REU 2005 Summer Program

Catalytic Pellet Reactor Design under Uncertainty

Student's Name: Celia Xue

Advisors: Professor Andreas Linninger
Dr. Libin Zhang

Laboratory for Product and Process Design
University of Illinois at Chicago
LPPD-Project Final Report: Aug.5th, 2005

Abstract

Design of catalytic pellet reactor is a challenging task because of the complicated mechanism and transport phenomenon inherent to this problem. Process uncertainties such as the feed flow rate, porosity of the pellet, reaction rate coefficient, etc. present additional complexity. This paper investigates the method to quantify uncertainties of design parameters in the case study of catalytic pellet reactor design.

The case study starts with developing models for a single catalytic pellet. Concentration and temperature profiles are solved numerically by collocation method for isothermal and non-isothermal pellets under steady and dynamic conditions. To account for the mass diffusion effect in the rate equation, the effectiveness factor η is introduced and solved for steady state pellets. Its sensitivity with respect to pellet surface conditions and other physical parameters are extensively studied in this paper. To solve for temperature and concentration profiles inside a reactor, reactor and pellet governing equations need to be solved simultaneously. With collocation method, we are able to obtain concentration and temperature profiles for the reactor successfully.

Two important packed bed catalytic reactor behaviors concern us: "hotspot" and multiplicity. The multiple steady states in pellets are observed and analyzed in this paper. The reactor multiplicity is also studied combing the observation of "hotspots". Finally, we are able to demonstrate with a specific case that uncertainty presented in design parameters leads to "hotspots" and possible melt down of the reactor. Uncertainty tests should be extensively studied along with the design of a reactor.

Table of Content

1. Introduction

Parameters such as the properties of the chemicals, inlet flow rate, and reaction coefficient etc. present uncertainties in their values. The uncertainties in these parameters directly influence many aspects of our design such as the safety condition or the product quality. The traditional method to handle process uncertainties is to over-design the process so that a “safety factor” is incorporated. However, such traditional method often fails to predict the magnitude of safety factor and to provide insight of the processes. The lack of understanding uncertainties in design parameters often lead to loss of revenue or unsafe design. Quantifying the uncertain parameters will enable engineers to rigorously describe the safety condition of their design, guarantee product quality and other operating conditions. The importance of studying uncertainty is demonstrated by the case study of a catalytic pellet reactor design.

Catalytic pellet reactor has commercial applications such as reducing automotive emission gas, oxidation of hydrocarbons, paraffin dehydrogenation, hydrocracking, and dehydrocyclization etc. In designing a pellet reactor, packed-bed geometry with stationary spherical catalyst pellets shown in Fig.1 is usually adopted. The boundary conditions for an individual pellet are described by its surface concentration and temperature, whereas the reactor boundary conditions are described by inlet concentration ad temperature. The pellets can be treated as a porous media, where reactant diffuses axially along the reactor, and also radially into the pellets. To fully describe a reactor, several design unknowns require extensive attention: concentration profile and temperature profile inside the reactor. These two design unknowns are directly related to the conditions of individual pellets by pellet boundary conditions and effectiveness factor η. As we focus our attention on solving pellets shown Fig. 1, catalytic pellets are porous to allow diffusion of reactant, consequently creating a concentration gradient with respect to the radius. Discussed in Weisz and Hicks [4], if the chemical reaction is accompanied with a heat effect, significant temperature gradient also develops within the pellet. In such a case, concentration and temperature profiles of the pellet must be solved simultaneously.

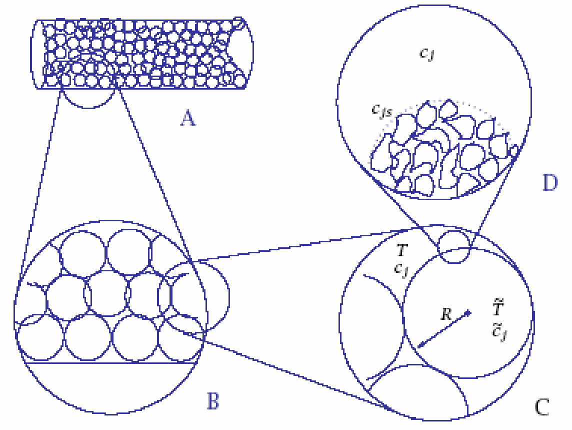

Fig. 1 Expanded view of a fixed-bed pellet reactor

2. Development of Mathematical Model

In this case study, SO_2 oxidation catalyst for production of sulfuric acid is chosen. This reaction as expressed in Eq. (1) is first order with respect to SO_2.

$$SO_{2\,(g)}+1/2O_{2(g)} \leftrightarrow SO_{3\,(g)} \tag{1}$$

SO_3 gas continues to react with water in later processes to form sulfuric acid. The physical parameters of the pellets are shown in Appendix.

