run();
model.result("pg1").label("Pellet concentration - c_i");
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr2", "lngr1");
model.result("pg1").run();
model.result("pg1").feature("lngr2").set("expr", "c_CO");
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_600 K", 0);
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_700 K", 1);
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_800 K", 2);
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_900 K", 3);
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_1000 K", 4);
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_1100 K", 5);
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO_1200 K", 6);
model.result("pg1").feature().duplicate("lngr3", "lngr2");
model.result("pg1").run();
model.result("pg1").feature("lngr3").set("expr", "c_CO2");
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_600 K", 0);
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_700 K", 1);
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_800 K", 2);
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_900 K", 3);
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_1000 K", 4);
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_1100 K", 5);
model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2_1200 K", 6);
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr4", "lngr2");
model.result("pg1").run();
model.result("pg1").feature("lngr4").set("expr", "c_H2");
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_600 K", 0);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_700 K", 1);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_800 K", 2);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_900 K", 3);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_1000 K", 4);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_1100 K", 5);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2_1200 K", 6);
model.result("pg1").feature().duplicate("lngr5", "lngr4");
model.result("pg1").run();
model.result("pg1").feature("lngr5").set("expr", "c_H2O");
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_600 K", 0);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_700 K", 1);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_800 K", 2);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_900 K", 3);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_1000 K", 4);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_1100 K", 5);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O_1200 K", 6);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "3", 0);
```

```
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "2", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "1", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "3", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "1", 0);
model.result("pg1").run();
model.result().duplicate("pg5", "pg1");
model.result("pg5").run();
model.result().move("pg5", 1);
model.result("pg5").label("Pellet molar fraction - x_i");
model.result("pg5").setIndex("looplevel", "2", 0);
model.result("pg1").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").feature("lngr1").set("expr", "x_CH4");
model.result("pg5").run();
model.result("pg5").feature("lngr2").set("expr", "x_CO");
model.result("pg5").run();
model.result("pg5").feature("lngr3").set("expr", "x_CO2");
model.result("pg5").run();
model.result("pg5").feature("lngr4").set("expr", "x_H2");
model.result("pg5").run();
model.result("pg5").feature("lngr5").set("expr", "x_H2O");
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "4", 0);
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "7", 0);
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "6", 0);
model.result("pg5").run();

model.variable("var5").set("Mn", "x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O");
model.variable("var5").set("w_CH4", "x_CH4*M_CH4/Mn");
model.variable("var5").set("w_CO", "x_CO*M_CO/Mn");
model.variable("var5").set("w_CO2", "x_CO2*M_CO2/Mn");
model.variable("var5").set("w_H2", "x_H2*M_H2/Mn");
model.variable("var5").set("w_H2O", "x_H2O*M_H2O/Mn");

model.result("pg4").run();
model.result("pg4").run();
model.result("pg4").run();
model.result("pg4").label("Pellet pressure (p_pellet)");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").label("Pellet Temperature (T)");
model.result("pg2").run();
model.result("pg2").label("Pellet");
model.result("pg2").run();
model.result("pg2").run();
model.result("pg5").run();
model.result("pg1").run();
model.result("pg5").run();
model.result().duplicate("pg6", "pg5");
model.result("pg6").run();
model.result().move("pg6", 2);
model.result("pg6").label("Pellet total mass flux (n_i)");
model.result("pg6").setIndex("looplevel", "2", 0);
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("expr", "n_CH4");
model.result("pg6").run();
model.result("pg6").feature("lngr2").set("expr", "n_CO");
model.result("pg6").run();
model.result("pg6").feature("lngr3").set("expr", "n_CO2");
model.result("pg6").run();
model.result("pg6").feature("lngr4").set("expr", "n_H2");
model.result("pg6").run();
model.result("pg6").feature("lngr5").set("expr", "n_H2O");
model.result("pg6").run();
model.result("pg6").run();

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg5").run();
model.result("pg6").run();
```

```
model.result("pg2").run();
model.result("pg3").run();
model.result("pg4").run();
model.result("pg5").run();
model.result().duplicate("pg7", "pg5");
model.result("pg7").run();
model.result().move("pg7", 2);
model.result("pg2").run();
model.result().remove("pg2");
model.result("pg6").run();
model.result("pg7").run();
model.result("pg7").label("Pellet mass fraction (w_i)");
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg5").run();
model.result("pg5").label("Pellet molar fraction (x_i)");
model.result("pg1").run();
model.result("pg1").label("Pellet concentration (c_i)");
model.result("pg1").setIndex("looplevel", "3", 0);
model.result("pg5").run();
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").feature("lngr1").set("expr", "w_CH4");
model.result("pg7").run();
model.result("pg7").feature("lngr2").set("expr", "w_CO");
model.result("pg7").run();
model.result("pg7").feature("lngr3").set("expr", "w_CO2");
model.result("pg7").run();
model.result("pg7").feature("lngr4").set("expr", "w_H2");
model.result("pg7").run();
model.result("pg7").feature("lngr5").set("expr", "w_H2O");
model.result("pg7").run();
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_600 K", 0);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_700 K", 1);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_800 K", 2);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_900 K", 3);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_1000 K", 4);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_1100 K", 5);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O_1200 K", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_600 K", 0);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_700 K", 1);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_800 K", 2);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_900 K", 3);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_1000 K", 4);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_1100 K", 5);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2_1200 K", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_600 K", 0);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_700 K", 1);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_800 K", 2);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_900 K", 3);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_1000 K", 4);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_1100 K", 5);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2_1200 K", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_600 K", 0);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_700 K", 1);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_800 K", 2);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_900 K", 3);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_1000 K", 4);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_1100 K", 5);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO_1200 K", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_600 K", 0);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_700 K", 1);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_800 K", 2);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_900 K", 3);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_1000 K", 4);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_1100 K", 5);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4_1200 K", 6);
model.result("pg7").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_600 K", 0);
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_700 K", 1);
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_800 K", 2);
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_900 K", 3);
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_1000 K", 4);
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_1100 K", 5);
model.result("pg5").feature("lngr1").setIndex("legends", "x_CH4_1200 K", 6);
model.result("pg5").run();
```

```
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_600 K", 0);
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_700 K", 1);
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_800 K", 2);
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_900 K", 3);
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_1000 K", 4);
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_1100 K", 5);
model.result("pg5").feature("lngr2").setIndex("legends", "x_CO_1200 K", 6);
model.result("pg5").run();
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_600 K", 0);
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_700 K", 1);
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_800 K", 2);
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_900 K", 3);
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_1000 K", 4);
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_1100 K", 5);
model.result("pg5").feature("lngr3").setIndex("legends", "x_CO2_1200 K", 6);
model.result("pg5").run();
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_600 K", 0);
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_700 K", 1);
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_800 K", 2);
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_900 K", 3);
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_1000 K", 4);
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_1100 K", 5);
model.result("pg5").feature("lngr4").setIndex("legends", "x_H2_1200 K", 6);
model.result("pg5").run();
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_600 K", 0);
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_700 K", 1);
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_800 K", 2);
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_900 K", 3);
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_1000 K", 4);
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_1100 K", 5);
model.result("pg5").feature("lngr5").setIndex("legends", "x_H2O_1200 K", 6);
model.result("pg5").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_600 K", 0);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_700 K", 1);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_800 K", 2);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_900 K", 3);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_1000 K", 4);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_1100 K", 5);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4_1200 K", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_600 K", 0);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_700 K", 1);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_800 K", 2);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_900 K", 3);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_1000 K", 4);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_1100 K", 5);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO_1200 K", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_600 K", 0);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_700 K", 1);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_800 K", 2);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_900 K", 3);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_1000 K", 4);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_1100 K", 5);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2_1200 K", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_600 K", 0);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_700 K", 1);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_800 K", 2);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_900 K", 3);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_1000 K", 4);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_1100 K", 5);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2_1200 K", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_600 K", 0);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_700 K", 1);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_800 K", 2);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_900 K", 3);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_1000 K", 4);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_1100 K", 5);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O_1200 K", 6);
model.result("pg6").run();
model.result("pg3").run();
model.result("pg4").run();
model.result().export().create("data1", "dset3", "Data");
model.result().export("data1").set("expr", new String[]{"c_CH4"});
model.result().export("data1").set("descr", new String[]{"Dependent variable c_CH4"});
```

```
  return model;
}

public static Model run9(Model model) {
  model.result().export("data1").setIndex("expr", "c_CO", 1);
  model.result().export("data1").setIndex("expr", "c_CO2", 2);
  model.result().export("data1").setIndex("expr", "c_H2", 3);
  model.result().export("data1").setIndex("expr", "c_H2O", 4);
  model.result().export("data1").setIndex("expr", "w_CH4", 5);
  model.result().export("data1").setIndex("expr", "w_CO", 6);
  model.result().export("data1").setIndex("expr", "w_CO2", 7);
  model.result().export("data1").setIndex("expr", "w_H2", 8);
  model.result().export("data1").setIndex("expr", "w_H2O", 9);
  model.result().export("data1").setIndex("descr", "Mass_fraction_CH4", 5);
  model.result().export("data1").setIndex("descr", "Mass_fraction_CO", 6);
  model.result().export("data1").setIndex("descr", "Mass_fraction_CO2", 7);
  model.result().export("data1").setIndex("descr", "Mass_fraction_H2", 8);
  model.result().export("data1").setIndex("descr", "Mass_fraction_H2O", 9);
  model.result().export("data1").setIndex("expr", "x_CH4", 10);
  model.result().export("data1").setIndex("unit", "1", 5);
  model.result().export("data1").setIndex("unit", "1", 6);
  model.result().export("data1").setIndex("unit", "1", 7);
  model.result().export("data1").setIndex("unit", "1", 8);
  model.result().export("data1").setIndex("unit", "1", 9);
  model.result().export("data1").setIndex("descr", "Molar_fraction_H2", 10);
  model.result().export("data1").setIndex("expr", "x_CO", 11);
  model.result().export("data1").setIndex("unit", "1", 5);
  model.result().export("data1").setIndex("unit", "1", 6);
  model.result().export("data1").setIndex("unit", "1", 7);
  model.result().export("data1").setIndex("unit", "1", 8);
  model.result().export("data1").setIndex("unit", "1", 9);
  model.result().export("data1").setIndex("descr", "Molar_fraction_CO", 11);
  model.result().export("data1").setIndex("descr", "Molar_fraction_CH4", 10);
  model.result().export("data1").setIndex("descr", "Molar_fraction_CO2", 12);
  model.result().export("data1").setIndex("expr", "x_CO2", 12);
  model.result().export("data1").setIndex("expr", "x_H2", 13);
  model.result().export("data1").setIndex("expr", "x_H2O", 14);
  model.result().export("data1").setIndex("descr", "Molar_fraction_H2", 13);
  model.result().export("data1").setIndex("descr", "Molar_fraction_H2O", 14);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 10, 11, 9);
  model.result().export("data1").move("unit", 10, 11, 9);
  model.result().export("data1").move("descr", 10, 11, 9);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 9, 10, 8);
  model.result().export("data1").move("unit", 9, 10, 8);
  model.result().export("data1").move("descr", 9, 10, 8);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 8, 9, 7);
  model.result().export("data1").move("unit", 8, 9, 7);
  model.result().export("data1").move("descr", 8, 9, 7);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 7, 8, 6);
  model.result().export("data1").move("unit", 7, 8, 6);
  model.result().export("data1").move("descr", 7, 8, 6);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 6, 7, 5);
  model.result().export("data1").move("unit", 6, 7, 5);
  model.result().export("data1").move("descr", 6, 7, 5);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 11, 12, 10);
  model.result().export("data1").move("unit", 11, 12, 10);
  model.result().export("data1").move("descr", 11, 12, 10);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 10, 11, 9);
  model.result().export("data1").move("unit", 10, 11, 9);
  model.result().export("data1").move("descr", 10, 11, 9);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 9, 10, 8);
  model.result().export("data1").move("unit", 9, 10, 8);
  model.result().export("data1").move("descr", 9, 10, 8);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 8, 9, 7);
  model.result().export("data1").move("unit", 8, 9, 7);
  model.result().export("data1").move("descr", 8, 9, 7);
  model.result().export("data1").set("allowdescrupdate", false);
  model.result().export("data1").move("expr", 7, 8, 6);
  model.result().export("data1").move("unit", 7, 8, 6);
  model.result().export("data1").move("descr", 7, 8, 6);
  model.result().export("data1").set("allowdescrupdate", false);
```

```
model.result().export("data1").move("expr", 12, 13, 11);
model.result().export("data1").move("unit", 12, 13, 11);
model.result().export("data1").move("descr", 12, 13, 11);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 11, 12, 10);
model.result().export("data1").move("unit", 11, 12, 10);
model.result().export("data1").move("descr", 11, 12, 10);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 10, 11, 9);
model.result().export("data1").move("unit", 10, 11, 9);
model.result().export("data1").move("descr", 10, 11, 9);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 9, 10, 8);
model.result().export("data1").move("unit", 9, 10, 8);
model.result().export("data1").move("descr", 9, 10, 8);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 8, 9, 7);
model.result().export("data1").move("unit", 8, 9, 7);
model.result().export("data1").move("descr", 8, 9, 7);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 13, 14, 12);
model.result().export("data1").move("unit", 13, 14, 12);
model.result().export("data1").move("descr", 13, 14, 12);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 12, 13, 11);
model.result().export("data1").move("unit", 12, 13, 11);
model.result().export("data1").move("descr", 12, 13, 11);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 11, 12, 10);
model.result().export("data1").move("unit", 11, 12, 10);
model.result().export("data1").move("descr", 11, 12, 10);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 10, 11, 9);
model.result().export("data1").move("unit", 10, 11, 9);
model.result().export("data1").move("descr", 10, 11, 9);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 9, 10, 8);
model.result().export("data1").move("unit", 9, 10, 8);
model.result().export("data1").move("descr", 9, 10, 8);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 14, 15, 13);
model.result().export("data1").move("unit", 14, 15, 13);
model.result().export("data1").move("descr", 14, 15, 13);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 13, 14, 12);
model.result().export("data1").move("unit", 13, 14, 12);
model.result().export("data1").move("descr", 13, 14, 12);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 12, 13, 11);
model.result().export("data1").move("unit", 12, 13, 11);
model.result().export("data1").move("descr", 12, 13, 11);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 11, 12, 10);
model.result().export("data1").move("unit", 11, 12, 10);
model.result().export("data1").move("descr", 11, 12, 10);
model.result().export("data1").set("allowdescrupdate", false);
model.result().export("data1").move("expr", 10, 11, 9);
model.result().export("data1").move("unit", 10, 11, 9);
model.result().export("data1").move("descr", 10, 11, 9);
model.result().export("data1").set("allowdescrupdate", true);
model.result().export("data1").set("alwaysask", "on");
model.result().export("data1").set("header", "off");
model.result().export("data1")
     .set("filename", "C:\\Users\\Flavio\\Documents\\COMSOL Models\\Flavio\\Full Pellet 2016 SOLVED.txt");
model.result("pg7").run();
model.result("pg7").run();
model.result().export("data1").set("solnum", new String[]{"3"});
model.result().export("data1").remove("descr", 14);
model.result().export("data1").remove("unit", 14);
model.result().export("data1").remove("expr", 14);
model.result().export("data1").remove("descr", 10);
model.result().export("data1").remove("unit", 10);
model.result().export("data1").remove("expr", 10);
model.result().export("data1").remove("descr", 10);
model.result().export("data1").remove("unit", 10);
model.result().export("data1").remove("expr", 10);
model.result().export("data1").remove("descr", 10);
model.result().export("data1").remove("unit", 10);
model.result().export("data1").remove("expr", 10);
model.result().export("data1").remove("descr", 10);
model.result().export("data1").remove("unit", 10);
```

```
model.result().export("data1").remove("expr", 10);

model.variable().remove("var2");

model.result().export("data1").set("solnum", new String[]{"1", "2", "3", "4", "5", "6", "7"});
model.result().export("data1")
     .set("filename", "C:\\Users\\Flavio\\Documents\\COMSOL Models\\Flavio\\Full Pellet 2016 SOLVED.txt");
model.result().export("data1").run();
model.result().export("data1").set("header", "on");
model.result().export("data1")
     .set("filename", "C:\\Users\\Flavio\\Documents\\COMSOL Models\\Flavio\\Full Pellet 2016 SOLVED.txt");
model.result().export("data1").run();

model.label("New Pellet Model 2016 - Full version (FINAL).mph");

model.physics("g3").feature("cons1").active(false);

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg6").run();

model.physics("g3").feature("cons1").active(true);
model.physics("g3").feature("cons1").active(false);
model.physics("g3").feature().remove("cons1");
model.physics("g3").feature().remove("flux1");
model.physics("g3").feature().create("dir1", "DirichletBoundary", 0);
model.physics("g2").feature("gfeq1").setIndex("Ga", "eps_A*(x^2)*n_CH4", 0);
model.physics("g2").feature("gfeq1").setIndex("Ga", "eps_A*(x^2)*n_CO", 1);
model.physics("g2").feature("gfeq1").setIndex("Ga", "eps_A*(x^2)*n_CO2", 2);
model.physics("g2").feature("gfeq1").setIndex("Ga", "eps_A*(x^2)*n_H2", 3);
model.physics("g2").feature("gfeq1").setIndex("Ga", "eps_A*(x^2)*n_H2O", 4);
model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_CH4*M_CH4*rho_catalyst*(1-eps_V)", 0);
model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_CO*M_CO*rho_catalyst*(1-eps_V)", 1);
model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_CO2*M_CO2*rho_catalyst*(1-eps_V)", 2);
model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_H2*M_H2*rho_catalyst*(1-eps_V)", 3);
model.physics("g2").feature("gfeq1").setIndex("f", "(x^2)*r_H2O*M_H2O*rho_catalyst*(1-eps_V)", 4);
model.physics("g3").feature("dir1").setIndex("r", "V_pellet*r_CH4*M_CH4*rho_catalyst*(1-eps_V)", 0);
model.physics("g3").feature("dir1").setIndex("r", "Vm_pellet*r_CH4*M_CH4*rho_catalyst*(1-eps_V)", 0);
model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_CH4*M_CH4*rho_catalyst*(1-eps_V)", 0);
model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_CO*M_CO*rho_catalyst*(1-eps_V)", 1);
model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_CO2*M_CO2*rho_catalyst*(1-eps_V)", 2);
model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_H2*M_H2*rho_catalyst*(1-eps_V)", 3);
model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_H2O*M_H2O*rho_catalyst*(1-eps_V)", 4);

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "6", 0);
model.result("pg1").run();
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "4", 0);
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "6", 0);
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "7", 0);
model.result("pg5").run();

model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CH4", 0);
model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CO", 1);
model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_CO2", 2);
model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_H2", 3);
model.physics("g2").feature("gfeq1").setIndex("Ga", "(x^2)*n_H2O", 4);
model.physics("g3").feature("dir1").setIndex("r", "eps_A*r_pellet*r_CH4*M_CH4*rho_catalyst*(1-eps_V)", 0);
model.physics("g3").feature("dir1").setIndex("r", "eps_A*r_pellet*r_CO*M_CO*rho_catalyst*(1-eps_V)", 1);
model.physics("g3").feature("dir1").setIndex("r", "eps_A*r_pellet*r_CO2*M_CO2*rho_catalyst*(1-eps_V)", 2);
model.physics("g3").feature("dir1").setIndex("r", "eps_A*r_pellet*r_H2*M_H2*rho_catalyst*(1-eps_V)", 3);
model.physics("g3").feature("dir1").setIndex("r", "eps_A*r_pellet*r_H2O*M_H2O*rho_catalyst*(1-eps_V)", 4);

model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg5").run();
model.result("pg5").setIndex("looplevel", "6", 0);
model.result("pg5").run();
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").setIndex("looplevel", "1", 0);
model.result("pg7").setIndex("looplevel", "7", 0);
```

```
    model.result("pg7").run();
    model.result("pg6").run();
    model.result("pg6").setIndex("looplevel", "7", 0);
    model.result("pg6").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg4").run();
    model.result("pg1").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "1", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "2", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "3", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "4", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "5", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "4", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "5", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "6", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "7", 0);
    model.result("pg5").run();

    model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_CH4*M_CH4*rho_catalyst*(1-eps_V)", 0);
    model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_CO*M_CO*rho_catalyst*(1-eps_V)", 1);
    model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_CO2*M_CO2*rho_catalyst*(1-eps_V)", 2);
    model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_H2*M_H2*rho_catalyst*(1-eps_V)", 3);
    model.physics("g3").feature("dir1").setIndex("r", "r_pellet*r_H2O*M_H2O*rho_catalyst*(1-eps_V)", 4);

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.physics("g3").feature("dir1").setIndex("r", "(4*pi/3)*r_pellet*r_CH4*M_CH4*rho_catalyst*(1-eps_V)",
0);
    model.physics("g3").feature("dir1").setIndex("r", "(4*pi/3)*r_pellet*r_CO*M_CO*rho_catalyst*(1-eps_V)", 1);
    model.physics("g3").feature("dir1").setIndex("r", "(4*pi/3)*r_pellet*r_CO2*M_CO2*rho_catalyst*(1-eps_V)",
2);
    model.physics("g3").feature("dir1").setIndex("r", "(4*pi/3)*r_pellet*r_H2*M_H2*rho_catalyst*(1-eps_V)", 3);
    model.physics("g3").feature("dir1").setIndex("r", "(4*pi/3)*r_pellet*r_H2O*M_H2O*rho_catalyst*(1-eps_V)",
4);

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "4", 0);
    model.result("pg5").setIndex("looplevel", "5", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "7", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "6", 0);
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "5", 0);
    model.result("pg3").run();

    model.label("New Pellet Model 2016 - Full version (FINAL)2.mph");

    model.result("pg1").run();
    model.result("pg1").set("xlabelactive", "on");
    model.result("pg1").set("xlabel", "r-coordinate (m)");
    model.result("pg1").set("ylabelactive", "on");
    model.result("pg1").set("ylabel", "concentration (mol/cm3)");
    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4", 0);
    model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4", 4);
    model.result("pg1").feature("lngr1").setIndex("legends", "c_CH4", 6);
    model.result("pg1").run();
    model.result("pg1").feature("lngr2").setIndex("legends", "c_CO", 4);
    model.result("pg1").feature("lngr2").setIndex("legends", "c_CO", 6);
    model.result("pg1").run();
    model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2", 4);
    model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2", 6);
    model.result("pg1").feature("lngr3").setIndex("legends", "c_CO2", 0);
    model.result("pg1").run();
```