To account for diffusion effect inside individual pellets, a variable known as the internal effectiveness factor η is introduced in Weisz and Hicks [4]. Its definition is expressed below:

$$\eta = \frac{actual\ reaction\ rate}{reaction\ rate\ at\ \ pellet\ surface} \tag{2}$$

The expression of rate of reaction incorporating Eq. (2) have been derived by Damkohler [5], Thiele [6], Zeldowitsch [7], and Wheeler [8], leading to the simplified form shown in Eq. (3) for a first order reaction.

$$\mathrm{Re}action\ Rate = \eta k(T)C_A \tag{3}$$

Where *k(T)* is the reaction rate coefficient and C_A is the concentration of reactant A. The reaction coefficient for non-isothermal pellet can be expressed with Arrhenius equation [2] shown in Eq. (4). For isothermal pellet, it is assumed to be a constant. Refer to Nomenclature section for parameter definitions.

$$k(T) = k_{ref}\exp\left[-\frac{E}{RT_{ref}}\left(\frac{T_{ref}}{T}-1\right)\right] \tag{4}$$

2.1 Individual Pellet at Steady State

Isothermal Condition

An isothermal pellet only presents mass transfer within the solid. Simplify the expression of a first order reaction to Eq.(5), a mass balance on the spherical shell control volume shown by Fig. 2 can be done easily.

$$A \xrightarrow{catalyst} B \tag{5}$$

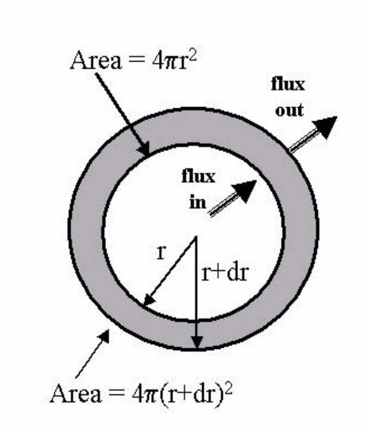

Fig. 2 Control volume of a spherical pellet

The concentration profile with respect to the radius of the pellet is then expressed in Eq.(6).

$$\frac{d^2C_A}{dr^2}+\frac{2}{r}\frac{dC_A}{dr}-\frac{kC_A}{D_e}=0 \tag{6}$$

D_e is diffusivity coefficient of the pellet. This equation is easily solved analytically.

Non-isothermal Condition

For non-isothermal condition, the temperature profile inside the pellet needs to be developed along with concentration profile. By doing an energy balance on the same control volume, the system of differential equations is shown below:

$$\frac{dC_A^{\ 2}}{d^2r}+\frac{2}{r}\frac{dC_A}{dr}-\frac{k(T)C_A}{D_e}=0 \tag{7}$$

$$\frac{dT^2}{d^2r}+\frac{2}{r}\frac{dT}{dr}+\frac{(-\Delta H)}{\lambda}k(T)C_A=0 \tag{8}$$

Notice that reaction coefficient *k* is no longer a constant. Shown in Villadsen and Michelsen [1], converting Eq.(7) and (8) into dimensionless form and inserting Arrhenius equation for *k(T)*, we obtain:

$$\frac{d^2\varphi}{d\theta^2}+\frac{2}{\theta}\frac{d\varphi}{d\theta}-\Phi^2\exp\left[\gamma\left(1-\frac{1}{\zeta}\right)\right]\varphi=0 \tag{9}$$

$$\frac{d^2\zeta}{d\theta^2}+\frac{2}{\theta}\frac{d\zeta}{d\theta}-\beta\exp\left[\gamma\left(1-\frac{1}{\zeta}\right)\right]\varphi=0 \tag{10}$$

Thiele Modulus: $\Phi^2=\frac{R^2}{D_e}\left[a\exp(-\frac{E_a}{RT_s})\right]$ (11)

$$\varphi = \frac{C_A}{C_{ref}}, \theta = \frac{r}{R}, \zeta = \frac{T}{T_{ref}}, \gamma = \frac{E_a}{RT}, \beta = \frac{\Delta H k_s}{\lambda} \tag{12}$$

a is a dimensionless variable that accounts for reference reaction rate coefficient. Shown in Eq. (12), these dimensionless variables were discussed by Pita et al. [9]. C_{ref} and T_{ref} are taken to be the concentration of reactant and temperature on the surface of the pellet. From here on, we will directly indicate reference conditions to be surface conditions, which are indicated by subscript *s*. *R* is the radius of the pellet. The boundary conditions for the dimensionless equations are:

$$[\zeta]_{\theta=1} = \frac{T}{T_s} = 1 \text{ and } \left[\frac{d\zeta}{d\theta}\right]_{\theta=0} = 0 \tag{13}$$

$$[\varphi]_{\theta=1} = \frac{C_A}{C_{As}} = 1 \text{ and } \left[\frac{d\varphi}{d\theta}\right]_{\theta=0} = 0 \tag{14}$$

At the surface of the pellet, reactant concentration and temperature are equal to surface conditions. At the center of the pellet, we assume the concentration and temperature are constant. Eq. (9) and (10) form a system of coupled second order non-linear differential equations. Numerical method must be employed to solve for φ and ζ.

2.2 Individual Pellet at Dynamic State

To calculate the pellet concentration and temperature profile at dynamic state, replace the right side of steady state equations with time variations:

$$\frac{d^2\varphi}{d\theta^2} + \frac{2}{\theta}\frac{d\varphi}{d\theta} - \Phi^2 \exp\left[\gamma(1-\frac{1}{\zeta})\right]\varphi = \frac{d\varphi}{dt} \tag{15}$$

$$\frac{d^2\zeta}{d\theta^2} + \frac{2}{\theta}\frac{d\zeta}{d\theta} - \beta \exp\left[\gamma(1-\frac{1}{\zeta})\right]\varphi = \frac{d\zeta}{dt} \tag{16}$$

$$\text{Initial Conditions: } \left[\frac{d\zeta}{dt}\right]_{t=0} = \frac{T_o}{T_s}, \text{ and } \left[\frac{d\varphi}{dt}\right]_{t=0} = \frac{C_o}{C_{As}} \tag{17}$$

T_o and C_o are variables to represent initial conditions for dynamic state at each radial position. Boundary conditions for Eq.(15) and (16) and are similar to those of steady state pellet governing equations shown in Eq. (13) and (14). We are free to choose the values of T_o and C_o. By choosing different initial profiles, we can observe how the pellets approach steady state on a time scale.

2.3 Effectiveness Factor

Derived by Villadsen and Michelsen [1] , η is expressed as the ratio of volume averaged reaction rate relative to the rate at surface temperature and concentration.

$$\eta = \frac{\int_0^1 \Phi^2 rate(\varphi,\zeta) dr^{s+1}}{\int_0^1 \Phi^2 rate(1,1) dr^{s+1}} = \int_0^1 rate(\varphi,\zeta) dr^{s+1} \qquad (18)$$

Inserting dimensionless first order rate equation into the equation above, we obtain the expression of effectiveness factor:

$$\eta = 3\int_0^1 \varphi(\theta) \exp\left[\gamma(1-\frac{1}{\zeta}) \right] \theta^2 d\theta \qquad (19)$$

If we take into account the diffusion effect from the bulk of the reactor to the surface of the pellet, we can express effectiveness factor using bulk concentrations and temperatures. Neglecting intraparticle temperature gradient, the resulting expression is shown in Eq. (20).

$$\eta = \frac{3Bi_M}{{\Phi_1}^2} \frac{\Phi_1 \coth \Phi_1 - 1}{\Phi_1 \coth \Phi_1 + Bi_M - 1} \qquad (20)$$

$${\Phi_1}^2 = \Phi^2 \exp\left[\gamma(1-\frac{1}{\zeta}) \right] \qquad (21)$$

Where Bi_M is the Biot number for mass transfer. Eq. (20) uses an averaged Biot number to compensate the use of an averaged dimensionless temperature.

Discussed in Pita et al. [9], for a specified set of parameters such as β, γ and Φ, there may exist more than one feasible steady state. All variables however, satisfy the governing equation (9) and (10), with boundary conditions (13) and (14). Shown in the form of effectiveness factor, one set of β, γ and Φ may correspond to multiple effectiveness factors. Also shown by Lee et al. [11], uncertainty in parameters may initiate the switching between one steady state to another.

2.4 Reactor Design at Steady State

By doing a mass balance and energy balance on a cylindrical control volume of the reactor shown in Fig. 3, the expression for concentration and temperature is derived in Fogler [2], shown in Eq. (22) and (23).

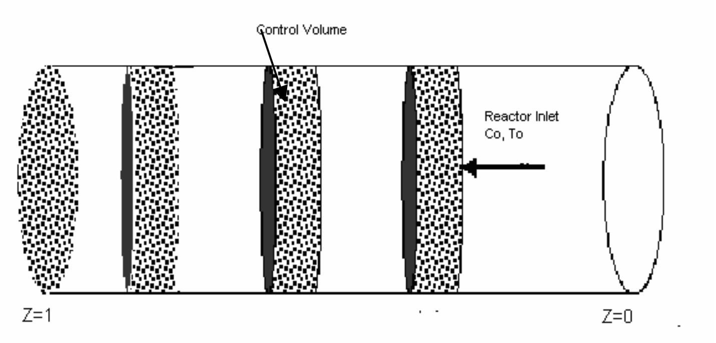

Fig. 3 Control Volume of a pellet reactor

$$D_{AB}\frac{d^2C_A}{dz^2} - U\frac{dC_A}{dz} + r'_A\,\rho_b = 0 \tag{22}$$

$$k_e\frac{d^2T}{dz^2} - \rho_g C_p U\frac{dT}{dz} + r'_A\,\Delta H = 0 \tag{23}$$

Where r_A' is the overall rate of reaction within the catalyst per unit mass of catalyst. With the expression of η derived in Eq. (20), overall rate of conversion can be expressed as Eq.(24).

$$-r_A' = \eta k(T) S_a C_A \tag{24}$$

S$_a$ is the internal surface area of the pellet. For the convenience of calculation, we prefer to work with scaled equations incorporating expression for *k(T)*. Convert Eq.(22) and Eq. (23) into scaled form with respect to inlet conditions, the resulting expression are shown Eq. (25) and (26).

$$\frac{d^2C_r}{dz^2} - \frac{U}{D_{AB}}\frac{dC_r}{dz} - \frac{\eta S_a k_o \rho_b}{D_{AB}} C_r \exp\left[\gamma(1-\frac{1}{T_r})\right] = 0 \tag{25}$$

$$\frac{d^2T_r}{dz^2} - \frac{\rho_g C_p U}{k_b} dT_r - \frac{\Delta H \eta S_a C_i \rho_b}{k_b T_i} C_r \exp\left[\gamma(1-\frac{1}{T_r})\right] = 0 \tag{26}$$

$$T_r = \frac{T}{T_i}\ ,\ C_r = \frac{C_A}{C_i} \tag{27}$$

The boundary conditions are shown in Eq.(28) and (29).

$$[C_r]_{z=0} = 1 \quad [T_r]_{z=0} = 1 \tag{28}$$

$$\left[\frac{dC_r}{dz}\right]_{z=1} = 0 \quad \left[\frac{dT_r}{dz}\right]_{z=1} = 0 \tag{29}$$

T_i and C_i represent inlet conditions and C_r, T_r represent scaled concentration and temperature profile along the axial direction of the reactor. At the inlet of the reactor, the temperature and concentration are given to be inlet conditions. And for a sufficiently long reactor, the concentration and temperature at the end of the reactor can be assumed constant. The design of the reactor is coupled with solving pellet governing equations because effectiveness factor is a function of pellet temperature profile. Further more, pellet surface conditions are replaced by the bulk conditions inside the reactor. In order to solve reactor profiles, Eq. (9), (10), (20), (25) and (26) need to be solved simultaneously. The boundary conditions are listed in Eq. (13), (14), (28) and (29) with T_s, C_s replaced by T_r, C_r.