```
model.result("pg1").feature("lngr2").setIndex("legends", "c_CO", 0);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2", 0);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2", 4);
model.result("pg1").feature("lngr4").setIndex("legends", "c_H2", 6);
model.result("pg1").run();
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O", 0);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O", 4);
model.result("pg1").feature("lngr5").setIndex("legends", "c_H2O", 6);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "5", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "7", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "5", 0);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").set("titletype", "manual");
model.result("pg1").set("ylabel", "concentration (mol/cm<sup>3</sup>)");
model.result("pg1").set("title", "Species concentration x Pellet radius  [T_bulk=1000]");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").set("legendpos", "upperleft");
model.result("pg1").setIndex("looplevel", "7", 0);
model.result("pg1").set("title", "Species concentration x Pellet radius  [T_bulk=1200]");
model.result("pg1").run();
model.result("pg7").run();
model.result("pg7").set("xlabelactive", "on");
model.result("pg7").set("xlabel", "r-coordinate (m)");
model.result("pg7").set("ylabelactive", "on");
model.result("pg7").set("ylabel", "mass fraction (dim)");
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4", 0);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4", 4);
model.result("pg7").feature("lngr1").setIndex("legends", "w_CH4", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO", 0);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO", 4);
model.result("pg7").feature("lngr2").setIndex("legends", "w_CO", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2", 0);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2", 4);
model.result("pg7").feature("lngr3").setIndex("legends", "w_CO2", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2", 0);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2", 4);
model.result("pg7").feature("lngr4").setIndex("legends", "w_H2", 6);
model.result("pg7").run();
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O", 0);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O", 4);
model.result("pg7").feature("lngr5").setIndex("legends", "w_H2O", 6);
model.result("pg7").run();
model.result("pg7").set("titletype", "manual");
model.result("pg7").set("title", "Mass fractions x Pellet radius  [T_bulk=1000K]");
model.result("pg7").set("legendpos", "upperleft");
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").setIndex("looplevel", "5", 0);
model.result("pg7").run();
model.result("pg7").set("title", "Species mass fractions x Pellet radius  [T_bulk=1000K]");
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "7", 0);
model.result("pg7").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "1", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "3", 0);
model.result("pg6").setIndex("looplevel", "4", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").run();
```

```
model.result("pg6").setIndex("looplevel", "7", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "4", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").run();
model.result("pg6").set("xlabelactive", "on");
model.result("pg6").set("xlabel", "r-coordinate (m)");
model.result("pg6").set("ylabelactive", "on");
model.result("pg6").set("ylabel", "Mass flux in r direction (g/(cm2.s))");
model.result("pg6").run();
model.result("pg7").run();
model.result("pg5").run();
model.result("pg1").run();
model.result("pg7").run();
model.result("pg6").run();
model.result("pg6").set("ylabel", "Mass flux in r direction (g/(/cm<sup>2</sup>.s))");
model.result("pg6").setIndex("looplevel", "3", 0);
model.result("pg6").set("ylabel", "Mass flux in r direction (g/(cm<sup>2</sup>.s))");
model.result("pg6").run();
model.result("pg6").set("titletype", "manual");
model.result("pg6").set("title", "Mass flux in the r direction x Pellet radius [T_bulk=1000]");
model.result("pg6").run();
model.result("pg1").run();
model.result("pg6").run();
model.result("pg6").set("legendpos", "upperleft");
model.result("pg6").run();
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4", 0);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4", 4);
model.result("pg6").feature("lngr1").setIndex("legends", "n_CH4", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO", 0);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO", 4);
model.result("pg6").feature("lngr2").setIndex("legends", "n_CO", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2", 0);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2", 4);
model.result("pg6").feature("lngr3").setIndex("legends", "n_CO2", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2", 0);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2", 4);
model.result("pg6").feature("lngr4").setIndex("legends", "n_H2", 6);
model.result("pg6").run();
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O", 0);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O", 4);
model.result("pg6").feature("lngr5").setIndex("legends", "n_H2O", 6);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "4", 0);
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").set("title", "Mass flux in the r-direction x Pellet radius [T_bulk=1000]");
model.result("pg6").run();
model.result("pg6").set("ylabel", "Mass flux in r-direction (g/(cm<sup>2</sup>.s))");
model.result("pg6").run();
model.result("pg6").set("ylabel", "Mass flux in the r-direction (g/(cm<sup>2</sup>.s))");
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "7", 0);
model.result("pg6").run();
model.result("pg6").set("title", "Mass flux in the r-direction x Pellet radius [T_bulk=1200]");
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").set("title", "Mass flux in the r-direction x Pellet radius [T_bulk=1000]");
model.result("pg6").setIndex("looplevel", "7", 0);
model.result("pg6").run();
model.result("pg6").set("title", "Mass flux in the r-direction x Pellet radius [T_bulk=1200]");
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").run();
```

```
    model.label("New Pellet Model 2016 - Full version (FINAL)2.mph");

    model.physics("g3").feature("dir1").active(false);

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "3", 0);
    model.result("pg5").setIndex("looplevel", "6", 0);
    model.result("pg5").setIndex("looplevel", "7", 0);
    model.result("pg5").run();
    model.result("pg6").run();
    model.result("pg6").setIndex("looplevel", "3", 0);
    model.result("pg6").setIndex("looplevel", "7", 0);
    model.result("pg6").run();
    model.result("pg6").setIndex("looplevel", "5", 0);
    model.result("pg6").run();
    model.result("pg6").run();

    model.label("New Pellet Model 2016 - Full version (FINAL)2.mph");

    model.result("pg6").run();

    return model;
  }

  public static void main(String[] args) {
    Model model = run();
    model = run2(model);
    model = run3(model);
    model = run4(model);
    model = run5(model);
    model = run6(model);
    model = run7(model);
    model = run8(model);
    run9(model);
  }

}
```

Java code: Simulation II – Reactor and Pellet coupled

```
/*
 * Pellet_1D_and_PBR_Final.java
 */

import com.comsol.model.*;
import com.comsol.model.util.*;

/** Model exported on May 27 2016, 11:23 by COMSOL 5.0.0.244. */
public class Pellet_1D_and_PBR_Final {

  public static Model run() {
    Model model = ModelUtil.create("Model");

    model
         .modelPath("C:\\Users\\Fl\u00e1vio\\OneDrive\\Research\\Reforming Reactor\\Master Thesis\\COMSOL-
Models\\Master Final Model");

    model.modelNode().create("comp1");

    model.geom().create("geom1", 1);

    model.mesh().create("mesh1", "geom1");

    model.physics().create("tcs", "ConcentratedSpecies", "geom1", new String[][]{{"x_CH4", "x_CO", "x_CO2",
"x_H2"}});
    model.physics().create("c", "CoefficientFormPDE", "geom1", new String[][]{{"P"}});
    model.physics("c").field("dimensionless").field("P");

    model.study().create("std1");
    model.study("std1").create("stat", "Stationary");
    model.study("std1").feature("stat").activate("tcs", true);
    model.study("std1").feature("stat").activate("c", true);
```

```
    model.geom("geom1").run();

    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 1);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 2);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 3);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 0);
    model.physics("tcs").field("massfraction").component(new String[]{"x_CH4", "x_CO", "x_CO2", "x_H2",
"x_CH45"});
    model.physics("tcs").field("massfraction").component(1, "w_CH4");
    model.physics("tcs").field("massfraction").component(2, "w_CO");
    model.physics("tcs").field("massfraction").component(1, "wb_CH4");
    model.physics("tcs").field("massfraction").component(2, "wb_CO");
    model.physics("tcs").field("massfraction").component(3, "wb_CO2");
    model.physics("tcs").field("massfraction").component(4, "wb_H2");
    model.physics("tcs").field("massfraction").component(5, "wb_H2O");
    model.physics("tcs").prop("SpeciesProperties").setIndex("FromMassConstraint", "5", 0);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 1);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 2);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 3);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 4);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 0);

    model.geom("geom1").create("i1", "Interval");
    model.geom("geom1").runPre("fin");
    model.geom("geom1").run();

    model.modelNode().create("comp2");

    model.geom().create("geom2", 2);

    model.mesh().create("mesh2", "geom2");

    model.param()
         .set("Bed_Parameters_Geometry", "0000000000000000000000000000000000000000", "Bed_Parameters -
Geometry");
    model.param().set("void_fraction", "0.7", "Void Fraction of the bed reactor");
    model.param().set("r_tube", "50[mm]", "Tube's radius");
    model.param().set("h_tube", "10[m]", "Tube's height");
    model.param().set("Ac_tube", "pi*r_tube^2", "Tube's cross sectional area");
    model.param().set("As_tube", "2*pi*r_tube*h_tube", "Tube's side area");
    model.param().set("V_tube", "Ac_tube*h_tube", "Tube's total volume");
    model.param()
         .set("Bed_Parameters_process", "0000000000000000000000000000000000000000000000", "Bed_Parameters -
Process operation");
    model.param().set("U_ave", "1.0[m/s]", "Average velocity of the fluid inside the pipe");
    model.param().set("p_in", "28.9[kg]*g_const/1[cm^2]+p_atm", "Inlet pressure");
    model.param().set("p_out", "25.5[kg]*g_const/1[cm^2]+p_atm", "Outlet pressure");
    model.param().set("p_atm", "1[atm]", "Atmosferic pressure");
    model.param().set("T_in", "550[K]", "Inlet gas temperature");
    model.param().set("T_furnace", "1100[K]", "Furnace temperature");
    model.param()
         .set("rho_gas_0", "c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO", "Gas density at
the inlet");
    model.param()
         .set("Concentratons", "0000000000000000000000000000000000000000000000", "Species inlet/initial
concentrations");
    model.param().set("CO.CH4_0", "1e-6", "Inlet CO-CH4 molar ratio, nCO/nCH4");
    model.param().set("CO2.CH4_0", "0.0072", "Inlet CO2-CH4 molar ratio, nCO2/nCH4");
    model.param().set("H2.CH4_0", "0.0001", "Inlet H2-CH4 molar ratio, nH2/nCH4");
    model.param().set("H2O.CH4_0", "3", "Inlet H2O-CH4 molar ratio, nH2O/nCH4");
    model.param()
         .set("c_CH4_0", "p_in/(R_const*T_in*(1+CO.CH4_0+CO2.CH4_0+H2.CH4_0+H2O.CH4_0))", "Inlet concentration,
CH4");
    model.param().set("c_CO_0", "CO.CH4_0*c_CH4_0", "Inlet concentration, CO");
    model.param().set("c_CO2_0", "CO2.CH4_0*c_CH4_0", "Inlet concentration, CO2");
    model.param().set("c_H2_0", "H2.CH4_0*c_CH4_0", "Inlet concentration, H2");
    model.param().set("c_H2O_0", "H2O.CH4_0*c_CH4_0", "Inlet concentration, H2O");
    model.param().set("c_TOT_0", "c_CH4_0+c_CO_0+c_CO2_0+c_H2_0+c_H2O_0", "Total inlet concentration");
    model.param()
         .set("Molar_fractions", "0000000000000000000000000000000000000000000000", "Species inlet/initial
molar fractions");
    model.param().set("x_CH4_0", "c_CH4_0/c_TOT_0", "Molar fraction (initial), CH4");
    model.param().set("x_CO_0", "c_CO_0/c_TOT_0", "Molar fraction (initial), CO");
    model.param().set("x_CO2_0", "c_CO2_0/c_TOT_0", "Molar fraction (initial), CO2");
    model.param().set("x_H2_0", "c_H2_0/c_TOT_0", "Molar fraction (initial), H2");
    model.param().set("x_H2O_0", "c_H2O_0/c_TOT_0", "Molar fraction (initial), H2O");
    model.param()
         .set("Mass_fractions", "0000000000000000000000000000000000000000000000", "Species inlet/initial
mass fractions");
    model.param()
```

```
        .set("w_CH4_0", "c_CH4_0*M_CH4/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), CH4");
    model.param()
        .set("w_CO_0", "c_CO_0*M_CO/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), CO");
    model.param()
        .set("w_CO2_0", "c_CO2_0*M_CO2/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), CO2");
    model.param()
        .set("w_H2_0", "c_H2_0*M_H2/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), H2");
    model.param()
        .set("w_H2O_0", "c_H2O_0*M_H2O/(c_CH4_0*M_CH4+c_H2O_0*M_H2O+c_H2_0*M_H2+c_CO2_0*M_CO2+c_CO_0*M_CO)",
"Mass fraction (initial), H2O");
    model.param().set("Pellet_parameters", "0000000000000000000000000000000000000000", "Catalyst pellet
paramenters");
    model.param().set("rho_bulk", "1100[kg/m^3]", "Catalyst bulk density (weight of catalyst per volume of the
bed)");
    model.param()
        .set("rho_catalyst", "rho_bulk/(1-void_fraction)", "Catalyst density (weight of catalyst per volume of
catalyst pellet)");
    model.param()
        .set("Pellet_parameters_geometry", "0000000000000000000000000000000000000000", "Pellet's parameters -
Geometry");
    model.param().set("r_pellet", "1[cm]", "Pellet's radius");
    model.param().set("Ax_pellet", "4*pi*r_pellet^2", "Pellet's external surface area");
    model.param().set("A_pores", "792679.4825[cm^2/g]", "Total pore area per weight");
    model.param().set("Vm_pellet", "(4/3)*pi*r_pellet^3", "Pellet's \"monolitic\" volume");
    model.param().set("eps_V", "0.016328", "Pellet's volumetric porosity");
    model.param().set("eps_A", "0.4824", "Pellet's surface porosity");
    model.param().set("eps_I", "A_pores*rho_catalyst*(1-eps_V)", "Area to volume interfacial factor");
    model.param().set("Tau", "1/(eps_V)^(1/3)", "Tortuosity");
    model.param().set("lambda_solid", "98.6[W/(m*K)]", "Heat transfer coeff - Conduction (Solid Phase)");
    model.param().set("Species_parameters", "0000000000000000000000000000000000000000000000.", "Species
parameters");
    model.param().set("M_CH4", "16.044[kg/kmol]", "Molar Mass, CH4");
    model.param().set("M_CO", "28.001[kg/kmol]", "Molar Mass, CO");
    model.param().set("M_CO2", "44.01[kg/kmol]", "Molar Mass, CO2");
    model.param().set("M_H2", "2.016[kg/kmol]", "Molar Mass, H2");
    model.param().set("M_H2O", "18.02[kg/kmol]", "Molar Mass, H2O");
    model.param().set("Reaction_factors", "0000000000000000000000000000000000000000", "Reaction rate contants");
    model.param()
        .set("A_k1", "4.255*1e15[(kmol*bar)/(kg*h)]", "Preexponential factor of rate coefficient, reaction
1");
    model.param()
        .set("A_k2", "1.955*1e6*1[kmol/(bar*kg*h)]", "Preexponential factor of rate coefficient, reaction 2");
    model.param()
        .set("A_k3", "1.02*1e15[(kmol*bar)/(kg*h)]", "Preexponential factor of rate coefficient, reaction 3");
    model.param().set("A_CO", "8.23*1e-5[1/bar]", "Preexponential factor of adsorption contant, CO");
    model.param().set("A_H2", "6.12*1e-9[1/bar]", "Preexponential factor of adsorption contant, H2");
    model.param().set("A_CH4", "6.65*1e-4[1/bar]", "Preexponential factor of adsorption contant, CH4");
    model.param().set("A_H2O", "1.77*1e5[bar/bar]", "Preexponential factor of adsorption contant, H2O");
    model.param()
        .set("Actv_energy", "0000000000000000000000000000000000000000000000000000000000000000", "Activation
energy of the reactions");
    model.param().set("E_1", "240.1[kJ/mol]", "Activation Energy, reaction 1");
    model.param().set("E_2", "67.13[kJ/mol]", "Activation Energy, reaction 2");
    model.param().set("E_3", "243.9[kJ/mol]", "Activation Energy, reaction 3");
    model.param()
        .set("Energy_prop", "0000000000000000000000000000000000000000000000000000000000000000", "Energy
balance related properties");
    model.param().set("dH_CO", "-70.65[kJ/mol]", "Adsorption enthalpy, CO");
    model.param().set("dH_H2", "-82.9[kJ/mol]", "Adsorption enthalpy, H2");
    model.param().set("dH_CH4", "-38.28[kJ/mol]", "Adsorption enthalpy, CH4");
    model.param().set("dH_H2O", "88.68[kJ/mol]", "Adsorption enthalpy, H2O");
    model.param().set("dH01", "206.1[kJ/mol]", "Enthalpy change of reaction I at 298K");
    model.param().set("dH02", "-41.15[kJ/mol]", "Enthalpy change of reaction II at 298K");
    model.param().set("dH03", "164.9[kJ/mol]", "Enthalpy change of reaction III at 298K");
    model.param()
        .set("Model_parameters", "0000000000000000000000000000000000000000000000000000000000000000", "Model
Parameters");
    model.param().set("p_TOT", "p_in", "Total pressure over the domain");
    model.param().set("T", "T_in", "Temperature over the domain");

    model.variable().create("var1");
    model.variable("var1").model("comp2");

    model.geom("geom2").run();

    model.variable("var1").set("x_H2O", "1-(x_CH4+x_CO+x_CO2+x_H2)", "Molar fraction, H2O");
    model.variable("var1").set("c_CH4", "c_TOT*x_CH4", "Molar concentration, CH4");
```

```
    model.variable("var1").set("c_CO", "c_TOT*x_CO", "Molar concentration, CO");
    model.variable("var1").set("c_CO2", "c_TOT*x_CO2", "Molar concentration, CO2");
    model.variable("var1").set("c_H2", "c_TOT*x_H2", "Molar concentration, H2");
    model.variable("var1").set("c_H2O", "c_TOT*x_H2O", "Molar concentration, H2O");
    model.variable("var1").set("c_TOT", "p_TOT/(R_const*T)", "Total concentration");
    model.variable("var1")
         .set("w_CH4", "c_CH4*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
CH4");
    model.variable("var1")
         .set("w_CO", "c_CO*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
CO");
    model.variable("var1")
         .set("w_CO2", "c_CO2*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
CO2");
    model.variable("var1")
         .set("w_H2", "c_H2*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
H2");
    model.variable("var1")
         .set("w_H2O", "c_H2O*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
H2O");
    model.variable("var1")
         .set("rho_gas", "p_TOT*(x_CH4*M_CH4+x_H2O*M_H2O+x_H2*M_H2+x_CO2*M_CO2+x_CO*M_CO)/(R_const*T)", "Gas
density");
    model.variable("var1").set("j_CH4", "c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4x", "Molar flux, CH4");
    model.variable("var1").set("j_CO", "c_TOT*(invA_12+invA_22+invA_32+invA_42)*x_COx", "Molar flux, CO");
    model.variable("var1").set("j_CO2", "c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2x", "Molar flux, CO2");
    model.variable("var1").set("j_H2", "c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2x", "Molar flux, H2");
    model.variable("var1")
         .set("j_H2O", "-
((DKeff_H2O/DKeff_CH4)*(c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4x)+(DKeff_H2O/DKeff_CO)*(c_TOT*(invA_12+in
vA_22+invA_32+invA_42)*x_COx)+(DKeff_H2O/DKeff_CO2)*(c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2x)+(DKeff_H2O
/DKeff_H2)*(c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2x))", "Molar flux, H2O");
    model.variable("var1").label("Fluid Properties - Concentrations, molar and mass fractions");
    model.variable().create("var2");
    model.variable("var2").model("comp2");
    model.variable("var2").set("k1", "A_k1*exp(-E_1/(R_const*T))", "Rate coefficient of reaction I");
    model.variable("var2").set("k2", "A_k2*exp(-E_2/(R_const*T))", "Rate coefficient of reaction II");
    model.variable("var2").set("k3", "A_k3*exp(-E_3/(R_const*T))", "Rate coefficient of reaction III");
    model.variable("var2").set("K_CO", "A_CO*exp(-dH_CO/(R_const*T))", "Adsorption constant for CO");
    model.variable("var2").set("K_H2", "A_H2*exp(-dH_H2/(R_const*T))", "Adsorption constant for H2");
    model.variable("var2").set("K_CH4", "A_CH4*exp(-dH_CH4/(R_const*T))", "Adsorption constant for CH4");
    model.variable("var2").set("K_H2O", "A_H2O*exp(-dH_H2O/(R_const*T))", "Dissociative adsorption constant for
H2O");
    model.variable("var2").set("K1", "exp(30.114-(26830[K]/T))[bar^2]", "Equilibrium constant of reaction I");
    model.variable("var2").set("K2", "exp(-4.036+(4400[K]/T))", "Equilibrium constant of reaction II");
    model.variable("var2").set("K3", "K1*K2", "Equilibrium constant of reaction III");
    model.variable("var2")
         .set("r1", "k1*sqrt(p_TOT/1[bar])*sqrt(x_H2)*((p_TOT*(x_CH4*x_H2O)/x_H2)-
((p_TOT^3)*(x_H2^2)*x_CO/K1))/(DEN_star)^2", "Rate of reaction I");
    model.variable("var2")
         .set("r2", "k2*p_TOT*x_H2*(((p_TOT^2)*x_CO*x_H2O)-((p_TOT^2)*x_H2*x_CO2/K2))/(DEN_star)^2", "Rate of
reaction II");
    model.variable("var2")
         .set("r3", "k3*sqrt(p_TOT/1[bar])*sqrt(x_H2)*((p_TOT*x_CH4*x_H2O/(x_H2^2))-
((p_TOT^3)*(x_H2^2)*x_CO2/K3))/(DEN_star)^2", "Rate of reaction III");
    model.variable("var2")
         .set("DEN_star",
"(p_TOT*x_H2)+((p_TOT^2)*K_CO*x_H2*x_CO)+((p_TOT^2)*K_H2*(x_H2^2))+((p_TOT^2)*K_CH4*x_H2*x_CH4)+(p_TOT*K_H2O*x_
H2O)", "Denominator of reaction equaitons");
    model.variable("var2")
         .set("r1_A", "r1*rho_catalyst*(1-eps_V)/eps_I", "Rate of reaction I, per unit of porous area");
    model.variable("var2")
         .set("r2_A", "r2*rho_catalyst*(1-eps_V)/eps_I", "Rate of reaction II, per unit of porous area");
    model.variable("var2")
         .set("r3_A", "r3*rho_catalyst*(1-eps_V)/eps_I", "Rate of reaction III, per unit of porous area");
    model.variable("var2").label("Reaction rates and reaction kinectic constants");
    model.variable().create("var3");
    model.variable("var3").model("comp2");
    model.variable("var3")
         .set("cp_CH4", "14.15[kJ/kmol/K]+75.499e-3*T[kJ/kmol/K^2]-17.991e-6*T^2[kJ/kmol/K^3]", "Heat capacity,
CH4");
    model.variable("var3")
         .set("S.cp.dT_CH4", "R_const*((1.702*T)+(((9.081e-3)/2)*T^2/1[K])-(((2.164e-6)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, CH4 - gives the change in enthalpy for the specie from ref. state to T
at CONSTANT PRESSURE");
    model.variable("var3")
         .set("S.cp.dT_CO", "R_const*((3.376*T)+(((0.557e-3)/2)*T^2/1[K])+(((0.031e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, CO - gives the change in enthalpy for the specie from ref. state to T at
CONSTANT PRESSURE");
    model.variable("var3")
```