Industrial packed-bed reactors usually adopt cooling process in addition to the reactor to remove the heat produced. This application requires us subtracting a "heat removal" term from the energy balance shown in Eq.(26). Heat removal produces the effect of "hotspot" for typical strongly exothermic reactors. Discussed in Froment et al. [3] and Syed et al. [10], the magnitude of hot spot depends on the heat of reaction, rate of reaction, inlet conditions etc.

2.5 Reactor Design at Dynamic State

With the steady state reactor governing equations defined, their dynamic forms are easily obtained by adding time variation:

$$\frac{d^2C_r}{dz^2}-\frac{U}{D_{AB}}\frac{dC_r}{dz}-\frac{\eta S_a k_o \rho_b}{D_{AB}}C_r\exp\left[\gamma(1-\frac{1}{T_r})\right]=\frac{dC_r}{dt} \tag{30}$$

$$\frac{d^2T_r}{dz^2}-\frac{\rho_g C_p U}{k_b}dT_r-\frac{\Delta H\eta S_a C_i \rho_b}{k_b T_i}C_r\exp\left[\gamma(1-\frac{1}{T_r})\right]=\frac{dT_r}{dt} \tag{31}$$

$$\text{Initial Conditions: }\left[\frac{dT_r}{dt}\right]_{t=0}=\frac{T_o}{T_i}\text{ and }\left[\frac{dC_r}{dt}\right]_{t=0}=\frac{C_o}{C_i} \tag{32}$$

Boundary conditions for Eq. (30) and (31) are identical to those of steady state reactor governing equations.

3. Methodology

The numerical method chosen to solve systems of second order differential equations is collocation method, a type of weight residual method (WRM). Weight residual method assumes that the analytical solution can be approximated in a piecewise fashion by a superposition of functions with unknown coefficients a_j shown in Eq.(33).

$$T_{approximate}(x,t)=\sum_{j=1}^{n}a_j(t)\phi_j(x) \tag{33}$$

Shown in Fig. 4, solving for x at each node we choose and connect all the nodes together linearly, we obtain an approximated function.

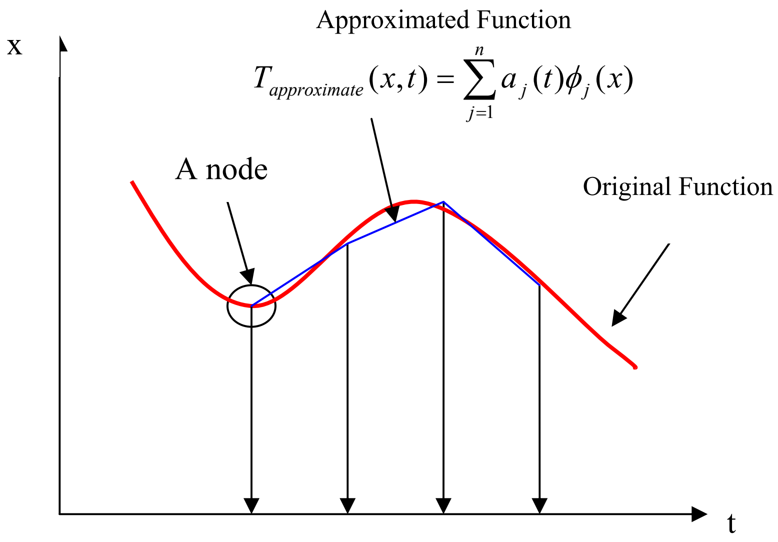

Fig. 4 Schematic of weight residual method

We expand Eq. (33) into a set of base (or trial) functions expressed as φ_j in Eq. (34). T_o(x,y,z,t) is chosen so that the boundary conditions are satisfied.

$$T(x,y,z,t)=T_0(x,y,z,t)+\sum_{j=1}^{n}a_j(t)\phi_j(x,y,z) \tag{34}$$

Weight residual method approximates the analytical solution by choosing coefficients a_j so that the difference between analytical solution and estimated solution (residual, R) is minimized. Express this condition using integral form:

$$\iiint W_m(x,y,z)R(x,y,z,t)dxdydz=0,\quad m=1,2,\cdots,M \tag{35}$$

Where W is a set of weight functions which are used to evaluate equation above.

Collocation method chooses the weight function to be Dirac delta functions show in Eq.(36) such that the residual at each chosen node is zero. Characteristic of collocation method is that it forces the residual to be zero at the nodes chosen, and has no control on the residual between nodes.

$$\begin{aligned} W_m\left(\vec{x}\right)&=\delta\left(\vec{x}-\vec{x}_m\right)\\ \iiint W_m\left(\vec{x}\right)R\left(\vec{x}\right)d\Omega&=\iiint\delta\left(\vec{x}-\vec{x}_m\right)R(\vec{x})d\Omega\\ &=R\left(\vec{x}_m\right)=R\left(x_m,y_m,z_m\right)\end{aligned} \tag{36}$$

As an example, we will use Eq.(9) and (10) with boundary conditions in Eq.(12) to show how collocation method solves second order nonlinear differential equations. The most essential step is to discretize the differential equations and boundary conditions using Lagrange polynomial (chosen for this case study). The expression of Lagrange polynomials is shown below:

$$y(x)=\sum_{i=0}^{m} l_i(x)\ \mathrm{y_i} \quad with \quad l_i(x)=\prod_{\substack{j=0\\ j\neq i}}^{N}\frac{r-r_j}{r_i-r_j} \tag{37}$$