```
        .set("S.cp.dT_CO2", "R_const*((5.457*T)+(((1.045e-3)/2)*T^2/1[K])+(((1.157e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, CO2 - gives the change in enthalpy for the specie from ref. state to T
at CONSTANT PRESSURE");
    model.variable("var3")
        .set("S.cp.dT_H2", "R_const*((3.249*T)+(((0.422e-3)/2)*T^2/1[K])-(((0.083e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, H2 - gives the change in enthalpy for the specie from ref. state to T at
CONSTANT PRESSURE");
    model.variable("var3")
        .set("S.cp.dT_H2O", "R_const*((3.470*T)+(((1.450e-3)/2)*T^2/1[K])-(((0.121e5)/3)*T^3/1[K^2]))",
"Integration of cp(T)dt from 298 to T, H2O - gives the change in enthalpy for the specie from ref. state to T
at CONSTANT PRESSURE");
    model.variable("var3")
        .set("dH1", "dH01+3*S.cp.dT_H2+S.cp.dT_CO-S.cp.dT_CH4-S.cp.dT_H2O", "Total change in enthalpy for
reaction 1 (assuming constant pressure)");
    model.variable("var3")
        .set("dH2", "dH02+S.cp.dT_CO2+S.cp.dT_H2-S.cp.dT_CO-S.cp.dT_H2O", "Total change in enthalpy for
reaction 2 (assuming constant pressure)");
    model.variable("var3")
        .set("dH3", "dH03+4*S.cp.dT_H2+S.cp.dT_CO2-S.cp.dT_CH4-2*S.cp.dT_H2O", "Total change in enthalpy for
reaction 3 (assuming constant pressure)");
    model.variable("var3").label("Enthalpy of reactions");
    model.variable().create("var4");
    model.variable("var4").model("comp2");
    model.variable("var4").set("A_11", "-
((x_CO/Deff_CH4.CO)+(x_CO2/Deff_CH4.CO2)+(x_H2/Deff_CH4.H2)+(1/DKeff_CH4))");
    model.variable("var4").set("A_12", "x_CH4/Deff_CH4.CO");
    model.variable("var4").set("A_13", "x_CH4/Deff_CH4.CO2");
    model.variable("var4").set("A_14", "x_CH4/Deff_CH4.H2");
    model.variable("var4").set("A_21", "x_CO/Deff_CH4.CO");
    model.variable("var4").set("A_22", "-
((x_CH4/Deff_CH4.CO)+(x_CO2/Deff_CO.CO2)+(x_H2/Deff_CO.H2)+(1/DKeff_CO))");
    model.variable("var4").set("A_23", "x_CO/Deff_CO.CO2");
    model.variable("var4").set("A_24", "x_CO/Deff_CO.H2");
    model.variable("var4").set("A_31", "x_CO2/Deff_CH4.CO2");
    model.variable("var4").set("A_32", "x_CO2/Deff_CO.CO2");
    model.variable("var4").set("A_33", "-
((x_CH4/Deff_CH4.CO2)+(x_CO/Deff_CO.CO2)+(x_H2/Deff_CO2.H2)+(1/DKeff_CO2))");
    model.variable("var4").set("A_34", "x_CO2/Deff_CO2.H2");
    model.variable("var4").set("A_41", "x_H2/Deff_CH4.H2");
    model.variable("var4").set("A_42", "x_H2/Deff_CO.H2");
    model.variable("var4").set("A_43", "x_H2/Deff_CO2.H2");
    model.variable("var4").set("A_44", "-
((x_CH4/Deff_CH4.H2)+(x_CO/Deff_CO.H2)+(x_CO2/Deff_CO2.H2)+(1/DKeff_H2))");
    model.variable("var4").set("AI_11", "1");
    model.variable("var4").set("B_22", "A_11*A_22-A_21*A_12");
    model.variable("var4").set("B_23", "A_11*A_23-A_21*A_13");
    model.variable("var4").set("B_24", "A_11*A_24-A_21*A_14");
    model.variable("var4").set("B_32", "A_11*A_32-A_31*A_12");
    model.variable("var4").set("B_33", "A_11*A_33-A_31*A_13");
    model.variable("var4").set("B_34", "A_11*A_34-A_31*A_14");
    model.variable("var4").set("B_42", "A_11*A_42-A_41*A_12");
    model.variable("var4").set("B_43", "A_11*A_43-A_41*A_13");
    model.variable("var4").set("B_44", "A_11*A_44-A_41*A_14");
    model.variable("var4").set("BI_21", "A_21*AI_11");
    model.variable("var4").set("BI_31", "A_31*AI_11");
    model.variable("var4").set("BI_41", "A_41*AI_11");
    model.variable("var4").set("C_33", "B_22*B_33-B_32*B_23");
    model.variable("var4").set("C_34", "B_22*B_34-B_32*B_24");
    model.variable("var4").set("C_43", "B_22*B_43-B_42*B_23");
    model.variable("var4").set("C_44", "B_22*B_44-B_42*B_24");
    model.variable("var4").set("CI_31", "B_22*BI_31-B_32*BI_21");
    model.variable("var4").set("CI_32", "-B_32*A_11");
    model.variable("var4").set("CI_33", "B_22*A_11");
    model.variable("var4").set("CI_41", "B_22*BI_41-B_42*BI_21");
    model.variable("var4").set("CI_42", "-B_42*A_11");
    model.variable("var4").set("CI_44", "B_22*A_11");
    model.variable("var4").set("D_44", "C_33*C_44-C_43*C_34");
    model.variable("var4").set("DI_41", "C_33*CI_41-C_43*CI_31");
    model.variable("var4").set("DI_42", "C_33*CI_42-C_43*CI_32");
    model.variable("var4").set("DI_43", "-C_43*CI_33");
    model.variable("var4").set("DI_44", "C_33*CI_44");
    model.variable("var4").set("E_11", "A_11*D_44");
    model.variable("var4").set("E_12", "A_12*D_44");
    model.variable("var4").set("E_13", "A_13*D_44");
    model.variable("var4").set("E_22", "B_22*D_44");
    model.variable("var4").set("E_23", "B_23*D_44");
    model.variable("var4").set("E_33", "C_33*D_44");
    model.variable("var4").set("EI_11", "D_44*AI_11-DI_41*A_14");
    model.variable("var4").set("EI_12", "-DI_42*A_14");
    model.variable("var4").set("EI_13", "-DI_43*A_14");
    model.variable("var4").set("EI_14", "-DI_44*A_14");
```

```
    model.variable("var4").set("EI_21", "D_44*BI_21-DI_41*B_24");
    model.variable("var4").set("EI_22", "D_44*A_11-DI_42*B_24");
    model.variable("var4").set("EI_23", "-DI_43*B_24");
    model.variable("var4").set("EI_24", "-DI_44*B_24");
    model.variable("var4").set("EI_31", "D_44*CI_31-DI_41*C_34");
    model.variable("var4").set("EI_32", "D_44*CI_32-DI_42*C_34");
    model.variable("var4").set("EI_33", "D_44*CI_33-DI_43*C_34");
    model.variable("var4").set("EI_34", "-DI_44*C_34");
    model.variable("var4").set("F_11", "E_33*E_11");
    model.variable("var4").set("F_12", "E_33*E_12");
    model.variable("var4").set("F_22", "E_33*E_22");
    model.variable("var4").set("FI_11", "E_33*EI_11-EI_31*E_13");
    model.variable("var4").set("FI_12", "E_33*EI_12-EI_32*E_13");
    model.variable("var4").set("FI_13", "E_33*EI_13-EI_33*E_13");
    model.variable("var4").set("FI_14", "E_33*EI_14-EI_34*E_13");
    model.variable("var4").set("FI_21", "E_33*EI_21-EI_31*E_23");
    model.variable("var4").set("FI_22", "E_33*EI_22-EI_32*E_23");
    model.variable("var4").set("FI_23", "E_33*EI_23-EI_33*E_23");
    model.variable("var4").set("FI_24", "E_33*EI_24-EI_34*E_23");
    model.variable("var4").set("G_11", "F_22*F_11");
    model.variable("var4").set("GI_11", "F_22*FI_11-FI_21*F_12");
    model.variable("var4").set("GI_12", "F_22*FI_12-FI_22*F_12");
    model.variable("var4").set("GI_13", "F_22*FI_13-FI_23*F_12");
    model.variable("var4").set("GI_14", "F_22*FI_14-FI_24*F_12");
    model.variable("var4").set("invA_11", "GI_11/G_11");
    model.variable("var4").set("invA_12", "GI_12/G_11");
    model.variable("var4").set("invA_13", "GI_13/G_11");
    model.variable("var4").set("invA_14", "GI_14/G_11");
    model.variable("var4").set("invA_21", "FI_21/F_22");
    model.variable("var4").set("invA_22", "FI_22/F_22");
    model.variable("var4").set("invA_23", "FI_23/F_22");
    model.variable("var4").set("invA_24", "FI_24/F_22");
    model.variable("var4").set("invA_31", "EI_31/E_33");
    model.variable("var4").set("invA_32", "EI_32/E_33");
    model.variable("var4").set("invA_33", "EI_33/E_33");
    model.variable("var4").set("invA_34", "EI_34/E_33");
    model.variable("var4").set("invA_41", "DI_41/D_44");
    model.variable("var4").set("invA_42", "DI_42/D_44");
    model.variable("var4").set("invA_43", "DI_43/D_44");
    model.variable("var4").set("invA_44", "DI_44/D_44");
    model.variable("var4").label("Inverse matrix A^(4x4)");
    model.variable().create("var5");
    model.variable("var5").model("comp2");
    model.variable("var5").set("D_CH4.CO", "D_CH4_CO(T)", "Binary diffusivity pair CH4-CO");
    model.variable("var5").set("D_CH4.CO2", "D_CH4_CO2(T)", "Binary diffusivity pair CH4-CO2");
    model.variable("var5").set("D_CH4.H2", "D_CH4_H2(T)", "Binary diffusivity pair CH4-H2");
    model.variable("var5").set("D_CH4.H2O", "D_CH4_H2O(T)", "Binary diffusivity pair CH4-H2O");
    model.variable("var5").set("D_CO.CO2", "D_CO_CO2(T)", "Binary diffusivity pair CO-CO2");
    model.variable("var5").set("D_CO.H2", "D_CO_H2(T)", "Binary diffusivity pair CO-H2");
    model.variable("var5").set("D_CO.H2O", "D_CO_H2O(T)", "Binary diffusivity pair CO-H2O");
    model.variable("var5").set("D_CO2.H2", "D_CO2_H2(T)", "Binary diffusivity pair CO2-H2");
    model.variable("var5").set("D_CO2.H2O", "D_CO2_H2O(T)", "Binary diffusivity pair CO2-H2O");
    model.variable("var5").set("D_H2.H2O", "D_H2_H2O(T)", "Binary diffusivity pair H2-H2O");
    model.variable("var5")
         .set("Effective_Diffusivity", "0000000000000000000000000000000000000000", "Effective diffusivities");
    model.variable("var5").set("Deff_CH4.CO", "(eps_V/Tau)*D_CH4.CO", "Effective diffusivity pair CH4-CO");
    model.variable("var5").set("Deff_CH4.CO2", "(eps_V/Tau)*D_CH4.CO2", "Effective diffusivity pair CH4-CO2");
    model.variable("var5").set("Deff_CH4.H2", "(eps_V/Tau)*D_CH4.H2", "Effective diffusivity pair CH4-H2");
    model.variable("var5").set("Deff_CH4.H2O", "(eps_V/Tau)*D_CH4.H2O", "Effective diffusivity pair CH4-H2O");
    model.variable("var5").set("Deff_CO.CO2", "(eps_V/Tau)*D_CO.CO2", "Effective diffusivity pair CO-CO2");
    model.variable("var5").set("Deff_CO.H2", "(eps_V/Tau)*D_CO.H2", "Effective diffusivity pair CO-H2");
    model.variable("var5").set("Deff_CO.H2O", "(eps_V/Tau)*D_CO.H2O", "Effective diffusivity pair CO-H2O");
    model.variable("var5").set("Deff_CO2.H2", "(eps_V/Tau)*D_CO2.H2", "Effective diffusivity pair CO2-H2");
    model.variable("var5").set("Deff_CO2.H2O", "(eps_V/Tau)*D_CO2.H2O", "Effective diffusivity pair CO2-H2O");
    model.variable("var5").set("Deff_H2.H2O", "(eps_V/Tau)*D_H2.H2O", "Effective diffusivity pair H2-H2O");
    model.variable("var5")
         .set("Knudsen_Diffusivity", "0000000000000000000000000000000000000000", "Knudsen diffusivities");
    model.variable("var5").set("DK_CH4", "1e-6[m^2/s]");
    model.variable("var5").set("DK_CO", "1e-6[m^2/s]");
    model.variable("var5").set("DK_CO2", "1e-6[m^2/s]");
    model.variable("var5").set("DK_H2", "1e-6[m^2/s]");
    model.variable("var5").set("DK_H2O", "1e-6[m^2/s]");
    model.variable("var5")
         .set("Knudsen_Eff_Diffusivity", "0000000000000000000000000000000000000000", "Knudsen effective
diffusivities");
    model.variable("var5").set("DKeff_CH4", "(eps_V/Tau)*DK_CH4");
    model.variable("var5").set("DKeff_CO", "(eps_V/Tau)*DK_CO");
    model.variable("var5").set("DKeff_CO2", "(eps_V/Tau)*DK_CO2");
    model.variable("var5").set("DKeff_H2", "(eps_V/Tau)*DK_H2");
    model.variable("var5").set("DKeff_H2O", "(eps_V/Tau)*DK_H2O");
    model.variable("var5").label("Diffusivities and Knudsen diffusivities");
```

```
    model.func().create("int1", "Interpolation");
    model.func("int1").model("comp2");
    model.func("int1")
         .set("table", new String[][]{{"300", "7.94241E-07"}, {"400", "1.32151E-06"}, {"500", "1.94561E-06"},
{"600", "2.65884E-06"}, {"700", "3.45523E-06"}, {"800", "4.32979E-06"}, {"900", "5.27835E-06"}, {"1000",
"6.29753E-06"}, {"1100", "7.38460E-06"}, {"1200", "8.53726E-06"},
         {"1300", "9.75361E-06"}});
    model.func("int1").label("Diff_Interpolation-CH4-CO");
    model.func("int1").label("Diffusivity CH4-CO");
    model.func("int1").set("funcname", "D_CH4_CO");
    model.func().duplicate("int2", "int1");
    model.func("int2").label("Diffusivity CH4-CO2");
    model.func("int2").set("funcname", "D_CH4_CO2");
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2")
         .set("table", new String[][]{{"300", "6.10147E-07"}, {"400", "1.03713E-06"}, {"500", "1.54683E-06"},
{"600", "2.13075E-06"}, {"700", "2.78367E-06"}, {"800", "3.50184E-06"}, {"900", "4.28212E-06"}, {"1000",
"5.12175E-06"}, {"1100", "6.01823E-06"}, {"1200", "6.96940E-06"},
         {"1300", "7.97337E-06"}});
    model.func().duplicate("int3", "int2");
    model.func("int3").label("Diffusivity CH4-H2");
    model.func("int3").set("funcname", "D_CH4_H2");
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3")
         .set("table", new String[][]{{"300", "2.57271E-06"}, {"400", "4.24235E-06"}, {"500", "6.21612E-06"},
{"600", "8.46912E-06"}, {"700", "1.09814E-05"}, {"800", "1.37373E-05"}, {"900", "1.67247E-05"}, {"1000",
"1.99340E-05"}, {"1100", "2.33575E-05"}, {"1200", "2.69887E-05"},
         {"1300", "3.08217E-05"}});
    model.func().duplicate("int4", "int3");
    model.func("int4").label("Diffusivity CH4-H2O");
    model.func("int4").set("funcname", "D_CH4_H2O");
    model.func().duplicate("int5", "int1");
    model.func("int5").label("Diffusivity CO-CO2");
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1")
         .set("table", new String[][]{{"300", "7.94241E-07"}, {"400", "1.32151E-06"}, {"500", "1.94561E-06"},
{"600", "2.65884E-06"}, {"700", "3.45523E-06"}, {"800", "4.32979E-06"}, {"900", "5.27835E-06"}, {"1000",
"6.29753E-06"}, {"1100", "7.38460E-06"}, {"1200", "8.53726E-06"},
         {"1300", "9.75361E-06"}});
```

```
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2")
         .set("table", new String[][]{{"300", "6.10147E-07"}, {"400", "1.03713E-06"}, {"500", "1.54683E-06"},
{"600", "2.13075E-06"}, {"700", "2.78367E-06"}, {"800", "3.50184E-06"}, {"900", "4.28212E-06"}, {"1000",
"5.12175E-06"}, {"1100", "6.01823E-06"}, {"1200", "6.96940E-06"},
         {"1300", "7.97337E-06"}});
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3")
         .set("table", new String[][]{{"300", "2.57271E-06"}, {"400", "4.24235E-06"}, {"500", "6.21612E-06"},
{"600", "8.46912E-06"}, {"700", "1.09814E-05"}, {"800", "1.37373E-05"}, {"900", "1.67247E-05"}, {"1000",
"1.99340E-05"}, {"1100", "2.33575E-05"}, {"1200", "2.69887E-05"},
         {"1300", "3.08217E-05"}});
    model.func("int4")
         .set("table", new String[][]{{"300", "7.55730E-07"}, {"400", "1.33939E-06"}, {"500", "2.06594E-06"},
{"600", "2.92083E-06"}, {"700", "3.89218E-06"}, {"800", "4.97034E-06"}, {"900", "6.14777E-06"}, {"1000",
"7.41867E-06"}, {"1100", "8.77864E-06"}, {"1200", "1.02242E-05"},
         {"1300", "1.17527E-05"}});

    return model;
  }

  public static Model run2(Model model) {
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").set("funcname", "D_CO_CO2");
    model.func("int5")
         .set("table", new String[][]{{"300", "5.59985E-07"}, {"400", "9.38342E-07"}, {"500", "1.38693E-06"},
{"600", "1.89985E-06"}, {"700", "2.47294E-06"}, {"800", "3.10275E-06"}, {"900", "3.78630E-06"}, {"1000",
"4.52106E-06"}, {"1100", "5.30492E-06"}, {"1200", "6.13611E-06"},
         {"1300", "7.01315E-06"}});
    model.func().duplicate("int6", "int5");
    model.func("int6").label("Diffusivity CO-H2");
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").set("funcname", "D_CO_H2");
    model.func().duplicate("int7", "int6");
    model.func("int6")
         .set("table", new String[][]{{"300", "2.71700E-06"}, {"400", "4.44794E-06"}, {"500", "6.49076E-06"},
{"600", "8.81834E-06"}, {"700", "1.14109E-05"}, {"800", "1.42539E-05"}, {"900", "1.73362E-05"}, {"1000",
"2.06490E-05"}, {"1100", "2.41846E-05"}, {"1200", "2.79362E-05"},
         {"1300", "3.18979E-05"}});
    model.func("int7").label("Diffusivity CO-H2O");
    model.func("int7").set("funcname", "D_CO_H2O");
```

```
    model.func("int7")
          .set("table", new String[][]{{"300", "7.81705E-07"}, {"400", "1.37584E-06"}, {"500", "2.10820E-06"},
{"600", "2.96364E-06"}, {"700", "3.93029E-06"}, {"800", "4.99923E-06"}, {"900", "6.16392E-06"}, {"1000",
"7.41957E-06"}, {"1100", "8.76251E-06"}, {"1200", "1.01897E-05"},
         {"1300", "1.16987E-05"}});
    model.func().duplicate("int8", "int6");
    model.func("int8").label("Diffusivity CO2-H2");
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").remove("table", 0);
    model.func("int8").set("funcname", "D_CO2_H2");
    model.func().duplicate("int9", "int8");
    model.func("int9").label("Diffusivity CO2-H2O");
    model.func("int9").set("funcname", "D_CO2_H2O");
    model.func().duplicate("int10", "int9");
    model.func("int10").label("Diffusivity H2-H2O");
    model.func("int10").set("funcname", "D_H2_H2O");
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int1").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int2").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int3").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int4").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int5").remove("table", 0);
    model.func("int6").remove("table", 0);
```

```
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int6").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int7").remove("table", 0);
    model.func("int1")
         .set("table", new String[][]{{"300", "7.94241E-07"}, {"400", "1.32151E-06"}, {"500", "1.94561E-06"},
{"600", "2.65884E-06"}, {"700", "3.45523E-06"}, {"800", "4.32979E-06"}, {"900", "5.27835E-06"}, {"1000",
"6.29753E-06"}, {"1100", "7.38460E-06"}, {"1200", "8.53726E-06"},
         {"1300", "9.75361E-06"}});
    model.func("int2")
         .set("table", new String[][]{{"300", "6.10147E-07"}, {"400", "1.03713E-06"}, {"500", "1.54683E-06"},
{"600", "2.13075E-06"}, {"700", "2.78367E-06"}, {"800", "3.50184E-06"}, {"900", "4.28212E-06"}, {"1000",
"5.12175E-06"}, {"1100", "6.01823E-06"}, {"1200", "6.96940E-06"},
         {"1300", "7.97337E-06"}});
    model.func("int3")
         .set("table", new String[][]{{"300", "2.57271E-06"}, {"400", "4.24235E-06"}, {"500", "6.21612E-06"},
{"600", "8.46912E-06"}, {"700", "1.09814E-05"}, {"800", "1.37373E-05"}, {"900", "1.67247E-05"}, {"1000",
"1.99340E-05"}, {"1100", "2.33575E-05"}, {"1200", "2.69887E-05"},
         {"1300", "3.08217E-05"}});
    model.func("int4")
         .set("table", new String[][]{{"300", "7.55730E-07"}, {"400", "1.33939E-06"}, {"500", "2.06594E-06"},
{"600", "2.92083E-06"}, {"700", "3.89218E-06"}, {"800", "4.97034E-06"}, {"900", "6.14777E-06"}, {"1000",
"7.41867E-06"}, {"1100", "8.77864E-06"}, {"1200", "1.02242E-05"},
         {"1300", "1.17527E-05"}});
    model.func("int5")
         .set("table", new String[][]{{"300", "5.59985E-07"}, {"400", "9.38342E-07"}, {"500", "1.38693E-06"},
{"600", "1.89985E-06"}, {"700", "2.47294E-06"}, {"800", "3.10275E-06"}, {"900", "3.78630E-06"}, {"1000",
"4.52106E-06"}, {"1100", "5.30492E-06"}, {"1200", "6.13611E-06"},
         {"1300", "7.01315E-06"}});
    model.func("int6")
         .set("table", new String[][]{{"300", "2.71700E-06"}, {"400", "4.44794E-06"}, {"500", "6.49076E-06"},
{"600", "8.81834E-06"}, {"700", "1.14109E-05"}, {"800", "1.42539E-05"}, {"900", "1.73362E-05"}, {"1000",
"2.06490E-05"}, {"1100", "2.41846E-05"}, {"1200", "2.79362E-05"},
         {"1300", "3.18979E-05"}});
    model.func("int7")
         .set("table", new String[][]{{"300", "7.81705E-07"}, {"400", "1.37584E-06"}, {"500", "2.10820E-06"},
{"600", "2.96364E-06"}, {"700", "3.93029E-06"}, {"800", "4.99923E-06"}, {"900", "6.16392E-06"}, {"1000",
"7.41957E-06"}, {"1100", "8.76251E-06"}, {"1200", "1.01897E-05"},
         {"1300", "1.16987E-05"}});
    model.func("int8")
         .set("table", new String[][]{{"300", "2.26297E-06"}, {"400", "3.75182E-06"}, {"500", "5.51309E-06"},
{"600", "7.52520E-06"}, {"700", "9.77087E-06"}, {"800", "1.22358E-05"}, {"900", "1.49086E-05"}, {"1000",
"1.77801E-05"}, {"1100", "2.08427E-05"}, {"1200", "2.40904E-05"},
         {"1300", "2.75178E-05"}});
    model.func("int9")
         .set("table", new String[][]{{"300", "5.42252E-07"}, {"400", "9.66518E-07"}, {"500", "1.49969E-06"},
{"600", "2.13157E-06"}, {"700", "2.85357E-06"}, {"800", "3.65849E-06"}, {"900", "4.54033E-06"}, {"1000",
"5.49427E-06"}, {"1100", "6.51647E-06"}, {"1200", "7.60389E-06"},
         {"1300", "8.75414E-06"}});
    model.func("int10")
         .set("table", new String[][]{{"300", "2.77926E-06"}, {"400", "4.80037E-06"}, {"500", "7.24387E-06"},
{"600", "1.00604E-05"}, {"700", "1.32171E-05"}, {"800", "1.66924E-05"}, {"900", "2.04704E-05"}, {"1000",
"2.45387E-05"}, {"1100", "2.88863E-05"}, {"1200", "3.35033E-05"},
         {"1300", "3.83805E-05"}});
    model.func("int10").set("argunit", "K");
    model.func("int10").set("fununit", "m^2/s");
    model.func("int9").set("fununit", "m^2/s");
    model.func("int9").set("argunit", "K");
    model.func("int8").set("fununit", "m^2/s");
    model.func("int8").set("argunit", "K");
    model.func("int7").set("fununit", "m^2/s");
    model.func("int7").set("argunit", "K");
    model.func("int6").set("fununit", "m^2/s");
    model.func("int6").set("argunit", "K");
```

```
model.func("int5").set("fununit", "m^2/s");
model.func("int5").set("argunit", "K");
model.func("int4").set("fununit", "m^2/s");
model.func("int4").set("argunit", "K");
model.func("int3").set("fununit", "m^2/s");
model.func("int3").set("argunit", "K");
model.func("int2").set("fununit", "m^2/s");
model.func("int2").set("argunit", "K");
model.func("int1").set("fununit", "m^2/s");
model.func("int1").set("argunit", "K");

model.geom("geom2").create("sq1", "Square");
model.geom("geom2").runPre("fin");

model.physics().create("c2", "CoefficientFormPDE", "geom2", new String[][]{{"u"}});

model.study("std1").feature("stat").activate("c2", true);

model.geom("geom2").run();

model.physics("c2").field("dimensionless").component(new String[]{"u", "u2", "u3", "u4", "u5"});
model.physics("c2").field("dimensionless").component(1, "u");
model.physics("c2").field("dimensionless").component(1, "x_CH4");
model.physics("c2").field("dimensionless").component(2, "x_CO");
model.physics("c2").field("dimensionless").component(3, "x_CO2");
model.physics("c2").field("dimensionless").component(4, "x_H2");
model.physics("c2").field("dimensionless").component(5, "x_H2O");
model.physics("c2").field("dimensionless").component(new String[]{"x_CH4", "x_CO", "x_CO2", "x_H2"});
model.physics("c2").feature("cfeq1").setIndex("c", "-(x^2)*invA_11*c_TOT", 0, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "0", 0, 0);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_11*c_TOT", 0, 3);
model.physics("c2").prop("Units").set("SourceTermQuantity", "molarsurfaceflux");
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_21*c_TOT", 1, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_31*c_TOT", 2, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_41*c_TOT", 3, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_12*c_TOT", 4, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_12*c_TOT", 5, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "0", 5, 0);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_22*c_TOT", 5, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_32*c_TOT", 6, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_42*c_TOT", 7, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_13*c_TOT", 8, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_23*c_TOT", 9, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "0", 10, 0);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_33*c_TOT", 10, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_43*c_TOT", 11, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_14*c_TOT", 12, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_24*c_TOT", 13, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_34*c_TOT", 14, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_44*c_TOT", 15, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "0", 15, 0);
model.physics("c2").feature("cfeq1").setIndex("f", "-(y^2)*(r1_A+r3_A)*eps_I", 0);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r1_A-r2_A)*eps_I", 1);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r2_A+r3_A)*eps_I", 2);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(3*r1_A+r2_A+4*r3_A)*eps_I", 3);
model.physics("c2").feature("cfeq1").setIndex("da", "0", 0);
model.physics("c2").feature("cfeq1").setIndex("da", "0", 5);
model.physics("c2").feature("cfeq1").setIndex("da", "0", 10);
model.physics("c2").feature("cfeq1").setIndex("da", "0", 15);
model.physics("c2").feature("init1").set("x_CH4", "1e-6");
model.physics("c2").feature("init1").set("x_CO", "1e-6");
model.physics("c2").feature("init1").set("x_CO2", "1e-6");
model.physics("c2").feature("init1").set("x_H2", "1e-6");
model.physics("c2").feature().create("dir1", "DirichletBoundary", 1);
model.physics("c2").feature("dir1").selection().set(new int[]{3});
model.physics("c2").feature("dir1").setIndex("r", "x", 0);
model.physics("c2").feature("dir1").setIndex("r", "x_CH4_b", 0);
model.physics("c2").feature("dir1").setIndex("r", "x_CO_b", 1);
model.physics("c2").feature("dir1").setIndex("r", "x_CO2_b", 2);
model.physics("c2").feature("dir1").setIndex("r", "x_H2_b", 3);

model.variable().create("var6");
model.variable("var6").model("comp2");
model.variable("var6").label("Bulk parameters");
model.variable("var6").set("x_CH4_b", "0.1");
model.variable("var6").set("x_CO_b", "0.2");
model.variable("var6").set("x_CO2_b", "0.3");
model.variable("var6").set("x_H2_b", "0.4");

model.mesh("mesh2").autoMeshSize(4);
model.mesh("mesh2").automatic(false);
```

```
model.mesh("mesh2").automatic(true);
model.mesh("mesh2").automatic(false);
model.mesh("mesh2").feature("ftri1").create("dis1", "Distribution");
model.mesh("mesh2").feature("ftri1").feature("dis1").selection().set(new int[]{1, 4});
model.mesh("mesh2").feature("ftri1").feature("dis1").set("type", "predefined");
model.mesh("mesh2").feature("ftri1").feature("dis1").set("elemcount", "500");
model.mesh("mesh2").run("ftri1");
model.mesh("mesh2").feature().remove("ftri1");
model.mesh("mesh2").create("map1", "Map");
model.mesh("mesh2").feature("map1").create("dis1", "Distribution");
model.mesh("mesh2").feature("map1").feature("dis1").selection().set(new int[]{1, 4});
model.mesh("mesh2").feature("map1").feature("dis1").set("type", "predefined");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "500");
model.mesh("mesh2").run("map1");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "10");
model.mesh("mesh2").run("map1");
model.mesh("mesh2").feature("map1").feature("dis1").set("reverse", "on");
model.mesh("mesh2").run("map1");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "200");
model.mesh("mesh2").run("map1");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "15");
model.mesh("mesh2").run("map1");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "5");
model.mesh("mesh2").run("map1");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "250");
model.mesh("mesh2").run("map1");

model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_11*c_TOT/r_pellet^2", 0, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_21*c_TOT/r_pellet^2", 1, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_31*c_TOT/r_pellet^2", 2, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_41*c_TOT/r_pellet^2", 3, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_12*c_TOT/r_pellet^2", 4, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_22*c_TOT/r_pellet^2", 5, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_32*c_TOT/r_pellet^2", 6, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_42*c_TOT/r_pellet^2", 7, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_13*c_TOT/r_pellet^2", 8, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_23*c_TOT/r_pellet^2", 9, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_33*c_TOT/r_pellet^2", 10, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_43*c_TOT/r_pellet^2", 11, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_14*c_TOT/r_pellet^2", 12, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_24*c_TOT/r_pellet^2", 13, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_34*c_TOT/r_pellet^2", 14, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_44*c_TOT/r_pellet^2", 15, 3);
model.physics("c2").prop("Units").set("SourceTermQuantity", "reactionrate");
model.physics("c2").feature("cfeq1").setIndex("f", "-(y^2)*(r1_A+r3_A)*eps_I/r_pellet^2", 0);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r1_A-r2_A)*eps_I/r_pellet^2", 1);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r2_A+r3_A)*eps_I/r_pellet^2", 2);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(3*r1_A+r2_A+4*r3_A)*eps_I/r_pellet^2", 3);

model.study("std1").feature("stat").set("physselection", "tcs");
model.study("std1").feature("stat").set("activate", new String[]{"tcs", "off", "c", "on", "c2", "on"});
model.study("std1").feature("stat").set("physselection", "tcs");
model.study("std1").feature("stat").set("activate", new String[]{"tcs", "off", "c", "off", "c2", "on"});

model.sol().create("sol1");
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.result().create("pg1", 2);
model.result("pg1").set("data", "dset2");
model.result("pg1").create("surf1", "Surface");
model.result("pg1").feature("surf1").set("expr", "x_CH4");

model.baseSystem("cgs");

model.sol("sol1").study("std1");
```

```
model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.variable("var6").set("x_CH4_b", "x_CH4_0");
model.variable("var6").set("x_CO_b", "x_CO_0");
model.variable("var6").set("x_CO2_b", "x_CO2_0");
model.variable("var6").set("x_H2_b", "x_H2_0");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.geom("geom2").feature("sq1").set("size", "1[cm]");
model.geom("geom2").run("sq1");
model.geom("geom2").run();

model.mesh("mesh2").run("map1");

model.study("std1").feature("stat").set("useadvanceddisable", "off");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_11*c_TOT", 0, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_21*c_TOT", 1, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_31*c_TOT", 2, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_41*c_TOT", 3, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_12*c_TOT", 4, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_22*c_TOT", 5, 3);
```

```
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_32*c_TOT", 6, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_42*c_TOT", 7, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-(y^2)*invA_43*c_TOT", 11, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_11*c_TOT*(y^2)", 0, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_21*c_TOT*(y^2)", 1, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_31*c_TOT*(y^2)", 2, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_41*c_TOT*(y^2)", 3, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_12*c_TOT*(y^2)", 4, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_22*c_TOT*(y^2)", 5, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_32*c_TOT*(y^2)", 6, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_42*c_TOT*(y^2)", 7, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_13*c_TOT*(y^2)", 8, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_23*c_TOT*(y^2)", 9, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_33*c_TOT*(y^2)", 10, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_43*c_TOT*(y^2)", 11, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_14*c_TOT*(y^2)", 12, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_24*c_TOT*(y^2)", 13, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_34*c_TOT*(y^2)", 14, 3);
model.physics("c2").feature("cfeq1").setIndex("c", "-invA_44*c_TOT*(y^2)", 15, 3);
model.physics("c2").prop("Units").set("SourceTermQuantity", "molarsurfaceflux");
model.physics("c2").feature("cfeq1").setIndex("f", "-(y^2)*(r1_A+r3_A)*eps_I", 0);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r1_A-r2_A)*eps_I", 1);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r2_A+r3_A)*eps_I", 2);
model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(3*r1_A+r2_A+4*r3_A)*eps_I", 3);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.physics("c2").feature("init1").set("x_CH4", "x_CH4_0");
model.physics("c2").feature("init1").set("x_CO", "x_CO_0");
model.physics("c2").feature("init1").set("x_CO2", "x_CO2_0");
model.physics("c2").feature("init1").set("x_H2", "x_H2_0");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "dDef");
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").attach("std1");

model.param().set("T_in", "750[K]");

model.sol("sol1").feature("s1").feature("fc1").set("maxiter", "250");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").run();

model.param().set("T_in", "600[K]");
```

```
    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel", "MaxwellStefan", 0);
    model.physics("c2").prop("Units").set("SourceTermQuantity", "reactionrate");
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_11*c_TOT*(y^2)/(r_pellet^2)", 0, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_21*c_TOT*(y^2)/(r_pellet^2)", 1, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_31*c_TOT*(y^2)/(r_pellet^2)", 2, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_41*c_TOT*(y^2)/(r_pellet^2)", 3, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_12*c_TOT*(y^2)/(r_pellet^2)", 4, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_22*c_TOT*(y^2)/(r_pellet^2)", 5, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_32*c_TOT*(y^2)/(r_pellet^2)", 6, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_42*c_TOT*(y^2)/(r_pellet^2)", 7, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_13*c_TOT*(y^2)/(r_pellet^2)", 8, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_23*c_TOT*(y^2)/(r_pellet^2)", 9, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_33*c_TOT*(y^2)/(r_pellet^2)", 10, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_43*c_TOT*(y^2)/(r_pellet^2)", 11, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_14*c_TOT*(y^2)/(r_pellet^2)", 12, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_24*c_TOT*(y^2)/(r_pellet^2)", 13, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_34*c_TOT*(y^2)/(r_pellet^2)", 14, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_44*c_TOT*(y^2)/(r_pellet^2)", 15, 3);
    model.physics("c2").feature("cfeq1").setIndex("f", "-(y^2)*(r1_A+r3_A)*eps_I/(r_pellet^2)", 0);
    model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r1_A-r2_A)*eps_I/(r_pellet^2)", 1);
    model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r2_A+r3_A)*eps_I/(r_pellet^2)", 2);
    model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(3*r1_A+r2_A+4*r3_A)*eps_I/(r_pellet^2)", 3);

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.variable().create("var7");
    model.variable("var7").model("comp1");
    model.variable("var7").set("beta_0", "G*1");
    model.variable("var7").set("G", "1");
    model.variable("var7").set("beta_0", "G*(1-void_fraction)/(rho_gas_0*(2*r_pellet)*(void_fraction^3))");
    model.variable("var7").set("G", "1[kg/(m^2*s)]");
    model.variable("var7")
         .set("beta_0", "G*(1-void_fraction)/(rho_gas_0*(2*r_pellet)*(void_fraction^3))*(150*(1-
void_fraction)*gas_viscosity)");

    model.param().set("mu_gas", "3e-5[Pa*s]");
    model.param().descr("mu_gas", "Gas viscosity - Need to check accuracy");

    model.variable("var7")
         .set("beta_0", "G*(1-void_fraction)/(rho_gas_0*(2*r_pellet)*(void_fraction^3))*(150*(1-
void_fraction)*mu_gas/(2*r_pellet)+150*G)");

    return model;
  }

  public static Model run3(Model model) {
    model.variable("var7").set("G", "rho_gas_0*U_ave");

    model.physics("c").feature("cfeq1").setIndex("f", "Px+beta_0", 0);
    model.physics("c").prop("Units").set("SourceTermQuantity", "springconstantperarea");
    model.physics("c").prop("Units").set("DependentVariableQuantity", "pressure");
    model.physics("c").feature("cfeq1").setIndex("f", "px+beta_0*(p_in/p)*(T/T_in)*(FT/FT_0)", 0);
    model.physics("c").field("dimensionless").field("p");
    model.physics("c").field("dimensionless").component(1, "p");
    model.physics("c").prop("Units").set("SourceTermQuantity", "none");
    model.physics("c").prop("Units").set("CustomSourceTermUnit", "N/m^3");

    model.label("Pellet 1D and bed reactor - 1st try-rev.A.mph");

    model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "30");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "0.5");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "5");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "100");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "3");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "10");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "15");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").create("dis2", "Distribution");
    model.mesh("mesh2").feature("map1").feature("dis2").selection().set(new int[]{2, 3});
```

```
    model.mesh("mesh2").feature("map1").feature("dis2").set("numelem", "120");
    model.mesh("mesh2").run("map1");
    model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "80");
    model.mesh("mesh2").run("map1");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg1").feature("surf1").set("expr", "j_CH4");
    model.result("pg1").run();

    model.variable("var1").set("j_CH4", "c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4y");
    model.variable("var1").set("j_CO", "c_TOT*(invA_12+invA_22+invA_32+invA_42)*x_COy");
    model.variable("var1").set("j_CO2", "c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2y");
    model.variable("var1").set("j_H2", "c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y");
    model.variable("var1")
         .set("j_H2O", "-
((DKeff_H2O/DKeff_CH4)*(c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4y)+(DKeff_H2O/DKeff_CO)*(c_TOT*(invA_12+in
vA_22+invA_32+invA_42)*x_COy)+(DKeff_H2O/DKeff_CO2)*(c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2y)+(DKeff_H2O
/DKeff_H2)*(c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y))");

    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").run();
    model.result("pg1").feature("surf1").set("expr", "j_H2");
    model.result("pg1").run();

    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 1);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 2);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 3);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 4);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 0);
    model.physics("tcs").feature("cdm1").setIndex("M", "M_CH4", 0);
    model.physics("tcs").feature("cdm1").setIndex("M", "M_CO", 1);
    model.physics("tcs").feature("cdm1").setIndex("M", "M_CO2", 2);
    model.physics("tcs").feature("cdm1").setIndex("M", "M_H2", 3);
    model.physics("tcs").feature("cdm1").setIndex("M", "M_H2O", 4);
    model.physics("tcs").feature("cdm1")
         .set("Dik", new String[]{"1", "comp2.D_CH4.CO", "comp2.D_CH4.CO2", "comp2.D_CH4.H2",
"comp2.D_CH4.H2O", "comp2.D_CH4.CO", "1", "comp2.D_CO.CO2", "comp2.D_CO.H2", "comp2.D_CO.H2O",
         "comp2.D_CH4.CO2", "comp2.D_CO.CO2", "1", "comp2.D_CO2.H2", "comp2.D_CO2.H2O", "comp2.D_CH4.H2",
"comp2.D_CO.H2", "comp2.D_CO2.H2", "1", "comp2.D_H2.H2O",
         "comp2.D_CH4.H2O", "comp2.D_CO.H2O", "comp2.D_CO2.H2O", "comp2.D_H2.H2O", "1"});
    model.physics("tcs").feature("cdm1").set("u", new String[]{"u", "0", "0"});

    model.param().set("u_0", "3[m/s]");
    model.param().descr("u_0", "Entrance fluid velocity");
    model.param().descr("u_0", "Entrance gas velocity");
    model.param().rename("U_ave", "u_ave");
    model.param().set("u_ave", "1.0[m/s]", "Average velocity of the fluid inside the pipe");

    model.variable("var7").set("G", "rho_gas_0*u_0");

    model.result("pg1").run();

    model.variable("var1").set("j_CH4", "c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4y*(y^2/r_pellet^2)");
    model.variable("var1").set("j_CO", "c_TOT*(invA_12+invA_22+invA_32+invA_42)*x_COy*(y^2/r_pellet^2)");
    model.variable("var1").set("j_CO2", "c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2y*(y^2/r_pellet^2)");
    model.variable("var1").set("j_H2", "c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y*(y^2/r_pellet^2)");
    model.variable("var1")
         .set("j_H2O", "-
(((DKeff_H2O/DKeff_CH4)*(c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4y)+(DKeff_H2O/DKeff_CO)*(c_TOT*(invA_12+i
nvA_22+invA_32+invA_42)*x_COy)+(DKeff_H2O/DKeff_CO2)*(c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2y)+(DKeff_H2
O/DKeff_H2)*(c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y)))*(y^2/r_pellet^2)");

    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.physics("tcs").feature("cdm1").setIndex("minput_temperature", "T", 0);
    model.physics("tcs").feature("cdm1").setIndex("minput_pressure", "p", 0);

    model.variable("var7").set("u", "u_0*(cb_TOT/c_TOT_0)*(p_in/p)");
    model.variable("var1").set("c_TOT", "comp1.p/(R_const*T)");
    model.variable("var7").set("cb_TOT", "1");
    model.variable("var7").set("cb_CH4", "1");
    model.variable("var7")
         .set("M_mean", "1/((wb_CH4/M_CH4)+(wb_CO/M_CO)+(wb_CO2/M_CO2)+(wb_H2/M_H2)+(wb_H2O/M_H2O))", "Mean
molecular mass");
```

```
model.variable("var7").set("cb_TOT", "p/(R_const*T)");
model.variable("var7").set("xb_CH4", "(wb_CH4/M_CH4)*M_mean");
model.variable("var7").set("cb_CH4", "cb_TOT*xb_CH4");
model.variable("var7").set("xb_CO", "(wb_CO/M_CO)*M_mean");
model.variable("var7").set("xb_CO2", "(wb_CO2/M_CO2)*M_mean");
model.variable("var7").set("xb_H2", "(wb_H2/M_H2)*M_mean");
model.variable("var7").set("xb_H2O", "(wb_H2O/M_H2O)*M_mean");
model.variable("var7").set("cb_CO2", "cb_TOT*xb_CO2");
model.variable("var7").set("cb_CO", "cb_TOT*xb_CO");
model.variable("var7").set("cb_H2", "cb_TOT*xb_H2");
model.variable("var7").set("cb_H2O", "cb_TOT*xb_H2O");
model.variable("var7").set("F_0", "Ctot_feed*u_feed*Ra");
model.variable("var7").rename("F_0", "FT_0");
model.variable("var7").set("FT_0", "cb_TOT*u*Ac_tube");
model.variable("var7").set("FT", "cb_TOT*u*Ac_tube");
model.variable("var7").set("FT_0", "c_TOT_0*u_0*Ac_tube");

model.mesh("mesh1").automatic(false);
model.mesh("mesh1").feature("size").set("hauto", "3");
model.mesh("mesh1").create("dis1", "Distribution");
model.mesh("mesh1").feature("dis1").selection().set(new int[]{1});
model.mesh("mesh1").feature("dis1").set("type", "predefined");
model.mesh("mesh1").feature("dis1").set("elemcount", "120");
model.mesh("mesh1").run();
model.mesh("mesh1").feature("dis1").set("type", "number");
model.mesh("mesh1").feature("dis1").set("numelem", "120");
model.mesh("mesh1").run("dis1");
model.mesh("mesh1").feature("dis1").set("type", "explicit");
model.mesh("mesh1").feature().remove("dis1");
model.mesh("mesh1").feature("size").set("custom", "on");
model.mesh("mesh1").feature("size").set("hmax", "0.008");
model.mesh("mesh1").run("size");
model.mesh("mesh1").run();
model.mesh("mesh2").feature("map1").feature("dis2").set("numelem", "125");
model.mesh("mesh2").run();
model.mesh("mesh2").run("map1");
model.mesh("mesh2").run();

model.variable("var7").descr("u", "Check LATER");

model.physics("tcs").feature("init1").setIndex("w0", "w_CH4_0", 0);
model.physics("tcs").feature("init1").setIndex("w0", "w_CO_0", 1);
model.physics("tcs").feature("init1").setIndex("w0", "w_CO2_0", 2);
model.physics("tcs").feature("init1").setIndex("w0", "w_H2_0", 3);
model.physics("tcs").feature().create("in1", "Inflow", 0);
model.physics("tcs").feature("in1").selection().set(new int[]{1});
model.physics("tcs").feature("in1").setIndex("w0", "w_CO2_0", 2);
model.physics("tcs").feature("in1").setIndex("w0", "w_CO_0", 1);
model.physics("tcs").feature("in1").setIndex("w0", "w_CH4_0", 0);
model.physics("tcs").feature("in1").setIndex("w0", "w_H2_0", 3);
model.physics("tcs").feature().create("out1", "Outflow", 0);
model.physics("tcs").feature("out1").selection().set(new int[]{2});
model.physics("c").feature("init1").set("p", "p_in");
model.physics("c").feature().create("dir1", "DirichletBoundary", 0);
model.physics("c").feature("dir1").selection().set(new int[]{1});
model.physics("c").feature("dir1").setIndex("r", "p_in", 0);

model.study("std1").feature("stat").set("physselection", "tcs");
model.study("std1").feature("stat").set("activate", new String[]{"tcs", "off", "c", "on", "c2", "on"});
model.study("std1").feature("stat").set("physselection", "tcs");
model.study("std1").feature("stat").set("activate", new String[]{"tcs", "on", "c", "on", "c2", "on"});
model.study("std1").feature("stat").set("physselection", "tcs");
model.study("std1").feature("stat").set("activate", new String[]{"tcs", "on", "c", "on", "c2", "off"});

model.geom("geom1").feature("i1").set("p2", "h_tube");
model.geom("geom1").runPre("fin");
model.geom("geom1").run();

model.mesh("mesh1").feature("size").set("hmax", "0.08");
model.mesh("mesh1").run();

model.cpl().create("genext1", "GeneralExtrusion", "geom2");
model.cpl("genext1").set("usesrcmap", "on");
model.cpl("genext1").selection().geom("geom2", 1);
model.cpl("genext1").selection().set(new int[]{3});
model.cpl("genext1").set("dstmap", new String[]{"x", ""});
model.cpl("genext1").set("srcmap", new String[]{"x", ""});
model.cpl("genext1").set("dstmap", new String[]{"x/1000", ""});

model.variable("var7").set("rb_CH4", "(1-void_fraction)*(3/r_pellet)*genext1(comp2.j_CH4)");
model.variable("var7").set("rb_CO", "(1-void_fraction)*(3/r_pellet)*genext1(comp2.j_CO)");
```