Its first order derivative and second order derivative are solved to be:

$$y'(r)\big|_{x=x_1}=\frac{dy(r)}{dr}=\sum_{i=0}^{N}\frac{d(l_i(r))}{dr}y_i=\sum_{i=0}^{N}l'_i(r)y_i \tag{38}$$

$$y''(r)\big|_{x=x_1}=\frac{d^2y}{dr^2}=\sum_{i=0}^{N}\frac{d^2(l_i(r))}{dr^2}y_i=\sum_{i=0}^{N}l''_i(r)y_i=\sum_{i=0}^{N}l''_{i,x_1}y_i \tag{39}$$

Inserting Lagrange polynomial and its derivatives into Eq. (9) and Eq. (10) for every appearance of unknown variables: φ and ζ, Eq. (9) and (10) are transformed into discretized form:

$$\sum_{i=0}^{N}\frac{d^2(l_i(\theta_j))}{d\theta^2}\varphi_i+\left(\frac{2}{\theta_j}\sum_{i=0}^{N}\frac{d(l_i(\theta_j))}{d\theta}\varphi_i\right)-\Phi^2\exp\left[-\gamma\left(\frac{1}{\zeta_j}-1\right)\right]\varphi_j\ =\ 0 \tag{40}$$

$$\sum_{i=0}^{N}\frac{d^2(l_i(\theta_j))}{d\theta^2}\zeta_i+\left(\frac{2}{\theta_j}\sum_{i=0}^{N}\frac{d(l_i(\theta_j))}{d\theta}\zeta_i\right)-\beta\exp\left[-\gamma\left(\frac{1}{\zeta_j}-1\right)\right]\varphi_j\ =\ 0 \tag{41}$$

Where j=1…N, and i=0…N. And boundary conditions are also transformed into:

$$\begin{aligned}&\sum_{i=0}^{N}\frac{d(l_i(\theta=0))}{d\theta}\varphi_i=0\\&\sum_{i=0}^{N}l_i(\theta=1)\varphi_i=\varphi_N=k_c(\varphi_N-1)\\&\sum_{i=0}^{N}\frac{d(l_i(\theta=0))}{d\theta}\zeta_i=0\\&\sum_{i=0}^{N}l_i(\theta-1)\zeta_i=1\end{aligned} \tag{42}$$

The discretized equations shown above can be solved in commercial available software. All numerical results in this report are obtained through software MATLAB.

4. Results

4.1 Individual Pellet at Steady State

Isothermal Condition

To solve for the concentration profile in Eq. (6) and effectiveness factor in Eq. (19) for an individual pellet, collocation method is employed. In order to demonstrate the accuracy of collocation method, it is programmed into MATLAB and the results are compared to those of analytical method. Fig. 5 shows that collocation method completely captures the accuracy of analytical solution. For an arbitrary physical condition (Thiele modulus=2), the percentage difference between η solved analytically and numerically is less than 0.1%.

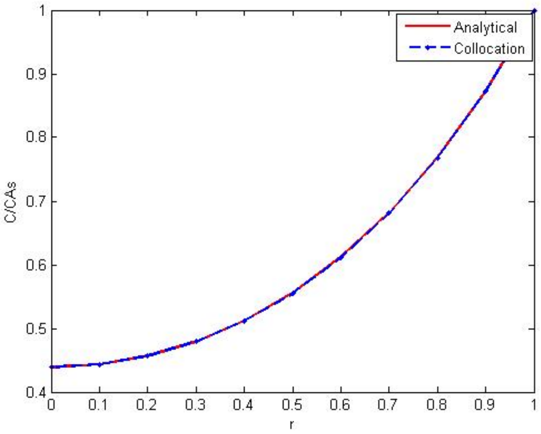

Fig. 5 Concentration profile for an isothermal Pellet.

Non-Isothermal Condition

For a non-isothermal pellet, temperature profile must be coupled with concentration profile. The system of differential equations in Eq.(9) and (10) are solved using collocation method. We choose five collocation points or radial position to develop numerical profiles. For an arbitrary physical condition of an exothermic reaction, the concentration and temperature profiles are plotted in Fig. 6. Notice the temperature is higher in the center of the pellet (r=0) due to the exothermic reaction chosen and decreases radially to satisfy the boundary condition in Eq.(27). Because of the non-linearity of the problem, initial guesses are especially important for solutions to converge.

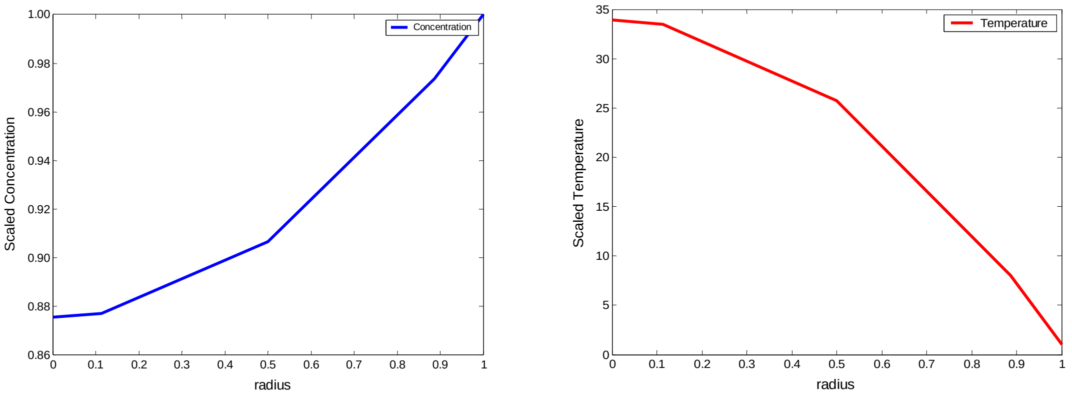

Fig. 6 Concentration profile (Left) and temperature profile (Right) for an arbitrary physical condition