```
model.variable("var7").set("rb_CO2", "(1-void_fraction)*(3/r_pellet)*genext1(comp2.j_CO2)");
model.variable("var7").set("rb_H2", "(1-void_fraction)*(3/r_pellet)*genext1(comp2.j_H2)");
model.variable("var7").set("rb_H2O", "(1-void_fraction)*(3/r_pellet)*genext1(comp2.j_H2O)");

model.physics("tcs").feature().create("reac1", "Reactions", 1);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CH4", 0);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CO", 1);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CH4*M_CH4", 0);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CO*M_CO", 1);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CO2*M_CO2", 2);
model.physics("tcs").feature("reac1").setIndex("R", "rb_H2*M_H2", 3);
model.physics("tcs").feature("reac1").selection().set(new int[]{1});

model.study("std1").feature("stat").set("physselection", "tcs");
model.study("std1").feature("stat").set("activate", new String[]{"tcs", "on", "c", "on", "c2", "on"});

model.sol("sol1").clearSolution();
model.sol().remove("sol1");
model.sol().create("sol1");
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result().create("pg1", 1);
model.result("pg1").set("data", "dset1");
model.result("pg1").create("lngr1", "LineGraph");
model.result("pg1").feature("lngr1").set("xdata", "expr");
model.result("pg1").feature("lngr1").set("xdataexpr", "x");
model.result("pg1").feature("lngr1").selection().all();
model.result("pg1").feature("lngr1").set("expr", "wb_CH4");
model.result("pg1").feature("lngr1").set("descr", "Mass fraction");
model.result("pg1").label("Mass Fraction (tcs)");
model.result().create("pg2", 1);
model.result("pg2").set("data", "dset1");
model.result("pg2").create("lngr1", "LineGraph");
model.result("pg2").feature("lngr1").set("xdata", "expr");
model.result("pg2").feature("lngr1").set("xdataexpr", "x");
model.result("pg2").feature("lngr1").selection().all();
model.result("pg2").feature("lngr1").set("expr", "p");
model.result().create("pg3", 2);
model.result("pg3").set("data", "dset2");
model.result("pg3").create("surf1", "Surface");
model.result("pg3").feature("surf1").set("expr", "x_CH4");

model.variable("var7").set("rb_CH4", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.j_CH4)");
model.variable("var7").set("rb_CO", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.j_CO)");
model.variable("var7").set("rb_CO2", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.j_CO2)");
model.variable("var7").set("rb_H2", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.j_H2)");
model.variable("var7").set("rb_H2O", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.j_H2O)");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");
```

```
model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.variable("var1").set("c_TOT", "p_in/(R_const*T)");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.cpl().create("genext2", "GeneralExtrusion", "geom1");
model.cpl("genext2").set("usesrcmap", "on");
model.cpl("genext2").selection().all();
model.cpl("genext2").set("dstmap", new String[]{"x*1000"});

model.variable("var6").set("x_CH4_b", "comp1.genext2(comp1.xb_CH4)");
model.variable("var6").set("x_CO_b", "comp1.genext2(comp1.xb_CO)");
model.variable("var6").set("x_CO2_b", "comp1.genext2(comp1.xb_CO2)");
model.variable("var6").set("x_H2_b", "comp1.genext2(comp1.xb_H2)");
model.variable("var1").set("c_TOT", "comp1.p/(R_const*T)");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
```

```
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.variable("var1").set("c_TOT", "root.comp1.p/(R_const*T)");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.variable("var1").set("c_TOT", "p_in/(R_const*T)");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
```

```
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.baseSystem("SI");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.baseSystem("cgs");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.variable("var7").set("u", "u_0*(cb_TOT/c_TOT_0)*(p_in/p_out)");
model.variable("var7").set("cb_TOT", "p_out/(R_const*T)");

model.physics("tcs").feature("cdm1").setIndex("minput_pressure", "p_out", 0);
```

```
  model.study("std1").feature("stat").set("physselection", "tcs");
  model.study("std1").feature("stat").set("activate", new String[]{"tcs", "on", "c", "off", "c2", "on"});

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");
  model.sol("sol1").runAll();

  model.result("pg1").run();
  model.result("pg2").run();
  model.result("pg3").run();
  model.result("pg2").run();
  model.result("pg2").feature("lngr1").set("expr", "cb_CH4");
  model.result("pg2").run();
  model.result("pg2").feature("lngr1").set("expr", "xb_CH4");
  model.result("pg2").run();

  model.param().set("T_in", "800[K]");

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");

  return model;
}

public static Model run4(Model model) {
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
```

```
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg2").run();
model.result("pg1").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_H2");
model.result("pg3").run();

model.param().set("u_0", "1[m/s]");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "c_H2");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "c_H2O");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("colortable", "RainbowLight");
model.result("pg3").feature("surf1").set("expr", "j_H2O");
model.result("pg3").run();
model.result("pg1").run();

model.param().set("u_0", "0.01[m/s]");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
```

```
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.param().set("u_0", "0.05[m/s]");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_H2O");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_H2");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("data", "dset2");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("data", "parent");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr2", "lngr1");
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr3", "lngr2");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature("lngr2").set("expr", "wb_CO");
model.result("pg1").run();
model.result("pg1").feature("lngr3").set("expr", "wb_CO2");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature().duplicate("lngr4", "lngr3");
model.result("pg1").run();
model.result("pg1").feature("lngr4").set("expr", "wb_H2");
model.result("pg1").run();
model.result("pg1").feature("lngr4").set("legend", "on");
model.result("pg1").feature("lngr4").set("legendmethod", "manual");
```

```
model.result("pg1").feature("lngr4").setIndex("legends", "w_H2", 0);
model.result("pg1").feature().duplicate("lngr5", "lngr4");
model.result("pg1").run();
model.result("pg1").feature("lngr5").set("expr", "wb_H2O");
model.result("pg1").feature("lngr5").setIndex("legends", "w_H2O", 0);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature("lngr1").set("legend", "off");
model.result("pg1").feature("lngr1").set("legendmethod", "manual");
model.result("pg1").feature("lngr1").setIndex("legends", "w_CH4", 0);
model.result("pg1").feature("lngr1").set("legend", "on");
model.result("pg1").run();
model.result("pg1").feature("lngr2").set("legend", "on");
model.result("pg1").feature("lngr2").set("legendmethod", "manual");
model.result("pg1").feature("lngr2").setIndex("legends", "w_CO", 0);
model.result("pg1").run();
model.result("pg1").feature("lngr3").set("legend", "on");
model.result("pg1").feature("lngr3").set("legendmethod", "manual");
model.result("pg1").feature("lngr3").setIndex("legends", "w_CO2", 0);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").set("legendpos", "middleright");
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").feature("lngr1").set("legend", "on");
model.result("pg2").feature("lngr1").set("legendmethod", "manual");
model.result("pg2").feature("lngr1").setIndex("legends", "xb_CH4", 0);
model.result("pg2").feature().duplicate("lngr2", "lngr1");
model.result("pg2").run();
model.result("pg2").feature("lngr2").set("expr", "xb_CO");
model.result("pg2").feature("lngr2").setIndex("legends", "xb_CO", 0);
model.result("pg2").feature().duplicate("lngr3", "lngr2");
model.result("pg2").run();
model.result("pg2").feature("lngr3").setIndex("legends", "xb_CO2", 0);
model.result("pg2").feature("lngr3").set("expr", "xb_CO2");
model.result("pg2").run();
model.result("pg2").feature().duplicate("lngr4", "lngr3");
model.result("pg2").run();
model.result("pg2").feature("lngr4").set("expr", "xb_H2");
model.result("pg2").feature("lngr4").setIndex("legends", "xb_H2", 0);
model.result("pg2").feature().duplicate("lngr5", "lngr4");
model.result("pg2").run();
model.result("pg2").feature("lngr5").set("expr", "xb_H2O");
model.result("pg2").feature("lngr5").setIndex("legends", "xb_H2O", 0);
model.result("pg2").run();
model.result().duplicate("pg4", "pg2");
model.result("pg4").run();
model.result().move("pg4", 2);
model.result("pg4").run();
model.result("pg4").feature("lngr1").set("expr", "cb_CH4");
model.result("pg4").feature("lngr1").setIndex("legends", "cb_CH4", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr2").set("expr", "cb_CO");
model.result("pg4").feature("lngr2").setIndex("legends", "cb_CO", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr3").set("expr", "cb_CO2");
model.result("pg4").feature("lngr3").setIndex("legends", "cb_CO2", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr4").set("expr", "cb_H2");
model.result("pg4").feature("lngr4").setIndex("legends", "cb_H2", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr5").set("expr", "cb_H2O");
model.result("pg4").feature("lngr5").setIndex("legends", "cb_H2O", 0);
model.result("pg4").run();
model.result("pg4").run();
model.result().duplicate("pg5", "pg4");
model.result("pg5").run();
model.result().move("pg5", 3);
model.result("pg5").run();
model.result("pg5").feature("lngr1").set("expr", "rb_CH4");
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").feature("lngr2").set("expr", "rb_CO");
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").feature("lngr3").set("expr", "rb_CO2");
model.result("pg5").run();
model.result("pg5").run();
```

```
model.result("pg5").feature("lngr4").set("expr", "rb_H2");
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").feature("lngr5").set("expr", "rb_H2O");
model.result("pg5").run();
model.result("pg5").feature("lngr5").setIndex("legends", "rb_H2O", 0);
model.result("pg5").run();
model.result("pg5").feature("lngr4").setIndex("legends", "rb_H2", 0);
model.result("pg5").run();
model.result("pg5").feature("lngr3").setIndex("legends", "rb_CO2", 0);
model.result("pg5").run();
model.result("pg5").feature("lngr2").setIndex("legends", "rb_CO", 0);
model.result("pg5").run();
model.result("pg5").feature("lngr1").setIndex("legends", "rb_CH4", 0);
model.result("pg5").run();

model.label("Pellet 1D and bed reactor - 1st try-rev.A1.mph");

model.result("pg5").run();
model.result("pg4").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();

model.study("std1").feature("stat").set("useparam", "on");
model.study("std1").feature("stat").setIndex("pname", "Bed_Parameters_Geometry", 0);
model.study("std1").feature("stat").setIndex("plistarr", "", 0);
model.study("std1").feature("stat").setIndex("punit", "", 0);
model.study("std1").feature("stat").setIndex("pname", "Bed_Parameters_Geometry", 0);
model.study("std1").feature("stat").setIndex("plistarr", "", 0);
model.study("std1").feature("stat").setIndex("punit", "", 0);
model.study("std1").feature("stat").setIndex("pname", "T_in", 0);
model.study("std1").feature("stat").setIndex("plistarr", "range(800,100,1100)", 0);
model.study("std1").feature("stat").setIndex("punit", "K", 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").setIndex("looplevelinput", "manualindices", 0);
model.result("pg1").setIndex("looplevelindices", "3", 0);
model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").setIndex("looplevelinput", "manualindices", 0);
model.result("pg2").setIndex("looplevelindices", "3", 0);
model.result("pg2").run();
```

```
model.label("Pellet 1D and bed reactor - 1st try-rev.A1.mph");

model.param().set("rho_bulk", "rho_catalyst*(1-void_fraction)");
model.param().set("rho_catalyst", "2355.2[kg/m^3]");
model.param().set("eps_V", "rho_catalyst*0.2508[cm^3/g]");
model.param().set("eps_I", "A_pores*rho_catalyst");
model.param().set("eps_A", "eps_V/(3*Tau)");
model.param().set("Tau", "3.54");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg3").run();
model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").setIndex("looplevelindices", "1", 0);
model.result("pg2").run();
model.result("pg2").setIndex("looplevelinput", "last", 0);
model.result("pg2").run();
model.result("pg2").setIndex("looplevelinput", "first", 0);
model.result("pg2").run();

model.variable("var5").set("DK_CH4", "(4*(4.86e-5[cm])/3)*sqrt(R_const*T/(2*pi*M_CH4))");
model.variable("var5").set("DK_CO", "(4*(4.86e-5[cm])/3)*sqrt(R_const*T/(2*pi*M_CO))");
model.variable("var5").set("DK_CO2", "(4*(4.86e-5[cm])/3)*sqrt(R_const*T/(2*pi*M_CO2))");
model.variable("var5").set("DK_H2", "(4*(4.86e-5[cm])/3)*sqrt(R_const*T/(2*pi*M_H2))");
model.variable("var5").set("DK_H2O", "(4*(4.86e-5[cm])/3)*sqrt(R_const*T/(2*pi*M_H2O))");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
```

```
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.study("std1").feature("stat").setIndex("plistarr", "range(400,100,800)", 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result("pg1").run();
model.result("pg1").set("solnumindices", "4");
model.result("pg1").run();
model.result("pg1").set("solnumindices", "3");
model.result("pg1").run();
model.result("pg1").set("solnumindices", "2");
model.result("pg1").run();
model.result("pg1").set("solnumindices", "3");
model.result("pg1").run();
model.result("pg2").run();
model.result("pg3").run();

model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))");
model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))");
model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))");
model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))");
```

```
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 1);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 2);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 3);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 4);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result("pg1").run();
model.result("pg1").set("innerinput", "all");
model.result("pg1").run();
model.result("pg1").set("innerinput", "manual");
model.result("pg1").set("solnum", new String[]{"3"});
model.result("pg1").run();
model.result("pg1").set("solnum", new String[]{"2"});
model.result("pg1").run();
model.result("pg1").set("solnum", new String[]{"1"});
model.result("pg1").run();
model.result("pg1").set("solnum", new String[]{"3"});
model.result("pg1").run();
model.result("pg1").set("solnum", new String[]{"4"});
model.result("pg1").run();
model.result("pg1").set("solnum", new String[]{"3"});
model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").set("innerinput", "manual");
model.result("pg2").set("solnum", new String[]{"3"});
model.result("pg2").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg5").run();
model.result("pg5").set("innerinput", "manual");
model.result("pg5").set("solnum", new String[]{"3"});
model.result("pg5").run();

model.mesh("mesh2").feature("map1").feature("dis2").set("numelem", "250");
model.mesh("mesh2").run();
model.mesh("mesh1").feature("size").set("hmax", "0.04");
model.mesh("mesh1").run();

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
```

```
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");

  model.result("pg3").run();
  model.result("pg1").run();
  model.result("pg1").set("solnum", new String[]{"4"});
  model.result("pg1").run();
  model.result("pg1").set("solnum", new String[]{"3"});
  model.result("pg2").run();

  model.sol("sol1").feature("s1").feature("d1").set("linsolver", "pardiso");

  model.result("pg3").run();
  model.result("pg2").run();
  model.result("pg2").set("solnum", new String[]{"4"});
  model.result("pg2").run();

  model.sol("sol1").feature("s1").feature("d1").set("linsolver", "mumps");

  model.study("std1").feature("stat").setIndex("plistarr", "range(400,50,900)", 0);
  model.study("std1").feature("stat").set("pcontinuationmode", "no");
  model.study("std1").feature("stat").set("preusesol", "yes");

  return model;
}

public static Model run5(Model model) {
  model.study("std1").feature("stat").setIndex("plistarr", "range(400,100,900)", 0);

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
```

```
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.result("pg1").run();
    model.result("pg1").set("solnum", new String[]{"4"});
    model.result("pg1").run();
    model.result("pg1").set("solnum", new String[]{"3"});
    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg2").set("solnum", new String[]{"3"});
    model.result("pg2").run();
    model.result("pg4").run();
    model.result("pg5").run();
    model.result("pg5").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").set("solnum", "3");
    model.result("pg3").run();

    model.label("Pellet 1D and bed reactor - 1st try-rev.A2.mph");

    model.result("pg3").run();

    model.param().set("T_in", "600[K]");
    model.param().set("r_pellet", "5[mm]");

    model.geom("geom2").run("sq1");
    model.geom("geom2").create("r1", "Rectangle");
    model.geom("geom2").feature().remove("sq1");
    model.geom("geom2").run();
    model.geom("geom2").feature("r1").set("size", new String[]{"h_tube", "r_pellet"});
    model.geom("geom2").runPre("fin");
    model.geom("geom2").run();

    model.physics("c2").prop("Units").set("SourceTermQuantity", "molarsurfaceflux");

    model.variable("var1").rename("x_H2O", "x_H2");
    model.variable("var1").set("x_H2", "1-(x_CH4+x_CO+x_CO2+x_H2)", "Molar fraction, H2O");
    model.variable("var1").set("x_H2", "1-(x_CH4+x_CO+x_CO2+x_H2O)");

    model.physics("c2").field("dimensionless").component(4, "x_H2O");

    model.variable("var1").remove("w_H2");
    model.variable("var1").rename("w_H2O", "w_H2");
    model.variable("var1")
         .set("w_H2", "c_H2O*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
H2O");
    model.variable("var1")
         .set("w_H2O", "c_H2*M_CH4/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)", "Mass fraction,
H2");
    model.variable("var1").set("w_H2O",
"c_H2O*M_H2O/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)");
    model.variable("var1").set("w_CO2",
"c_CO2*M_CO2/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)");
    model.variable("var1").set("w_CO", "c_CO*M_CO/(c_CH4*M_CH4+c_H2O*M_H2O+c_H2*M_H2+c_CO2*M_CO2+c_CO*M_CO)");
    model.variable("var1").set("w_H2", "1-(w_CH4+w_CO+w_CO2+w_H2O)");
    model.variable("var1").set("w_CH4",
"x_CH4*M_CH4/(x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O)");
    model.variable("var1").set("w_CO", "x_CO*M_CO/(x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O)");
    model.variable("var1").set("w_CO2",
"x_CO2*M_CO2/(x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O)");
    model.variable("var1").set("w_H2O",
"x_H2O*M_H2O/(x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O)");
    model.variable("var1")
         .set("rho_gas", "p_TOT*(x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O)/(R_const*T)");
    model.variable("var1").set("j_CH4", "c_TOT*(invA_11*x_CH4x+invA_12*x_COx+invA_13*x_CO2x+invA_14*x_H2Ox)");
    model.variable("var1").set("j_CO", "c_TOT*(invA_21*x_CH4x+invA_22*x_COx+invA_23*x_CO2x+invA_24*x_H2Ox)");
    model.variable("var1").set("j_CO2", "c_TOT*(invA_31*x_CH4x+invA_32*x_COx+invA_33*x_CO2x+invA_34*x_H2Ox)");
    model.variable("var1").remove("j_H2");
    model.variable("var1").rename("j_H2O", "j_H2");
    model.variable("var1")
```

```
        .set("j_H2", "-
(((DKeff_H2O/DKeff_CH4)*(c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4y)+(DKeff_H2O/DKeff_CO)*(c_TOT*(invA_12+i
nvA_22+invA_32+invA_42)*x_COy)+(DKeff_H2O/DKeff_CO2)*(c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2y)+(DKeff_H2
O/DKeff_H2)*(c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y)))*(y^2/r_pellet^2)", "Molar flux, H2O");
    model.variable("var1")
        .set("j_H2O", "c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y*(y^2/r_pellet^2)", "Molar flux, H2");
    model.variable("var1").rename("j_CH4", "N_CH4");
    model.variable("var1")
        .set("N_CH4", "c_TOT*(invA_11*x_CH4x+invA_12*x_COx+invA_13*x_CO2x+invA_14*x_H2Ox)", "Molar flux,
CH4");
    model.variable("var1").rename("j_CO", "N_CO");
    model.variable("var1")
        .set("N_CO", "c_TOT*(invA_21*x_CH4x+invA_22*x_COx+invA_23*x_CO2x+invA_24*x_H2Ox)", "Molar flux, CO");
    model.variable("var1").rename("j_CO2", "N_CO2");
    model.variable("var1")
        .set("N_CO2", "c_TOT*(invA_31*x_CH4x+invA_32*x_COx+invA_33*x_CO2x+invA_34*x_H2Ox)", "Molar flux,
CO2");
    model.variable("var1").rename("j_H2O", "N_H2O");
    model.variable("var1")
        .set("N_H2O", "c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y*(y^2/r_pellet^2)", "Molar flux, H2");
    model.variable("var1").rename("j_H2", "N_H2");
    model.variable("var1")
        .set("N_H2", "-
(((DKeff_H2O/DKeff_CH4)*(c_TOT*(invA_11+invA_21+invA_31+invA_41)*x_CH4y)+(DKeff_H2O/DKeff_CO)*(c_TOT*(invA_12+i
nvA_22+invA_32+invA_42)*x_COy)+(DKeff_H2O/DKeff_CO2)*(c_TOT*(invA_13+invA_23+invA_33+invA_43)*x_CO2y)+(DKeff_H2
O/DKeff_H2)*(c_TOT*(invA_14+invA_24+invA_34+invA_44)*x_H2y)))*(y^2/r_pellet^2)", "Molar flux, H2O");
    model.variable("var1").set("N_H2O", "c_TOT*(invA_41*x_CH4x+invA_42*x_COx+invA_43*x_CO2x+invA_44*x_H2Ox)");
    model.variable("var1").set("N_CH4", "c_TOT*(invA_11*x_CH4y+invA_12*x_COy+invA_13*x_CO2y+invA_14*x_H2Oy)");
    model.variable("var1").set("N_CO", "c_TOT*(invA_21*x_CH4y+invA_22*x_COy+invA_23*x_CO2y+invA_24*x_H2Oy)");
    model.variable("var1").set("N_CO2", "c_TOT*(invA_31*x_CH4y+invA_32*x_COy+invA_33*x_CO2y+invA_34*x_H2Oy)");
    model.variable("var1").set("N_H2O", "c_TOT*(invA_41*x_CH4y+invA_42*x_COy+invA_43*x_CO2y+invA_44*x_H2Oy)");
    model.variable("var1")
        .set("N_H2", "-
((DKeff_H2/DKeff_CH4)*N_CH4+(DKeff_H2/DKeff_CO)*N_CO+(DKeff_H2/DKeff_CO2)*N_CO2+(DKeff_H2/DKeff_H2O)*N_H2O)");

    model.label("Pellet 1D and bed reactor - 2nd try-rev.0.mph");

    model.variable("var1").set("J_CH4", "M_CH4*N_CH4");
    model.variable("var1").set("J_CO2", "M_CO*N_CO");
    model.variable("var1").rename("J_CO2", "J_CO");
    model.variable("var1").set("J_CO", "M_CO*N_CO");
    model.variable("var1").set("J_CO2", "M_CO2*N_CO2");
    model.variable("var1").set("J_H2O", "M_H2O*N_H2O");
    model.variable("var1").set("J_H2", "M_H2*N_H2");
    model.variable("var1").descr("J_CH4", "Mass flux, CH4");
    model.variable("var1").descr("J_CO", "Mass flux, CO");
    model.variable("var1").descr("J_CO2", "Mass flux, CO2");
    model.variable("var1").descr("J_H2O", "Mass flux, H2O");
    model.variable("var1").descr("J_H2", "Mass flux, H2");
    model.variable("var2").set("r1_A", "r1*rho_catalyst/eps_I");
    model.variable("var2").set("r2_A", "r2*rho_catalyst/eps_I");
    model.variable("var2").set("r3_A", "r3*rho_catalyst/eps_I");