Sensitivity of Effectiveness Factor

The sensitivity of effectiveness factor is tested by varying the target parameter while other parameters are fixed. Fig. 7 shows the behavior of effectiveness factor against pellet surface concentration C_{As} and surface temperature T_s.

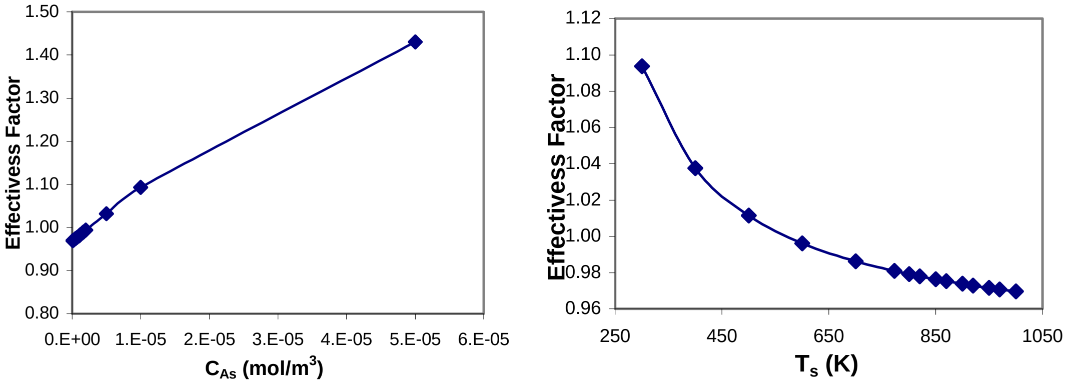

Fig. 7 Behavior of effectiveness factor with respect to surface concentration (left) and surface temperature (right).

As shown in Fig. 7, effectiveness factor increases as surface concentration increases and decreases as surface temperature increases. This behavior is predicted from the definition of η in Eq.(2): At high temperature, diffusion effect becomes the limiting factor toward the conversion rate and thus increases the denominator of η. On the other hand, increasing surface concentration minimizes the impact of diffusion and η behaves the opposite. A 3D plot of effectiveness factor with respect to T_s and C_{As} is also generated in Fig. 8 to predict the general trend of effectiveness factor.

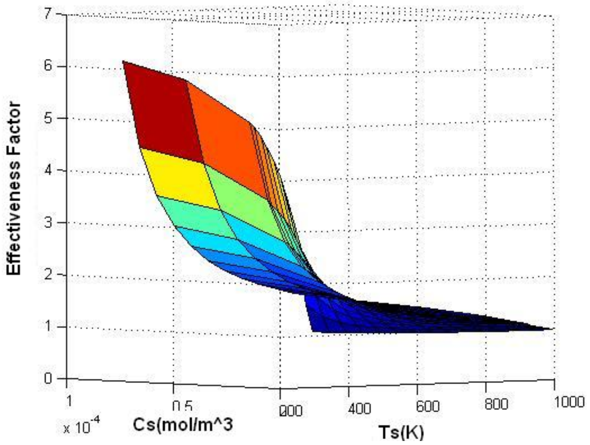

Fig. 8 Effectiveness factor vs. T_s and C_{As}

Continue to test η sensitivity toward physical properties of the pellet, we choose our variables to be diffusivity coefficient (D_e), heat conductivity (K_e), and finally reaction rate coefficient in the form of dimensionless variable a. The results in Fig. 9 and Fig. 10 are obtained using surface condition of C_{As}=2.46E-4 mol/m^3 and T_s=550K.

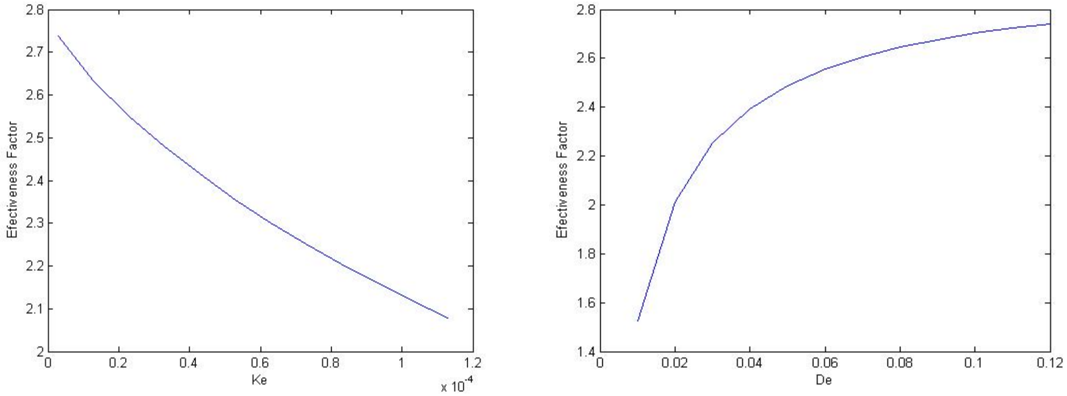

Fig. 9 Effectiveness factor vs. K_e (left) and D_e (right)