    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_11*c_TOT*(y^2)", 0, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_21*c_TOT*(y^2)", 1, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_31*c_TOT*(y^2)", 2, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_41*c_TOT*(y^2)", 3, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_12*c_TOT*(y^2)", 4, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_22*c_TOT*(y^2)", 5, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_32*c_TOT*(y^2)", 6, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_42*c_TOT*(y^2)", 7, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_13*TOc_T*(y^2)/(r_pellet^2)", 8, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_13*c_TOT*(y^2)", 8, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_23*c_TOT*(y^2)", 9, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_33*c_TOT*(y^2)", 10, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_43*c_TOT*(y^2)", 11, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_14*c_TOT*(y^2)", 12, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_24*c_TOT*(y^2)", 13, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_34*c_TOT*(y^2)", 14, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_44*c_TOT*(y^2)", 15, 3);
    model.physics("c2").feature("cfeq1").setIndex("f", "-(y^2)*(r1_A+r3_A)*eps_I", 0);
    model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r1_A-r2_A)*eps_I", 1);
    model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(r2_A+r3_A)*eps_I", 2);
    model.physics("c2").feature("cfeq1").setIndex("f", "(y^2)*(-r1_A-r2_A-2*r3_A)*eps_I", 3);
    model.physics("c2").feature("init1").set("x_H2O", "x_H2O_0");
    model.physics("c2").feature("dir1").setIndex("weakConstraints", "1", 0);
    model.physics("c2").feature("dir1").setIndex("weakConstraints", "0", 0);
    model.physics("c2").feature("dir1").setIndex("constraintMethod", "nodal", 0);
    model.physics("c2").feature("dir1").setIndex("constraintMethod", "elemental", 0);
    model.physics("tcs").prop("SpeciesProperties").setIndex("FromMassConstraint", "4", 0);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 1);
```

```
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 2);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 3);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 4);
model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 0);
model.physics("tcs").prop("Convection").setIndex("Convection", "0", 0);
model.physics("tcs").prop("Convection").setIndex("Convection", "1", 0);
model.physics("tcs").feature("init1").setIndex("w0", "w_H2O_0", 4);
model.physics("tcs").feature("in1").setIndex("w0", "w_H2O_0", 4);

model.variable("var7").set("cb_TOT", "p_in/(R_const*T)");
model.variable("var7").set("rb_CH4", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_CH4)");
model.variable("var7").set("rb_CO", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_CO)");
model.variable("var7").set("rb_CO2", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_CO2)");
model.variable("var7").set("rb_H2", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_H2)");
model.variable("var7").set("rb_H2O", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_H2O)");

model.physics("tcs").feature("reac1").setIndex("R", "rb_CH4", 0);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CO", 1);
model.physics("tcs").feature("reac1").setIndex("R", "rb_CO2", 2);
model.physics("tcs").feature("reac1").setIndex("R", "rb_H2O", 4);

model.geom("geom2").feature("r1").set("size", new String[]{"h_tube/1000", "r_pellet"});
model.geom("geom2").runPre("fin");
model.geom("geom2").run();

model.cpl("genext1").set("usenan", "off");

model.mesh("mesh1").run();
model.mesh("mesh1").run();
model.mesh("mesh1").run();
model.mesh("mesh1").run("size");
model.mesh("mesh1").run();
model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "150");
model.mesh("mesh2").run();

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
model.result("pg1").setIndex("looplevel", "6", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "3", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "1", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
```

```
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "6", 0);
model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").setIndex("looplevel", "4", 0);
model.result("pg2").run();
model.result("pg2").setIndex("looplevel", "6", 0);
model.result("pg2").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "5", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "6", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "1", 0);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "4", 0);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_H2O");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_CO2");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_CH4");
model.result("pg3").run();

model.physics("c2").feature("dir1").setIndex("constraintMethod", "nodal", 0);
model.physics("c2").feature("dir1").setIndex("weakConstraints", "1", 0);
model.physics("c2").feature("dir1").setIndex("weakConstraints", "0", 0);

model.study("std1").feature("stat").setIndex("plistarr", "range(800,100,900)", 0);

model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "80");
model.mesh("mesh2").run();
model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "10");
model.mesh("mesh2").run();
model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "8");
model.mesh("mesh2").run();

model.variable("var6").set("x_CH4_b", "comp1.genext2(comp1.xb_CH4*eps_V)");
model.variable("var6").set("x_CO_b", "comp1.genext2(comp1.xb_CO)*eps_V");
model.variable("var6").set("x_CH4_b", "comp1.genext2(comp1.xb_CH4)*eps_V");
model.variable("var6").set("x_CO2_b", "comp1.genext2(comp1.xb_CO2)*eps_V");
model.variable("var6").set("x_H2_b", "comp1.genext2(comp1.xb_H2O)*eps_V");
model.variable("var6").rename("x_H2_b", "x_H2O_b");
model.variable("var6").set("x_H2O_b", "comp1.genext2(comp1.xb_H2O)*eps_V");

model.physics("c2").feature("dir1").setIndex("r", "x_H2O_b", 3);
model.physics("c2").feature("dir1").setIndex("constraintMethod", "elemental", 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
```

```
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.variable("var6").set("x_CH4_b", "comp1.genext2(comp1.xb_CH4)");
model.variable("var6").set("x_CO_b", "comp1.genext2(comp1.xb_CO)");
model.variable("var6").set("x_CO2_b", "comp1.genext2(comp1.xb_CO2)");
model.variable("var6").set("x_H2O_b", "comp1.genext2(comp1.xb_H2O)");

model.study("std1").feature("stat").setIndex("plistarr", "range(700,100,800)", 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,1000)", 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
```

```
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result("pg2").run();

model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "10");
model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "120");
model.mesh("mesh2").run();
model.mesh("mesh2").feature("map1").feature("dis2").set("numelem", "125");
model.mesh("mesh2").run();
model.mesh("mesh1").feature("size").set("hmax", "0.08");
model.mesh("mesh1").run();

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");

model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").set("solnum", new String[]{"1"});
model.result("pg2").run();

model.physics("tcs").prop("SpeciesProperties").setIndex("FromMassConstraint", "5", 0);
model.physics("tcs").prop("SpeciesProperties").setIndex("FromMassConstraint", "4", 0);
model.physics("tcs").prop("SpeciesProperties").setIndex("FromMassConstraint", "5", 0);
model.physics("tcs").prop("Migration").setIndex("Migration", "1", 0);
model.physics("tcs").prop("Migration").setIndex("Migration", "0", 0);
model.physics("tcs").feature("reac1").setIndex("R", "rb_H2", 3);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
```

```
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");

  model.study("std1").feature("stat").setIndex("plistarr", "range(800,100,1100)", 0);

  model.mesh("mesh2").feature("map1").feature("dis1").set("elemcount", "80");
  model.mesh("mesh2").feature("map1").feature("dis1").set("elemratio", "15");
  model.mesh("mesh2").run();

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);

  return model;
}

public static Model run6(Model model) {
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");

  model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))*0+1e-6[m^2/s]");
  model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))*0+1e-6[m^2/s]");
  model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))*0+1e-6[m^2/s]");
  model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))*0+1e-6[m^2/s]");
  model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))*0+1e-6[m^2/s]");
```

```
model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "4", 0);
model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "1", 0);
model.result("pg1").run();

model.param().set("c_TOT_0teste", "p_TOT/(R_const*T)");
model.param().remove("c_TOT_0teste");

model.result().dataset().create("cln1", "CutLine2D");
model.result().dataset("cln1").setIndex("genpoints", "0.1", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.1", 1, 0);
model.result().dataset("cln1").setIndex("genpoints", "5", 1, 1);
model.result().dataset("cln1").run();
model.result().dataset("cln1").set("bndsnap", "off");
model.result().dataset("cln1").setIndex("genpoints", "1", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "1", 1, 0);
model.result().dataset("cln1").run();
model.result().dataset("cln1").set("genparaactive", "off");
model.result().dataset("cln1").setIndex("genpoints", "(5/100)*(h_tube/1000)", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "(5/100)*(h_tube/1000)", 1, 0);
model.result().dataset("cln1").setIndex("genpoints", "r_pellet", 0, 1);
model.result().dataset("cln1").setIndex("genpoints", "r_pellet", 1, 1);
model.result().dataset("cln1").run();
model.result().dataset("cln1").setIndex("genpoints", "(50/100)*(h_tube/1000)", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "(50/100)*(h_tube/1000)", 1, 0);
model.result().dataset("cln1").run();
model.result().dataset("cln1").run();
model.result().create("pg6", "PlotGroup1D");
model.result("pg6").run();
model.result("pg6").set("data", "cln1");
model.result("pg6").setIndex("looplevelinput", "first", 0);
model.result("pg6").setIndex("looplevelinput", "manual", 0);
model.result("pg6").setIndex("looplevel", "2", 0);
model.result("pg6").setIndex("looplevel", "1", 0);
model.result("pg6").create("lngr1", "LineGraph");
model.result("pg6").feature("lngr1").set("cutline", "cln1");
model.result("pg6").feature("lngr1").set("linefirst", new double[]{0.004999999888241291, 0, 0});
model.result("pg6").feature("lngr1")
     .set("linesecond", new double[]{0.004999999888241291, 0.004999999888241291, 0});
model.result("pg6").feature("lngr1").set("lineisinit", true);
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.005}});
model.result("pg6").feature("lngr1").set("cutmode", "off");
```

```
model.result("pg6").run();
model.result().dataset("cln1").setIndex("genpoints", "0.005*0.1", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.005*0.1", 1, 0);
model.result().dataset("cln1").run();
model.result().dataset("cln1").setIndex("genpoints", "0.005*0.01", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.005*0.01", 1, 0);
model.result().dataset("cln1").run();
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("cutline", "cln1");
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.005}});
model.result("pg6").feature("lngr1").set("cutmode", "off");
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("data", "cln1");
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("cutline", "cln1");
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.005}});
model.result("pg6").feature("lngr1").set("cutmode", "off");
model.result().dataset("cln1").run();
model.result().dataset("cln1").setIndex("genpoints", "0.00005", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.00005", 1, 0);
model.result().dataset("cln1").run();
model.result().dataset("cln1").run();
model.result().dataset("cln1").run();
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("cutline", "cln1");
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.005}});
model.result("pg6").feature("lngr1").set("cutmode", "off");
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("cutline", "cln1");
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.005}});
model.result("pg6").feature("lngr1").set("cutmode", "on");
model.result("pg6").feature("lngr1").set("inputmode", "linesecond");
model.result("pg6").feature("lngr1")
     .set("linesecond", new double[]{1.1461228132247925E-4, 0.004977494012564421, -1});
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.14612E-4, 0.00497749}});
model.result("pg6").feature("lngr1")
     .set("linesecond", new double[]{9.885296458378434E-5, 0.0049932533875107765, -1});
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {9.8853E-5, 0.00499325}});
model.result("pg6").feature("lngr1")
     .set("linesecond", new double[]{1.7764954827725887E-4, 1.3310112990438938E-5, -1});
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.7765E-4, 1.33101E-5}});
model.result("pg6").feature("lngr1")
     .set("linesecond", new double[]{1.1461228132247925E-4, 0.004977494012564421, -1});
model.result().dataset("cln1").set("method", "twopoint");
model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.14612E-4, 0.00497749}});
model.result("pg6").feature("lngr1").set("cutmode", "off");
model.result().dataset("cln1").setIndex("genpoints", "0.0005", 1, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.0005", 0, 0);
model.result().dataset("cln1").run();
model.result().dataset("cln1").setIndex("genpoints", "0.00005", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.00005", 1, 0);
model.result().dataset("cln1").run();
model.result("pg6").run();
model.result("pg6").feature("lngr1").set("expr", "x_H2O");
model.result("pg6").run();
model.result("pg6").feature().duplicate("lngr2", "lngr1");
model.result("pg6").run();
model.result("pg6").feature("lngr2").set("expr", "x_H2");
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "2", 0);
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "4", 0);
model.result("pg6").setIndex("looplevel", "3", 0);
model.result("pg6").setIndex("looplevelinput", "all", 0);
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevelinput", "last", 0);
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").feature("lngr1").setIndex("looplevelinput", "first", 0);
model.result("pg6").feature("lngr1").setIndex("looplevelinput", "last", 0);
```

```
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").feature("lngr2").setIndex("looplevelinput", "last", 0);
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").feature("lngr1").set("data", "parent");
    model.result("pg6").run();
    model.result("pg6").feature("lngr2").set("data", "parent");
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").feature().duplicate("lngr3", "lngr1");
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").feature("lngr1").set("expr", "x_CH4");
    model.result("pg6").run();
    model.result("pg6").feature("lngr2").set("expr", "x_CO");
    model.result("pg6").run();
    model.result("pg6").feature("lngr3").set("expr", "x_CO2");
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").feature("lngr1").set("legend", "on");
    model.result("pg6").feature("lngr1").set("legendmethod", "manual");
    model.result("pg6").feature("lngr1").setIndex("legends", "x_CH4", 0);
    model.result("pg6").run();
    model.result("pg6").feature("lngr2").set("legend", "on");
    model.result("pg6").feature("lngr2").set("legendmethod", "manual");
    model.result("pg6").feature("lngr2").setIndex("legends", "x_CO", 0);
    model.result("pg6").run();
    model.result("pg6").feature("lngr3").set("legend", "on");
    model.result("pg6").feature("lngr3").set("legendmethod", "manual");
    model.result("pg6").feature("lngr3").setIndex("legends", "x_CO2", 0);
    model.result("pg6").feature().duplicate("lngr4", "lngr3");
    model.result("pg6").run();
    model.result("pg6").feature("lngr4").set("expr", "x_H2");
    model.result("pg6").feature("lngr4").setIndex("legends", "x_H2", 0);
    model.result("pg6").run();
    model.result("pg6").feature().duplicate("lngr5", "lngr4");
    model.result("pg6").run();
    model.result("pg6").feature("lngr5").set("expr", "x_H2O");
    model.result("pg6").feature("lngr5").setIndex("legends", "x_H2O", 0);
    model.result("pg6").run();
    model.result("pg3").run();
    model.result("pg6").run();

    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_11*eps_V*c_TOT*(y^2)", 0, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_21*eps_V*c_TOT*(y^2)", 1, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_31*eps_V*c_TOT*(y^2)", 2, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_41*eps_V*c_TOT*(y^2)", 3, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_12*eps_V*c_TOT*(y^2)", 4, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_22*eps_V*c_TOT*(y^2)", 5, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_32*eps_V*c_TOT*(y^2)", 6, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_42*eps_V*c_TOT*(y^2)", 7, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_13*eps_V*c_TOT*(y^2)", 8, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_23*eps_V*c_TOT*(y^2)", 9, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_33*eps_V*c_TOT*(y^2)", 10, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_43*eps_V*c_TOT*(y^2)", 11, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_14*eps_V*c_TOT*(y^2)", 12, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_24*eps_V*c_TOT*(y^2)", 13, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_34*eps_V*c_TOT*(y^2)", 14, 3);
    model.physics("c2").feature("cfeq1").setIndex("c", "-invA_44*eps_V*c_TOT*(y^2)", 15, 3);

    model.variable("var1").set("c_CH4", "eps_V*c_TOT*x_CH4");
    model.variable("var1").set("c_CO", "eps_V*c_TOT*x_CO");
    model.variable("var1").set("c_CO2", "eps_V*c_TOT*x_CO2");
    model.variable("var1").set("c_H2", "eps_V*c_TOT*x_H2");
    model.variable("var1").set("c_H2O", "eps_V*c_TOT*x_H2O");
    model.variable("var1").set("c_TOT", "eps_V*p_in/(R_const*T)");
    model.variable("var1")
         .set("rho_gas", "eps_V*p_TOT*(x_CH4*M_CH4+x_CO*M_CO+x_CO2*M_CO2+x_H2*M_H2+x_H2O*M_H2O)/(R_const*T)");
    model.variable("var1").set("N_CH4",
"eps_V*c_TOT*(invA_11*x_CH4y+invA_12*x_COy+invA_13*x_CO2y+invA_14*x_H2Oy)");
    model.variable("var1").set("N_CO",
"eps_V*c_TOT*(invA_21*x_CH4y+invA_22*x_COy+invA_23*x_CO2y+invA_24*x_H2Oy)");
    model.variable("var1").set("N_CO2",
"eps_V*c_TOT*(invA_31*x_CH4y+invA_32*x_COy+invA_33*x_CO2y+invA_34*x_H2Oy)");
    model.variable("var1").set("N_H2O",
"eps_V*c_TOT*(invA_41*x_CH4y+invA_42*x_COy+invA_43*x_CO2y+invA_44*x_H2Oy)");
    model.variable("var1").set("c_TOT", "eps_V*p_TOT/(R_const*T)");
```

```
    model.variable("var2")
         .set("r1", "k1*sqrt(eps_V*p_TOT/1[bar])*sqrt(x_H2)*((eps_V*p_TOT*(x_CH4*x_H2O)/x_H2)-
(((eps_V*p_TOT)^3)*(x_H2^2)*x_CO/K1))/(DEN_star)^2");
    model.variable("var2")
         .set("r2", "k2*eps_V*p_TOT*x_H2*((((eps_V*p_TOT)^2)*x_CO*x_H2O)-
(((eps_V*p_TOT^2))*x_H2*x_CO2/K2))/(DEN_star)^2");
    model.variable("var2")
         .set("r3", "k3*sqrt((eps_V*p_TOT)/1[bar])*sqrt(x_H2)*(((eps_V*p_TOT)*x_CH4*x_H2O/(x_H2^2))-
(((eps_V*p_TOT)^3)*(x_H2^2)*x_CO2/K3))/(DEN_star)^2");
    model.variable("var2")
         .set("DEN_star",
"(eps_V*p_TOT*x_H2)+(((eps_V*p_TOT)^2)*K_CO*x_H2*x_CO)+(((eps_V*p_TOT)^2)*K_H2*(x_H2^2))+(((eps_V*p_TOT)^2)*K_C
H4*x_H2*x_CH4)+(eps_V*p_TOT*K_H2O*x_H2O)");
    model.variable("var7").set("rb_CH4", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_CH4)*eps_A");
    model.variable("var7").set("rb_CO", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_CO)*eps_A");
    model.variable("var7").set("rb_CO2", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_CO2)*eps_A");
    model.variable("var7").set("rb_H2", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_H2)*eps_A");
    model.variable("var7").set("rb_H2O", "(1-void_fraction)*(3/r_pellet)*comp2.genext1(comp2.J_H2O)*eps_A");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg1").setIndex("looplevel", "3", 0);
    model.result("pg1").setIndex("looplevel", "4", 0);
    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg2").setIndex("looplevel", "4", 0);
    model.result("pg2").run();
    model.result("pg4").run();
    model.result("pg4").setIndex("looplevelinput", "last", 0);
    model.result("pg4").run();
    model.result("pg5").run();
    model.result("pg3").run();
    model.result("pg6").run();

    model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))*0+1.665e-
6[m^2/s]");
    model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))*0+1.26e-6[m^2/s]");
    model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))*0+1.005e-
6[m^2/s]");
    model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))*0+4.695e-6[m^2/s]");
    model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))*0+1.57e-
6[m^2/s]");
    model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))*0+1.665e-
4[m^2/s]");
    model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))*0+1.26e-4[m^2/s]");
```

```
    model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))*0+1.005e-
4[m^2/s]");
    model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))*0+4.695e-4[m^2/s]");
    model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))*0+1.57e-
4[m^2/s]");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,800)", 0);

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").runAll();
```

```
    model.result("pg1").run();
    model.result("pg6").run();

    model.variable("var7").rename("u", "u_OLD");
    model.variable("var7").set("u_OLD", "u_0*(cb_TOT/c_TOT_0)*(p_in/p_out)", "Check LATER");
    model.variable("var7").set("u", "u_ave");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg6").run();

    model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))*0+1.665e-
2[m^2/s]");
    model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))*0+1.26e-2[m^2/s]");
    model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))*0+1.005e-
2[m^2/s]");
    model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))*0+4.695e-2[m^2/s]");
    model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))*0+1.57e-
2[m^2/s]");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
```

```
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");
  model.sol("sol1").runAll();

  model.result("pg1").run();
  model.result("pg6").run();

  model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))*0");
  model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))*0");
  model.variable("var5").set("DK_CO2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO2))");
  model.variable("var5").set("DK_CH4", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CH4))");
  model.variable("var5").set("DK_CO", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_CO))");
  model.variable("var5").set("DK_H2", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2))");
  model.variable("var5").set("DK_H2O", "(4/3)*(4.86e-5[cm])*sqrt((R_const*T)/(2*pi*M_H2O))");

  model.result("pg3").run();
  model.result("pg3").run();
  model.result("pg3").feature("surf1").set("expr", "x_H2");
  model.result("pg3").run();

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");
  model.sol("sol1").runAll();

  model.result("pg1").run();
  model.result("pg1").setIndex("looplevel", "4", 0);

  return model;
}

public static Model run7(Model model) {
  model.result("pg1").run();
  model.result("pg6").run();
  model.result("pg3").run();
  model.result("pg3").feature("surf1").set("expr", "x_H2O");
  model.result("pg3").run();
  model.result("pg3").feature("surf1").create("hght1", "Height");
  model.result("pg3").run();
  model.result("pg3").run();
  model.result("pg3").feature("surf1").set("expr", "x_H2");
```

```
model.result("pg3").run();
model.result("pg3").run();
model.result("pg2").run();
model.result("pg2").setIndex("looplevel", "4", 0);
model.result("pg2").run();
model.result("pg4").run();
model.result("pg4").run();

model.study("std1").feature("stat").setIndex("plistarr", "range(500,100,1100)", 0);

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg1").setIndex("looplevel", "7", 0);
model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").setIndex("looplevel", "7", 0);
model.result("pg2").run();
model.result("pg4").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "7", 0);
model.result("pg3").run();
model.result("pg3").set("allowtableupdate", false);
model.result("pg3").set("title", "T_in(7)=1100 Surface: Molar fraction, H2O (1)");
model.result("pg3").set("xlabel", "");
model.result("pg3").set("ylabel", "");
model.result("pg3").feature("surf1").set("rangeunit", "1");
model.result("pg3").feature("surf1").set("rangecolormin", 2.4954451886882723E-5);
model.result("pg3").feature("surf1").set("rangecolormax", 0.4651411700590467);
model.result("pg3").feature("surf1").set("rangecoloractive", "off");
model.result("pg3").feature("surf1").set("rangedatamin", 2.4954451886882723E-5);
model.result("pg3").feature("surf1").set("rangedatamax", 0.4651411700590467);
model.result("pg3").feature("surf1").set("rangedataactive", "off");
model.result("pg3").feature("surf1")
     .set("rangeactualminmax", new double[]{2.4954451886882723E-5, 0.4651411700590467});
model.result("pg3").feature("surf1").feature("hght1").set("scale", 0.016124137106693703);
model.result("pg3").feature("surf1").feature("hght1").set("scaleactive", false);
model.result("pg3").set("renderdatacached", false);
model.result("pg3").set("allowtableupdate", true);
model.result("pg3").set("renderdatacached", true);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
```