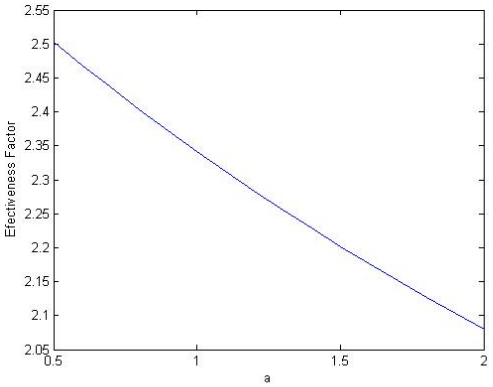

Fig. 10 Effectiveness factor vs. dimensionless rate coefficient α

The range of the sensitivity tests is approximately 100% above and below nominal value used in the case study. Effectiveness factor decreases as heat conductivity or α increase due to relative large limiting effect of diffusion. As diffusivity coefficient becomes larger, effectiveness factor is limited by surface reaction rate and thus increases.

4.2 Individual Pellet at Dynamic State

For dynamic condition, we need to solve Eq. (15) and Eq. (16) along with specified initial conditions. This task can be easily done in MATLAB using solver ODE45. Since the convergence of solution is very sensitive to the initial conditions, we choose the initial conditions for scaled concentration profile to be the result of the steady state and scaled temperature profile to be 1. Fig. 11 shows the dynamic concentration and temperature variation with respect to time for five collocation nodes *r* we chose to use.

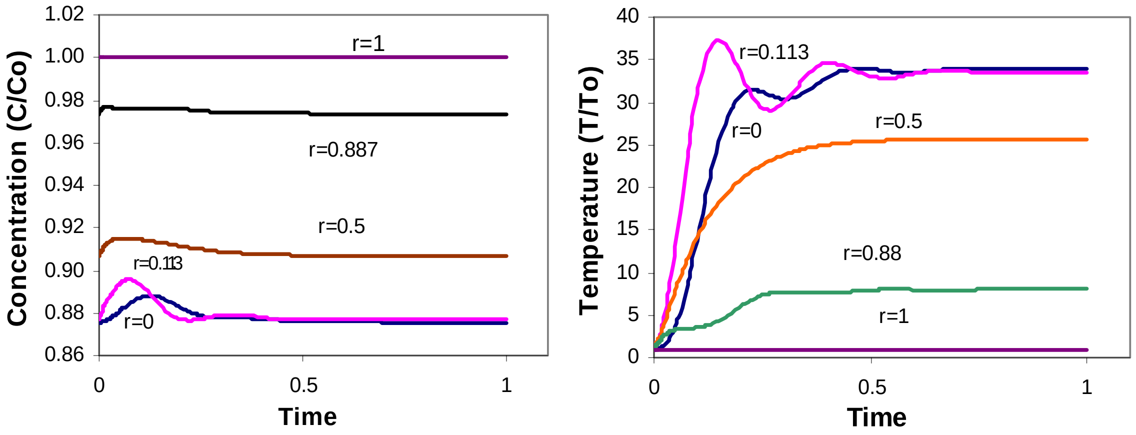

Fig. 11 Concentration (left) and Temperature (right) profile for a dynamic state pellet

The graphs show that our program successfully predicts the dynamic behavior of a pellet according. The x-axis in the graphs above is a dimensionless time scale. As time progresses, the concentration and temperature inside the pellet both converge to steady state conditions as predicted. The steady results above are obtained at T_s=550K and C_{As}=2.46E-4 mol/m^3.

Sensitivity of Dynamic Pellet

For the dynamic behavior of pellets, concentration and temperature profiles are obtained as shown in the last section. For practical purposes, the surface temperature and concentration around the pellets are both subject to disturbances with respect to time. Therefore the sensitivity test toward the dynamic pellet is aimed at those two variables: T_s and $C_{As.}$ Concentration and temperature inside a pellet are functions of both radial position and time. The pellet profile sensitivities with respect to surface concentration for a scaled radius of 0.113 are shown in Fig. 12. The nominal surface conditions are chosen to be: C_{As}=2.46E-6 mol/m^3, T_s= 550K.

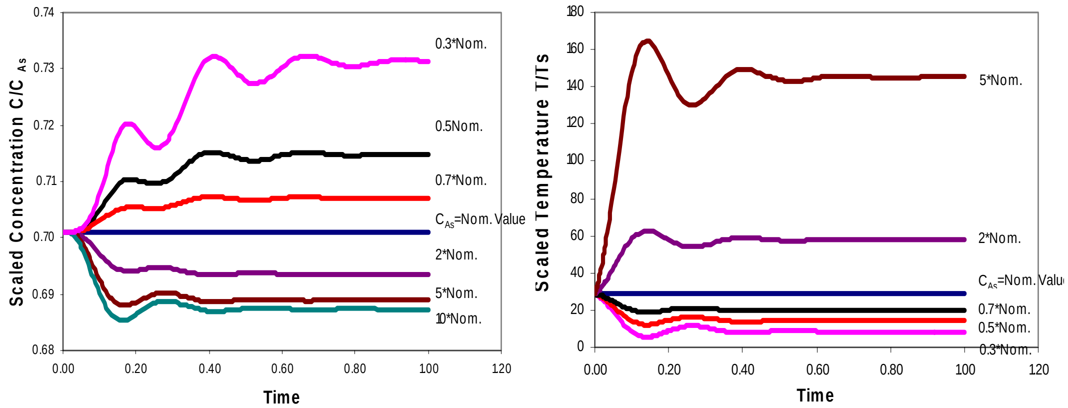

Fig. 12 Dynamic pellet concentration (left) and temperature (right) profile vs. C_{As} (Nominal Value=2.46E-6 mol/m^3)

From Fig. 12, temperature profile increases drastically as the inlet concentration increases. This is predicted as more reactant is introduced into the reactor, more heat is released due to the exothermic nature of the reaction. We also observe an increasing scaled concentration profile as we decrease the inlet concentration. Dynamic profile sensitivity against changing surface temperature is shown in Fig. 13.