```
model.result("pg3").set("allowtableupdate", false);
model.result("pg3").set("title", "T_in(7)=1100 Surface: Molar fraction, H2O (1)");
model.result("pg3").set("xlabel", "");
model.result("pg3").set("ylabel", "");
model.result("pg3").feature("surf1").set("rangeunit", "1");
model.result("pg3").feature("surf1").set("rangecolormin", 2.4954451886882723E-5);
model.result("pg3").feature("surf1").set("rangecolormax", 0.4651411700590467);
model.result("pg3").feature("surf1").set("rangecoloractive", "off");
model.result("pg3").feature("surf1").set("rangedatamin", 2.4954451886882723E-5);
model.result("pg3").feature("surf1").set("rangedatamax", 0.4651411700590467);
model.result("pg3").feature("surf1").set("rangedataactive", "off");
model.result("pg3").feature("surf1")
     .set("rangeactualminmax", new double[]{2.4954451886882723E-5, 0.4651411700590467});
model.result("pg3").feature("surf1").feature("hght1").set("scale", 0.016124137106693703);
model.result("pg3").feature("surf1").feature("hght1").set("scaleactive", false);
model.result("pg3").set("renderdatacached", false);
model.result("pg3").set("allowtableupdate", true);
model.result("pg3").set("renderdatacached", true);
model.result("pg6").run();

model.param().set("u_ave", "0.5[m/s]");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();

model.param().set("u_ave", "0.4[m/s]");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
```

```
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg2").run();
    model.result("pg4").run();
    model.result("pg2").run();
    model.result("pg4").run();
    model.result("pg5").run();
    model.result("pg5").setIndex("looplevel", "4", 0);
    model.result("pg5").setIndex("looplevel", "7", 0);
    model.result("pg5").run();
    model.result("pg5").run();

    model.variable("var7").set("rp_CH4R", "comp2.genext1(-comp2.r1_A-comp2.r3_A)");
    model.variable("var7").set("rp_COR", "comp2.genext1(comp2.r1_A-comp2.r2_A)");
    model.variable("var7").set("rp_CO2R", "comp2.genext1(comp2.r2_A+comp2.r3_A)");
    model.variable("var7").set("rp_H2OR", "comp2.genext1(-comp2.r1_A-comp2.r2_A-2*comp2.r3_A)");
    model.variable("var7").set("rp_H2R", "-(rp_CH4R+rp_COR+rp_CO2R+rp_H2OR)");

    model.result("pg5").run();
    model.result().duplicate("pg7", "pg5");
    model.result("pg7").run();
    model.result("pg7").run();

    model.variable("var7")
         .set("effect_fac",
"0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000");
    model.variable("var7").rename("rp_CH4R", "W_CH4R");
    model.variable("var7").set("W_CH4R", "comp2.genext1(-comp2.r1_A-comp2.r3_A)");
    model.variable("var7").rename("W_CH4R", "rp_CH4R");
    model.variable("var7").set("rp_CH4R", "comp2.genext1(-comp2.r1_A-comp2.r3_A)");
    model.variable("var7").set("W_CH4_R", "comp2.genext1(comp2.N_CH4)*eps_A*Ax_pellet");
    model.variable("var7").set("W_CO_R", "comp2.genext1(comp2.N_CO)*eps_A*Ax_pellet");
    model.variable("var7").set("W_CO2_R", "comp2.genext1(comp2.N_CO2)*eps_A*Ax_pellet");
    model.variable("var7").set("W_H2_R", "comp2.genext1(comp2.N_H2)*eps_A*Ax_pellet");
    model.variable("var7").set("W_H2O_R", "comp2.genext1(comp2.N_H2O)*eps_A*Ax_pellet");
    model.variable("var7").set("W_CH4_R0", "rp_CH4R*1");

    model.param().descr("Vm_pellet", "Pellet's volume");

    model.variable("var7").set("W_CH4_R0", "rp_CH4R*eps_I*Vm_pellet");
    model.variable("var7").set("W_CO_R0", "rp_COR*eps_I*Vm_pellet");
    model.variable("var7").set("W_CO2_R0", "rp_CO2R*eps_I*Vm_pellet");
    model.variable("var7").set("W_H2_R0", "rp_H2R*eps_I*Vm_pellet");
    model.variable("var7").set("W_H2O_R0", "rp_H2OR*eps_I*Vm_pellet");
    model.variable("var7").set("eta_CH4", "W_CH4_R/W_CH4_R0");
    model.variable("var7").set("eta_CO", "W_CO_R/W_CO_R0");
    model.variable("var7").set("eta_CO2", "W_CO2_R/W_CO2_R0");
    model.variable("var7").set("eta_H2", "W_H2_R/W_H2_R0");
    model.variable("var7").set("eta_H2O", "W_H2O_R/W_H2O_R0");

    model.param().set("u_ave", "0.3[m/s]");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
```

```
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").feature("lngr1").set("expr", "eta_CH4");
model.result("pg7").feature("lngr1").setIndex("legends", "eta_CH4", 0);
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").feature("lngr2").set("expr", "eta_CO");
model.result("pg7").feature("lngr2").setIndex("legends", "eta_CO", 0);
model.result("pg7").run();
model.result("pg7").feature("lngr3").set("expr", "eta_CO2");
model.result("pg7").feature("lngr3").setIndex("legends", "eta_CO2", 0);
model.result("pg7").run();
model.result("pg7").feature("lngr4").set("expr", "eta_H2");
model.result("pg7").feature("lngr4").setIndex("legends", "eta_H2", 0);
model.result("pg7").run();
model.result("pg7").feature("lngr5").set("expr", "eta_H2O");
model.result("pg7").feature("lngr5").setIndex("legends", "eta_H2O", 0);
model.result("pg7").run();
model.result().duplicate("pg8", "pg7");
model.result("pg8").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_CO2");
model.result("pg3").run();
model.result("pg6").run();
model.result().dataset("cln1").setIndex("genpoints", "0.0005", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.0005", 1, 0);
model.result().dataset("cln1").run();
model.result().dataset("cln1").setIndex("genpoints", "0.005", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.005", 1, 0);
model.result().dataset("cln1").run();
model.result("pg6").run();
model.result().dataset("cln1").setIndex("genpoints", "0.0025", 0, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.0025", 1, 0);
model.result().dataset("cln1").run();
model.result("pg6").run();
model.result().dataset("cln1").setIndex("genpoints", "0.0015", 1, 0);
model.result().dataset("cln1").setIndex("genpoints", "0.0015", 0, 0);
model.result().dataset("cln1").run();
model.result("pg6").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg7").run();
model.result("pg7").run();
model.result("pg8").run();
model.result("pg8").feature("lngr1").set("expr", "rp_CH4R");
model.result("pg8").feature("lngr1").setIndex("legends", "rp_CH4R", 0);
model.result("pg8").run();
model.result("pg8").feature("lngr2").set("expr", "rp_COR");
model.result("pg8").feature("lngr2").setIndex("legends", "rp_COR", 0);
model.result("pg8").run();
model.result("pg8").feature("lngr3").set("expr", "rp_CO2R");
model.result("pg8").feature("lngr3").setIndex("legends", "rp_CO2R", 0);
```

```
model.result("pg8").run();
model.result("pg8").feature("lngr4").set("expr", "rp_H2R");
model.result("pg8").feature("lngr4").setIndex("legends", "rp_H2R", 0);
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").feature("lngr5").set("expr", "rp_H2OR");
model.result("pg8").feature("lngr5").setIndex("legends", "rp_H2OR", 0);
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").setIndex("looplevel", "1", 0);
model.result("pg8").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "1", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "3", 0);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "4", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "5", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "6", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "5", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "6", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "7", 0);
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "4", 0);
model.result("pg3").setIndex("looplevel", "5", 0);
model.result("pg3").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevelinput", "manual", 0);
model.result("pg6").setIndex("looplevel", "4", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "6", 0);
model.result("pg6").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "7", 0);
model.result("pg6").run();
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "5", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "7", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").setIndex("looplevel", "1", 0);
model.result("pg7").run();
model.result("pg7").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg8").run();
```

```
model.result("pg5").run();
model.result("pg8").run();
model.result("pg8").feature("lngr1").set("expr", "W_CH4_R");
model.result("pg8").feature("lngr1").setIndex("legends", "W_CH4_R", 0);
model.result("pg8").run();
model.result("pg8").feature("lngr2").set("expr", "W_CO_R");
model.result("pg8").feature("lngr2").setIndex("legends", "W_CO_R", 0);
model.result("pg8").run();
model.result("pg8").feature("lngr3").set("expr", "W_CO2_R");
model.result("pg8").feature("lngr3").setIndex("legends", "W_CO2_R", 0);
model.result("pg8").run();
model.result("pg8").feature("lngr4").set("expr", "W_H2_R");
model.result("pg8").feature("lngr4").setIndex("legends", "W_H2_R", 0);
model.result("pg8").run();
model.result("pg8").feature("lngr5").set("expr", "W_H2O_R");
model.result("pg8").feature("lngr5").setIndex("legends", "W_H2O_R", 0);
model.result("pg8").run();
model.result("pg8").setIndex("looplevel", "4", 0);
model.result("pg8").run();
model.result("pg8").setIndex("looplevel", "3", 0);
model.result("pg8").run();
model.result("pg8").setIndex("looplevel", "7", 0);
model.result("pg8").run();
model.result("pg8").setIndex("looplevel", "2", 0);
model.result("pg8").run();
model.result().duplicate("pg9", "pg8");
model.result("pg9").run();
model.result("pg9").run();
model.result("pg9").feature("lngr1").set("expr", "W_CH4_R0");
model.result("pg9").feature("lngr1").setIndex("legends", "W_CH4_R0", 0);
model.result("pg9").run();
model.result("pg9").feature("lngr2").set("expr", "W_CO_R0");
model.result("pg9").feature("lngr2").setIndex("legends", "W_CO_R0", 0);
model.result("pg9").run();
model.result("pg9").feature("lngr3").set("expr", "W_CO2_R0");
model.result("pg9").feature("lngr3").setIndex("legends", "W_CO2_R0", 0);
model.result("pg9").run();
model.result("pg9").feature("lngr4").set("expr", "W_H2_R0");
model.result("pg9").feature("lngr4").setIndex("legends", "W_H2_R0", 0);
model.result("pg9").run();
model.result("pg9").feature("lngr5").set("expr", "W_H2O_R0");
model.result("pg9").feature("lngr5").setIndex("legends", "W_H2O_R0", 0);
model.result("pg9").run();
model.result("pg9").setIndex("looplevel", "1", 0);
model.result("pg9").run();
model.result("pg9").setIndex("looplevel", "4", 0);
model.result("pg9").run();
model.result("pg9").setIndex("looplevel", "6", 0);
model.result("pg9").run();
model.result("pg4").run();
model.result("pg2").run();
model.result("pg9").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature().duplicate("surf2", "surf1");
model.result("pg3").run();
model.result("pg3").feature("surf2").set("expr", "x_CO");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf2").feature("hght1").set("isheightaxisshown", "on");
model.result("pg3").feature("surf2").feature("hght1").set("offset", ".005");
model.result("pg3").run();
model.result("pg3").feature("surf2").feature("hght1").set("offset", ".01");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature().duplicate("surf3", "surf2");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_CH4");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf2").set("expr", "x_CO2");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf3").feature("hght1").set("offset", ".02");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature().duplicate("surf4", "surf3");
```

```
model.result("pg3").run();
model.result("pg3").feature("surf4").set("colorlegend", "on");
model.result("pg3").feature("surf4").set("expr", "x_H2");
model.result("pg3").run();
model.result("pg3").feature("surf4").feature("hght1").set("offset", ".03");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature().duplicate("surf5", "surf4");
model.result("pg3").run();
model.result("pg3").feature("surf5").set("expr", "x_H2O");
model.result("pg3").run();
model.result("pg3").feature("surf5").feature("hght1").set("offset", ".04");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "6", 0);
model.result("pg3").run();
model.result("pg9").run();
model.result("pg9").setIndex("looplevel", "4", 0);
model.result("pg9").run();

model.variable("var7")
     .set("q_r1", "(1-void_fraction)*comp2.genext1(comp2.r1_A*comp2.dH1*eps_I)*Ac_tube*x/(2*pi*r_tube*x)");
model.variable("var7")
     .set("q_r2", "(1-void_fraction)*comp2.genext1(comp2.r2_A*comp2.dH1*eps_I)*Ac_tube*x/(2*pi*r_tube*x)");
model.variable("var7")
     .set("q_r3", "(1-void_fraction)*comp2.genext1(comp2.r3_A*comp2.dH1*eps_I)*Ac_tube*x/(2*pi*r_tube*x)");
model.variable("var7").set("q_tot", "q_r1+q_r2+q_r3");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg9").run();
model.result("pg8").run();
model.result("pg9").run();
model.result("pg1").run();
model.result("pg4").run();
model.result("pg2").run();
model.result("pg5").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg8").run();
model.result("pg9").run();
model.result().duplicate("pg10", "pg9");
model.result("pg10").run();
model.result("pg10").run();
model.result("pg10").run();
```

```
  model.result("pg10").feature("lngr2").active(false);
  model.result("pg10").feature("lngr3").active(false);
  model.result("pg10").feature("lngr4").active(false);
  model.result("pg10").feature("lngr5").active(false);
  model.result("pg10").run();
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "q_tot");
  model.result("pg10").run();
  model.result("pg10").run();
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "dH1");
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").selection().set(new int[]{1});
  model.result("pg10").feature("lngr1").set("data", "cln1");
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").setIndex("looplevelinput", "last", 0);
  model.result("pg10").run();
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "dH2");
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "dH1");
  model.result("pg10").run();
  model.result().dataset("cln1").setIndex("genpoints", "0.005", 1, 1);
  model.result().dataset("cln1").run();
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("xdata", "arc");
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "dH1*r1");
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "dH1*r1+dH2*r2+dH3*r3");
  model.result("pg10").run();
  model.result("pg10").feature("lngr1").set("expr", "(dH1*r1_A+dH2*r2_A+dH3*r3_A)*eps_I");
  model.result("pg10").run();

  model.cpl().create("genproj1", "GeneralProjection", "geom2");
  model.cpl("genproj1").selection().set(new int[]{1});
  model.cpl("genproj1").selection().all();
  model.cpl("genproj1").set("srcmap", new String[]{"", "dH1*r1"});

  model.result("pg10").run();

  model.cpl("genproj1").set("srcmap", new String[]{"x", "y"});

  model.variable("var7")
       .set("q_pellet", "genproj1((comp2.dH1*comp2.r1_A+comp2.dH2*comp2.r2_A+comp2.dH3*comp2.r3_A)*eps_I)");
  model.variable("var7").remove("q_r1");
  model.variable("var7").remove("q_r2");
  model.variable("var7").remove("q_r3");
  model.variable("var7").set("q_tot", "(1-void_fraction)*q_pellet");

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");

  return model;
}

public static Model run8(Model model) {
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
```

```
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").runAll();

    model.result("pg1").run();
    model.result("pg10").run();
    model.result("pg10").feature("lngr1").set("expr", "");
    model.result("pg10").feature("lngr1").set("data", "dset1");
    model.result("pg10").run();
    model.result("pg10").run();
    model.result("pg9").run();
    model.result().duplicate("pg11", "pg9");
    model.result("pg11").run();
    model.result("pg11").run();
    model.result("pg11").run();
    model.result("pg11").feature("lngr2").active(false);
    model.result("pg11").feature("lngr3").active(false);
    model.result("pg11").feature("lngr4").active(false);
    model.result("pg11").feature("lngr5").active(false);
    model.result("pg11").run();
    model.result("pg11").run();
    model.result("pg11").feature("lngr1").set("expr", "q_tot");
    model.result("pg11").feature("lngr1").setIndex("legends", "q_tot", 0);

    model.variable("var7")
         .set("q_pellet",
"comp2.genproj1((comp2.dH1*comp2.r1_A+comp2.dH2*comp2.r2_A+comp2.dH3*comp2.r3_A)*eps_I)");

    model.result("pg11").run();

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");
    model.sol("sol1").runAll();

    model.result("pg1").run();

    model.variable("var7")
         .set("q_pellet",
"comp2.genext1((comp2.dH1*comp2.r1_A+comp2.dH2*comp2.r2_A+comp2.dH3*comp2.r3_A)*eps_I*r_pellet^2)");

    model.sol("sol1").study("std1");
```

```
model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();
model.result("pg11").run();

model.label("Pellet 1D and bed reactor - 2nd try-rev.0.mph");

model.result("pg11").run();

model.mesh("mesh1").feature("size").set("hmax", "0.02");
model.mesh("mesh1").run();
model.mesh("mesh2").feature("map1").feature("dis2").set("numelem", "500");
model.mesh("mesh2").run();

model.label("Pellet 1D and bed reactor - 2nd try-rev.0.mph");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
```

```
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg3").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg11").run();
model.result("pg10").run();
model.result("pg9").run();
model.result("pg8").run();
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").setIndex("looplevel", "5", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "7", 0);
model.result("pg7").run();

model.label("Pellet 1D and bed reactor - 2nd try-rev.A.mph");

model.result("pg7").run();
model.result("pg11").run();
model.result("pg8").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg1").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf2").active(false);
model.result("pg3").feature("surf3").active(false);
model.result("pg3").feature("surf4").active(false);
model.result("pg3").feature("surf5").active(false);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").setIndex("looplevel", "7", 0);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf2").active(true);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").active(false);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf2").active(false);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf1").active(true);
model.result("pg3").run();
```

```
model.result("pg3").feature("surf1").set("expr", "x_CO");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_CO2");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_H2");
model.result("pg3").run();
model.result("pg3").feature("surf1").set("expr", "x_H2O");
model.result("pg3").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg8").run();
model.result("pg9").run();
model.result("pg8").run();
model.result("pg7").run();
model.result("pg5").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg7").set("xlabelactive", "off");
model.result("pg7").set("ylabelactive", "off");
model.result("pg7").setIndex("looplevel", "1", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "5", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "7", 0);
model.result("pg7").run();

model.label("Pellet 1D and bed reactor - 2nd try-rev.A.mph");

model.result("pg7").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg3").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "5", 0);
model.result("pg6").setIndex("looplevel", "6", 0);
model.result("pg6").setIndex("looplevel", "7", 0);
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "6", 0);
model.result("pg7").run();
model.result("pg8").run();
model.result("pg9").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg7").run();
model.result("pg9").run();
model.result("pg10").run();
model.result("pg8").run();
model.result("pg6").run();

model.mesh("mesh2").feature("map1").feature("dis2").set("numelem", "100");
model.mesh("mesh2").run();
model.mesh("mesh1").feature("size").set("hmax", "0.1");
model.mesh("mesh1").run();

model.label("Pellet 1D and bed reactor - 2nd try-rev.A1.mph");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");
```

```
model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg2").run();

model.param().descr("rho_catalyst", "Catalyst density (weight of catalyst per volume of solid+void)");
model.param().set("eps_A", "eps_V/Tau");

model.sol("sol1").study("std1");

model.study("std1").feature("stat").set("notlistsolnum", 1);
model.study("std1").feature("stat").set("notsolnum", "1");
model.study("std1").feature("stat").set("listsolnum", 1);
model.study("std1").feature("stat").set("solnum", "1");

model.sol("sol1").feature().remove("s1");
model.sol("sol1").feature().remove("v1");
model.sol("sol1").feature().remove("st1");
model.sol("sol1").create("st1", "StudyStep");
model.sol("sol1").feature("st1").set("study", "std1");
model.sol("sol1").feature("st1").set("studystep", "stat");
model.sol("sol1").create("v1", "Variables");
model.sol("sol1").feature("v1").set("control", "stat");
model.sol("sol1").create("s1", "Stationary");
model.sol("sol1").feature("s1").create("p1", "Parametric");
model.sol("sol1").feature("s1").feature().remove("pDef");
model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
model.sol("sol1").feature("s1").set("control", "stat");
model.sol("sol1").feature("s1").create("seDef", "Segregated");
model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").create("d1", "Direct");
model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
model.sol("sol1").feature("s1").feature().remove("fcDef");
model.sol("sol1").feature("s1").feature().remove("seDef");
model.sol("sol1").attach("std1");
model.sol("sol1").runAll();

model.result("pg1").run();
model.result("pg2").run();
model.result("pg4").run();
model.result("pg2").run();
model.result("pg5").run();
model.result("pg3").run();
model.result("pg6").run();
model.result("pg6").setIndex("looplevel", "4", 0);
```

```
    model.result("pg6").setIndex("looplevel", "7", 0);
    model.result("pg6").run();
    model.result("pg6").setIndex("looplevel", "5", 0);
    model.result("pg6").setIndex("looplevel", "7", 0);
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").feature("lngr1").set("cutline", "cln1");
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.14612E-4, 0.00497749}});
    model.result("pg6").feature("lngr1").set("cutmode", "off");
    model.result("pg6").run();
    model.result("pg6").feature("lngr1").set("cutline", "cln1");
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.14612E-4, 0.00497749}});
    model.result("pg6").feature("lngr1").set("cutmode", "on");
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{1.1393241584300995E-4, 0.004991251043975353, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.13932E-4, 0.00499125}});
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{0.0015926104970276356, 0.002162475837394595, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.00159261, 0.00216248}});
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{0.0038749179802834988, 0.0049591064453125, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.00387492, 0.00495911}});
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{0.004646402318030596, 0.00463765487074852, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.0046464, 0.00463765}});
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{0.004903563763946295, 0.004926961846649647, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.00490356, 0.004927}});
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{0.004999999422580004, 0.004991251043975353, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.00499125}});
    model.result("pg6").feature("lngr1").set("cutmode", "off");
    model.result("pg6").run();
    model.result("pg6").feature("lngr1").set("cutline", "cln1");
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.00499125}});
    model.result("pg6").feature("lngr1").set("cutmode", "on");
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{1.7496757209300995E-5, 0.004926961846649647, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {1.74968E-5, 0.004927}});
    model.result("pg6").feature("lngr1").set("linesecond", new double[]{2.74658203125E-4, 7.303920574486256E-5,
-1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {2.74658E-4, 7.30392E-5}});
    model.result("pg6").feature("lngr1")
         .set("linesecond", new double[]{8.17873515188694E-5, 0.004991251043975353, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1").set("genpoints", new double[][]{{0.005, 0}, {8.17873E-5, 0.00499125}});
    model.result("pg6").feature("lngr1").set("inputmode", "linefirst");
    model.result("pg6").feature("lngr1")
         .set("linefirst", new double[]{1.1393241584300995E-4, -2.3396452888846397E-5, -1});
    model.result().dataset("cln1").set("method", "twopoint");
    model.result().dataset("cln1")
         .set("genpoints", new double[][]{{1.13932E-4, -2.33963E-5}, {8.17873E-5, 0.00499125}});
    model.result("pg6").feature("lngr1").set("cutmode", "off");
    model.result("pg6").run();
    model.result("pg7").run();
    model.result("pg8").run();
    model.result("pg9").run();
    model.result("pg10").run();
    model.result("pg11").run();
    model.result("pg10").run();
    model.result("pg9").run();
    model.result("pg10").run();
    model.result("pg10").run();
    model.result().dataset().create("cln2", "CutLine2D");
    model.result().dataset("cln2").set("data", "dset2");
    model.result("pg10").feature("lngr1").set("cutline", "cln2");
    model.result("pg10").feature("lngr1").set("linefirst", new double[]{0.004999999888241291, 0, 0});
    model.result("pg10").feature("lngr1")
```

```
        .set("linesecond", new double[]{0.004999999888241291, 0.004999999888241291, 0});
    model.result("pg10").feature("lngr1").set("lineisinit", true);
    model.result().dataset("cln2").set("method", "twopoint");
    model.result().dataset("cln2").set("genpoints", new double[][]{{0.005, 0}, {0.005, 0.005}});
    model.result("pg10").feature("lngr1").set("cutmode", "on");
    model.result("pg10").feature("lngr1").set("listen", false);
    model.result("pg10").feature("lngr1").set("data", "cln2");
    model.result("pg10").feature("lngr1").set("listen", true);
    model.result("pg10").feature("lngr1").set("cutmode", "off");
    model.result("pg10").run();
    model.result("pg10").run();
    model.result("pg10").run();
    model.result("pg9").run();
    model.result("pg10").run();
    model.result("pg10").run();
    model.result("pg10").run();
    model.result("pg2").run();
    model.result("pg4").run();
    model.result("pg5").run();
    model.result("pg3").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();
    model.result("pg6").run();

    model.label("Pellet 1D and bed reactor - 2nd try-rev.A1.mph");

    model.result("pg6").run();

    model.label("Pellet 1D and bed reactor - 2nd try-rev.A1.mph");

    model.result("pg6").run();
    model.result("pg3").run();
    model.result("pg3").run();
    model.result("pg3").run();

    model.physics("c").feature("cfeq1").setIndex("f", "px+(beta_0*(p_in/p)*(T/T_in)*(FT/FT_0))", 0);
    model.physics("c").feature("cfeq1").setIndex("c", "0", 0);
    model.physics("c").feature("cfeq1").setIndex("da", "0", 0);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 1);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 2);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 3);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 4);
    model.physics("tcs").feature("cdm1").setIndex("um", new String[]{"1e-13[s*mol/kg]"}, 0);
    model.physics("tcs").feature("cdm1").setIndex("minput_pressure", "p", 0);

    model.param().set("p_TOT", "p_in");

    model.variable("var7").remove("u_OLD");
    model.variable("var7").set("cb_TOT", "p/(R_const*T)");
    model.variable("var2")
        .set("r1", "k1*sqrt(eps_V*p_pellet/1[bar])*sqrt(x_H2)*((eps_V*p_pellet*(x_CH4*x_H2O)/x_H2)-
(((eps_V*p_pellet)^3)*(x_H2^2)*x_CO/K1))/(DEN_star)^2");
    model.variable("var2")
        .set("r2", "k2*eps_V*p_pellet*x_H2*((((eps_V*p_pellet)^2)*x_CO*x_H2O)-
(((eps_V*p_pellet^2))*x_H2*x_CO2/K2))/(DEN_star)^2");
    model.variable("var2")
        .set("r3", "k3*sqrt((eps_V*p_pellet)/1[bar])*sqrt(x_H2)*(((eps_V*p_pellet)*x_CH4*x_H2O/(x_H2^2))-
(((eps_V*p_pellet)^3)*(x_H2^2)*x_CO2/K3))/(DEN_star)^2");
    model.variable("var2")
        .set("DEN_star",
"(eps_V*p_pellet*x_H2)+(((eps_V*p_pellet)^2)*K_CO*x_H2*x_CO)+(((eps_V*p_pellet)^2)*K_H2*(x_H2^2))+(((eps_V*p_pe
llet)^2)*K_CH4*x_H2*x_CH4)+(eps_V*p_pellet*K_H2O*x_H2O)");
    model.variable("var1").set("p_pellet", "comp1.p");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
```

```
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.study("std1").feature("stat").set("physselection", "tcs");
    model.study("std1").feature("stat").set("activate", new String[]{"tcs", "on", "c", "on", "c2", "on"});

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").feature("st1").set("study", "std1");
    model.sol("sol1").feature("st1").set("studystep", "stat");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").feature("v1").set("control", "stat");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").feature().remove("pDef");
    model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
    model.sol("sol1").feature("s1").set("control", "stat");
    model.sol("sol1").feature("s1").create("seDef", "Segregated");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").create("d1", "Direct");
    model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
    model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
    model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
    model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
    model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.sol("sol1").feature("s1").feature().remove("seDef");
    model.sol("sol1").attach("std1");

    model.variable("var7")
         .set("beta_0", "(G/(rho_gas*(2*r_pellet))*((1-void_fraction)/(void_fraction^3)))*(150*(1-
void_fraction)*mu_gas/(2*r_pellet)+1.75*G)");
    model.variable("var7").set("rho_gas", "rho_gas_0*(p/p_in)*(FT_0/FT)");

    model.sol("sol1").study("std1");

    model.study("std1").feature("stat").set("notlistsolnum", 1);
    model.study("std1").feature("stat").set("notsolnum", "1");
    model.study("std1").feature("stat").set("listsolnum", 1);
    model.study("std1").feature("stat").set("solnum", "1");

    model.sol("sol1").feature().remove("s1");
    model.sol("sol1").feature().remove("v1");
    model.sol("sol1").feature().remove("st1");
    model.sol("sol1").create("st1", "StudyStep");

    return model;
  }
```

```
public static Model run9(Model model) {
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");

  model.variable("var1").remove("p_pellet");
  model.variable("var6").set("p_pellet", "comp1.genext2(comp1.p)");

  model.sol("sol1").study("std1");

  model.study("std1").feature("stat").set("notlistsolnum", 1);
  model.study("std1").feature("stat").set("notsolnum", "1");
  model.study("std1").feature("stat").set("listsolnum", 1);
  model.study("std1").feature("stat").set("solnum", "1");

  model.sol("sol1").feature().remove("s1");
  model.sol("sol1").feature().remove("v1");
  model.sol("sol1").feature().remove("st1");
  model.sol("sol1").create("st1", "StudyStep");
  model.sol("sol1").feature("st1").set("study", "std1");
  model.sol("sol1").feature("st1").set("studystep", "stat");
  model.sol("sol1").create("v1", "Variables");
  model.sol("sol1").feature("v1").set("control", "stat");
  model.sol("sol1").create("s1", "Stationary");
  model.sol("sol1").feature("s1").create("p1", "Parametric");
  model.sol("sol1").feature("s1").feature().remove("pDef");
  model.sol("sol1").feature("s1").feature("p1").set("control", "stat");
  model.sol("sol1").feature("s1").set("control", "stat");
  model.sol("sol1").feature("s1").create("seDef", "Segregated");
  model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").create("d1", "Direct");
  model.sol("sol1").feature("s1").feature("fc1").set("linsolver", "d1");
  model.sol("sol1").feature("s1").feature("fc1").set("dtech", "auto");
  model.sol("sol1").feature("s1").feature("fc1").set("maxiter", 50);
  model.sol("sol1").feature("s1").feature("fc1").set("initstep", 0.01);
  model.sol("sol1").feature("s1").feature("fc1").set("minstep", 1.0E-6);
  model.sol("sol1").feature("s1").feature().remove("fcDef");
  model.sol("sol1").feature("s1").feature().remove("seDef");
  model.sol("sol1").attach("std1");
  model.sol("sol1").runAll();

  model.result("pg1").run();

  model.label("Pellet 1D and bed reactor - 2nd try-rev.A1 (updated w_ergun).mph");

  model.result("pg1").run();
  model.result("pg1").setIndex("looplevel", "3", 0);
  model.result("pg1").setIndex("looplevel", "6", 0);
  model.result("pg1").setIndex("looplevel", "7", 0);
  model.result("pg1").run();
  model.result("pg1").set("xlabelactive", "on");
  model.result("pg1").set("xlabel", "z-coordinate (m)");
  model.result("pg1").set("ylabelactive", "on");
  model.result("pg1").set("ylabel", "Mass fraction (dim)");
  model.result("pg1").set("titletype", "manual");
  model.result("pg1").set("title", "Species mass fraction x Reactor lenght  [T=1100K]");
```

```
model.result("pg1").set("legendpos", "upperright");
model.result("pg1").run();
model.result("pg1").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").setIndex("looplevel", "4", 0);
model.result("pg2").setIndex("looplevel", "7", 0);
model.result("pg2").run();
model.result("pg2").run();
model.result("pg2").set("xlabelactive", "on");
model.result("pg2").set("ylabelactive", "on");
model.result("pg2").set("xlabel", "z-coordinate (m)");
model.result("pg2").set("ylabel", "Molar fraction (dim)");
model.result("pg2").set("titletype", "manual");
model.result("pg2").set("title", "Species molar fraction x Reactor length   [T=1100]");
model.result("pg2").run();
model.result("pg2").feature("lngr1").setIndex("legends", "xr_CH4", 0);
model.result("pg2").run();
model.result("pg2").feature("lngr2").setIndex("legends", "xr_CO", 0);
model.result("pg2").run();
model.result("pg2").feature("lngr3").setIndex("legends", "xr_CO2", 0);
model.result("pg2").run();
model.result("pg2").feature("lngr4").setIndex("legends", "xr_H2", 0);
model.result("pg2").run();
model.result("pg2").feature("lngr5").setIndex("legends", "xr_H2O", 0);
model.result("pg2").run();
model.result("pg1").run();
model.result("pg1").run();
model.result("pg1").feature("lngr1").setIndex("legends", "wr_CH4", 0);
model.result("pg1").run();
model.result("pg1").feature("lngr2").setIndex("legends", "wr_CO", 0);
model.result("pg1").run();
model.result("pg1").feature("lngr3").setIndex("legends", "wr_CO2", 0);
model.result("pg1").run();
model.result("pg1").feature("lngr4").setIndex("legends", "wr_H2", 0);
model.result("pg1").run();
model.result("pg1").feature("lngr5").setIndex("legends", "wr_H2O", 0);
model.result("pg1").run();
model.result("pg1").run();
model.result("pg4").run();
model.result("pg4").run();
model.result("pg4").run();
model.result("pg4").set("xlabelactive", "on");
model.result("pg4").set("xlabel", "r-coordinate (m)");
model.result("pg4").set("ylabelactive", "on");
model.result("pg4").set("ylabel", "concentration (mol/cm^3)");
model.result("pg4").run();
model.result("pg4").set("ylabel", "concentration (mol/cm3)");
model.result("pg4").run();
model.result("pg4").set("titletype", "manual");
model.result("pg4").set("ylabel", "concentration (mol/<sup>3</sup>)");
model.result("pg4").set("title", "Species concentrations x reactor lenght   [T=1100K]");
model.result("pg4").run();
model.result("pg4").run();
model.result("pg4").feature("lngr1").setIndex("legends", "cr_CH4", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr2").setIndex("legends", "cr_CO", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr3").setIndex("legends", "cr_CO2", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr4").setIndex("legends", "cr_H2", 0);
model.result("pg4").run();
model.result("pg4").feature("lngr5").setIndex("legends", "cr_H2O", 0);
model.result("pg4").run();
model.result("pg4").set("xlabel", "z-coordinate (m)");
model.result("pg4").run();
model.result("pg4").run();
model.result("pg5").run();
model.result("pg5").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").setIndex("looplevel", "7", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "5", 0);
model.result("pg7").run();
```

```
model.result("pg6").run();
model.result("pg8").run();
model.result("pg8").run();
model.result("pg9").run();
model.result("pg9").run();
model.result("pg10").run();
model.result("pg8").run();
model.result("pg3").run();
model.result("pg3").set("edges", "on");
model.result("pg3").set("xlabelactive", "on");
model.result("pg3").set("xlabel", "Reactor length, z");
model.result("pg3").set("ylabelactive", "on");
model.result("pg3").set("ylabel", "Pellet radius, r");
model.result("pg3").run();
model.result("pg3").set("titletype", "manual");
model.result("pg3").set("title", "Molar fraction of H2O (dim) x Pellet radius (mm) x Reactor length (m)");
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").feature("surf2").active(true);
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result("pg3").run();
model.result().duplicate("pg12", "pg3");
model.result("pg12").run();
model.result("pg12").run();
model.result("pg12").feature().remove("surf2");
model.result("pg12").feature().remove("surf3");
model.result("pg12").feature().remove("surf4");
model.result("pg12").feature().remove("surf5");
model.result("pg12").run();
model.result("pg12").label("2D Plot Group xH2O");
model.result("pg3").run();
model.result().duplicate("pg13", "pg3");
model.result("pg13").run();
model.result("pg13").run();
model.result("pg13").feature().remove("surf1");
model.result("pg13").feature().remove("surf3");
model.result("pg13").feature().remove("surf4");
model.result("pg13").feature().remove("surf5");
model.result("pg13").run();
model.result("pg13").setIndex("looplevel", "2", 0);
model.result("pg13").run();
model.result("pg13").setIndex("looplevel", "3", 0);
model.result("pg13").run();
model.result("pg13").setIndex("looplevel", "7", 0);
model.result("pg13").run();
model.result("pg13").set("allowtableupdate", false);
model.result("pg13").feature("surf2").set("rangeunit", "1");
model.result("pg13").feature("surf2").set("rangecolormin", 0.001796720535841954);
model.result("pg13").feature("surf2").set("rangecolormax", 0.05701664520899866);
model.result("pg13").feature("surf2").set("rangecoloractive", "off");
model.result("pg13").feature("surf2").set("rangedatamin", 0.001796720535841954);
model.result("pg13").feature("surf2").set("rangedatamax", 0.05701664520899866);
model.result("pg13").feature("surf2").set("rangedataactive", "off");
model.result("pg13").feature("surf2")
     .set("rangeactualminmax", new double[]{0.001796720535841954, 0.05701664520899866});
model.result("pg13").feature("surf2").feature("hght1").set("scale", 0.1315405347422355);
model.result("pg13").feature("surf2").feature("hght1").set("scaleactive", false);
model.result("pg13").set("renderdatacached", false);
model.result("pg13").set("allowtableupdate", true);
model.result("pg13").set("renderdatacached", true);
model.result().table().create("evl2", "Table");
model.result().table("evl2").comments("Interactive 2D values");
model.result().table("evl2").label("Evaluation 2D");
model.result().table("evl2")
     .addRow(new double[]{0.0025236814581318383, 0.0034461390770241453, 0.047501520577070855});
model.result().table("evl2")
     .addRow(new double[]{0.0020356523955896546, 0.0028381047831523504, 0.047344809753300056});
model.result().table("evl2")
     .addRow(new double[]{0.001302150881663249, 0.0018189517526305795, 0.046399070121349084});
model.result("pg13").set("view", "new");
model.result("pg13").run();
model.result("pg13").run();
model.result("pg13").run();
model.result("pg13").run();
model.result("pg13").feature("surf2").set("titletype", "auto");
model.result("pg13").feature("surf2").set("descractive", "off");
model.result("pg13").run();
```

```
model.result("pg13").set("view", "auto");
model.result("pg13").set("showhiddenobjects", "off");
model.result("pg13").run();
model.result("pg13").feature("surf2").set("rangedataactive", "on");
model.result("pg13").feature("surf2").set("rangedatamin", "0.00179672053584195");
model.result("pg13").feature("surf2").set("rangedataactive", "off");
model.result("pg13").run();
model.result("pg13").set("allowtableupdate", false);
model.result("pg13").feature("surf2").set("rangeunit", "1");
model.result("pg13").feature("surf2").set("rangecolormin", 0.001796720535841954);
model.result("pg13").feature("surf2").set("rangecolormax", 0.05701664520899866);
model.result("pg13").feature("surf2").set("rangecoloractive", "off");
model.result("pg13").feature("surf2").set("rangedatamin", 0.001796720535841954);
model.result("pg13").feature("surf2").set("rangedatamax", 0.05701664520899866);
model.result("pg13").feature("surf2").set("rangedataactive", "off");
model.result("pg13").feature("surf2")
     .set("rangeactualminmax", new double[]{0.001796720535841954, 0.05701664520899866});
model.result("pg13").feature("surf2").feature("hght1").set("scale", 0.1315405347422355);
model.result("pg13").feature("surf2").feature("hght1").set("scaleactive", false);
model.result("pg13").set("renderdatacached", false);
model.result("pg13").set("allowtableupdate", true);
model.result("pg13").set("renderdatacached", true);
model.result("pg13").run();
model.result("pg6").run();
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "1", 0);
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "1", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "1", 0);
model.result("pg7").run();
model.result("pg7").set("titletype", "manual");
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=500K]");
model.result("pg7").set("xlabelactive", "on");
model.result("pg7").set("xlabel", "z-coordinate (m)");
model.result("pg7").set("ylabelactive", "on");
model.result("pg7").set("ylabel", "Effectiveness factor (dim)");
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "2", 0);
model.result("pg7").run();
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=600K]");
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "3", 0);
model.result("pg7").run();
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=700K]");
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "4", 0);
model.result("pg7").run();
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=800K]");
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "5", 0);
model.result("pg7").run();
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=900K]");
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "6", 0);
model.result("pg7").run();
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=1000K]");
model.result("pg7").run();
model.result("pg7").set("title", "Effectiveness factor x reactor length   [T=1100K]");
model.result("pg7").run();
model.result("pg7").setIndex("looplevel", "7", 0);
model.result("pg7").run();

model.label("Pellet 1D and PBR - Final.mph");

model.result("pg7").run();
```

```
    return model;
  }

  public static void main(String[] args) {
    Model model = run();
    model = run2(model);
    model = run3(model);
    model = run4(model);
    model = run5(model);
    model = run6(model);
    model = run7(model);
    model = run8(model);
    run9(model);
  }

}
```

8. References

[1] Thiele, E. (1939). Relation between catalytic activity and size of particle. *Industrial & Engineering Chemistry*, *31*(7), 916-920.

[2] Aris, R. (1957). On shape factors for irregular particles—I: The steady state problem. Diffusion and reaction. *Chemical Engineering Science*, *6*(6), 262-268.

[3] Fogler, H. S. (1999). Elements of chemical reaction engineering.

[4] Wakao, N., & Smith, J. M. (1962). Diffusion in catalyst pellets. *Chemical Engineering Science*, *17*(11), 825-834.

[5] Chen, O. T., & Rinker, R. G. (1979). Modification of the dusty-gas equation to predict mass transfer in general poruos media. *Chemical Engineering Science*,*34*(1), 51-61.

[6] Coppens, M. O., & Froment, G. F. (1995). Diffusion and reaction in a fractal catalyst pore—I. Geometrical aspects. *Chemical engineering science*, *50*(6), 1013-1026.

[7] Coppens, M. O., & Froment, G. F. (1995). Diffusion and reaction in a fractal catalyst pore—II. Diffusion and first-order reaction. *Chemical engineering science*, *50*(6), 1027-1039.

[8] Keil, F. J. (1999). Diffusion and reaction in porous networks. *Catalysis Today*,*53*(2), 245-258.

[9] Dadvar, M., & Sahimi, M. (2007). The effective diffusivities in porous media with and without nonlinear reactions. *Chemical engineering science*, *62*(5), 1466-1476.

[10] Iordanidi, A. (2002). *Mathematical modeling of catalytic fixed bed reactors*. Twente University Press.

[11] Wesenberg, M. H., & Svendsen, H. F. (2007). Mass and heat transfer limitations in a heterogeneous model of a gas-heated steam reformer. *Industrial & engineering chemistry research*, *46*(3), 667-676.

[12] Xu, J., & Froment, G. F. (1989). Methane steam reforming: II. Diffusional limitations and reactor simulation. *AIChE Journal*, *35*(1), 97-103.

[13] Krishna, R., & Wesselingh, J. A. (1997). The Maxwell-Stefan approach to mass transfer. *Chemical Engineering Science*, *52*(6), 861-911.

[14] Maxwell, J. C. (1860). On the dynamical theory of gases. *Philosophical transactions of the Royal Society of London*, 49-88.

[15] Cussler, E. L. (2009). *Diffusion: mass transfer in fluid systems*. Cambridge university press.

[16] Jackson, R. (1977). *Transport in porous catalysts* (Vol. 4). Elsevier Science & Technology.

[17] Mason, E. A., Malinauskas, A. P. (1983). *Gas transport in porous media: the dusty-gas model* (Vol. 17). Elsevier Science Ltd.

[18] Veldsink, J. W., Van Damme, R. M. J., Versteeg, G. F., & Van Swaaij, W. P. M. (1995). The use of the dusty-gas model for the description of mass transport with chemical reaction in porous media. *The Chemical Engineering Journal and the Biochemical Engineering Journal*, *57*(2), 115-125.

[19] Abashar, M. E., Elnashaie, S. S. (1993). Mathematical modelling of diffusion-reaction, and solution algorithm for complex reaction networks in porous catalyst pellets—steam reforming of natural gas. *Mathematical and computer modelling*, *18*(7), 85-100.

[20] Solsvik, J., Jakobsen, H. A. (2011). Modeling of multicomponent mass diffusion in porous spherical pellets: Application to steam methane reforming and methanol synthesis. *Chemical Engineering Science*, *66*(9), 1986-2000.

[21] Wesenberg, M. H. (2006). *Gas heated steam reformer modelling*. Fakultet for naturvitenskap og teknologi.

[22] Elnashaie, S. S. E. H., & Abashar, M. E. E. (1993). Steam reforming and methanation effectiveness factors using the dusty gas model under industrial conditions. *Chemical Engineering and Processing: Process Intensification*,*32*(3), 177-189.

[23] Behnam, M., Dixon, A. G., Wright, P. M., Nijemeisland, M., & Stitt, E. H. (2012). Comparison of CFD simulations to experiment under methane steam reforming reacting conditions. *Chemical Engineering Journal*, *207*, 690-700.

[24] Troupel, M. A. (2009). *CFD study of the intra and inter particles transport phenomena in a fixed-bed reactor* (Doctoral dissertation, Worcester Polytechnic Institute).

[25] Vafai, K. (Ed.). (2005). *Handbook of porous media*. Crc Press.

[26] Whitaker, S. (1987). Mass transport and reaction in catalyst pellets. *Transport in porous media*, *2*(3), 269-299.

[27] Bird, R.B., Stewart, W.E., Lightfoot, E.N., 2002. *Transport Phenomena*, second ed. Wiley, New York.

[28] Knudsen, M., 1950. The Kinectic Theory of Gases, Mathuenm, London, 3rd ed.

[29] Webb, S. W., Ho, C. K., 2006, *Gas Transport in Porous Media*, 20 Ed., Springer, New York

[30] Poling, B. E., Prausnitz, J. M., & O'connell, J. P. (2001). *The properties of gases and liquids* (Vol. 5). New York: McGraw-Hill.

[31] Chapman, S., & Cowling, T. G. (1970). *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press.

[32] Marrero, T. R., & Mason, E. A. (1972). Gaseous diffusion coefficients. *Journal of Physical and Chemical Reference Data*, *1*(1), 3-118.

[33] Kee, R. J., Coltrin, M. E., & Glarborg, P. (2005). *Chemically reacting flow: theory and practice*. John Wiley & Sons.

[34] Brokaw, R. S. (1969). Predicting transport properties of dilute gases. *Industrial & Engineering Chemistry Process Design and Development*, *8*(2), 240-253.

[35] Nield, D. A., & Bejan, A. Convection in Porous Media, Springer, New York, 1999.

[36] Jakobsen, H. A. Chemical Reactor Modeling - Multiphase Reactive Flows, p.306-311, Springer, New York, 2008.

[37] Froment, G. F., Bischoff, K. B., & De Wilde, J. (1990). Chemical reactor analysis and design (Vol. 2). New York: Wiley.

[38] Xu, J., & Froment, G. F. (1989). Methane steam reforming, methanation and water-gas shift: I. Intrinsic kinetics. *AIChE Journal*, 35(1), 88-96.

[39] Latham, D. (2009). Mathematical Modelling of an Industrial Steam Methane Reformer.