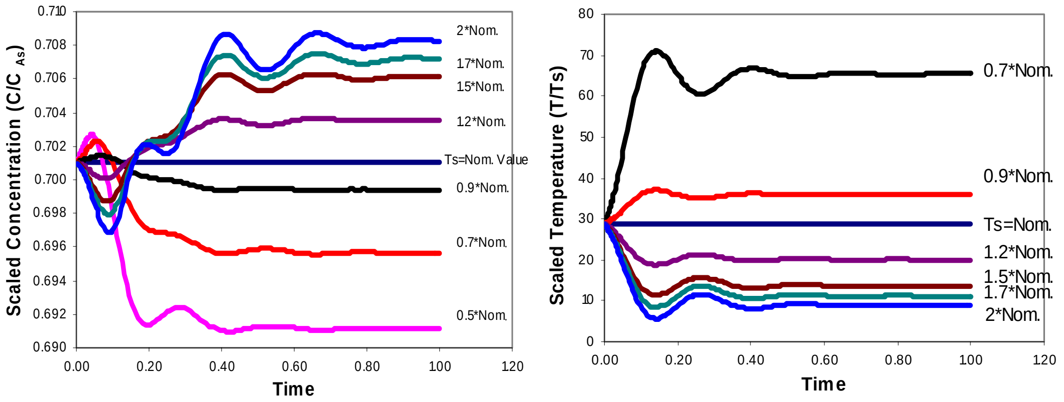

Fig. 13 Dynamic pellet concentration (left) and temperature (right) profile vs. T_s (Nominal Value=550K)

We observe from Fig. 13, the reactor concentration becomes increasingly unstable as we raise inlet temperature. For large increase in temperature, it takes the reactor much longer to return to steady state.

4.3 Reactor Design at Steady State

Eq. (25) and Eq. (26) present the governing equations for the reactor with boundary conditions described by Eq. (28). We choose five collocation points or axial position to obtain numerical profiles. Notice the effectiveness factor obtained from pellet profiles is used to solve reactor governing equation. Further more, the boundary conditions for the pellets are no long constant surface conditions, as they are now related to the bulk of the reactor. To obtain the profiles for the reactor, pellet and reactor governing equations have to be solved simultaneously. The four equations are coupled together through effectiveness factor and boundary conditions of the pellets.

The boundary conditions for the reactor are known inlet conditions: T_i and C_i. The program is written so that at each reactor collocation node (axial position, z), pellet profiles (with respect to five radial collocation node inside the pellet) are solved with one set of C_r and T_r, which are then associated with the rest of the reactor. The reactor profiles are then solved and displayed in Fig. 14. The results are generated with T_i=300K, C_i=1.6E-2 mol/m^3.

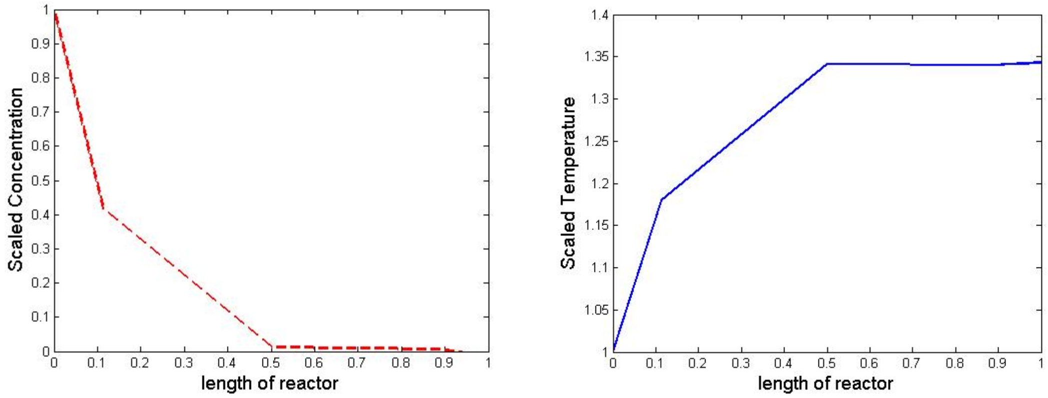

Fig. 14 Steady state concentration profile (left) and temperature profile (right) for coupled reactor design.

Due to an exothermic reaction chosen for the case study, the temperature rises with respect to the axial location of the reactor. With the inlet conditions chosen, the temperature increase across the reactor is around 90K.

Sensitivity of Steady State Reactor

As mentioned above if we want to observe how the reactor responds to uncertainties of different parameters, it is necessary for us to do sensitivity test of reactor profiles against selected variables. For practical purposes, inlet conditions are subject to changes. The inlet temperature fluctuation is from 270K to 390K, a reasonable industrially applicable range. Again, the nominal value chosen for our reactor is T_i=300K, C_i=1.6E-2 mol/m^3. The reactor sensitivity toward inlet concentration is shown in Fig. 15.

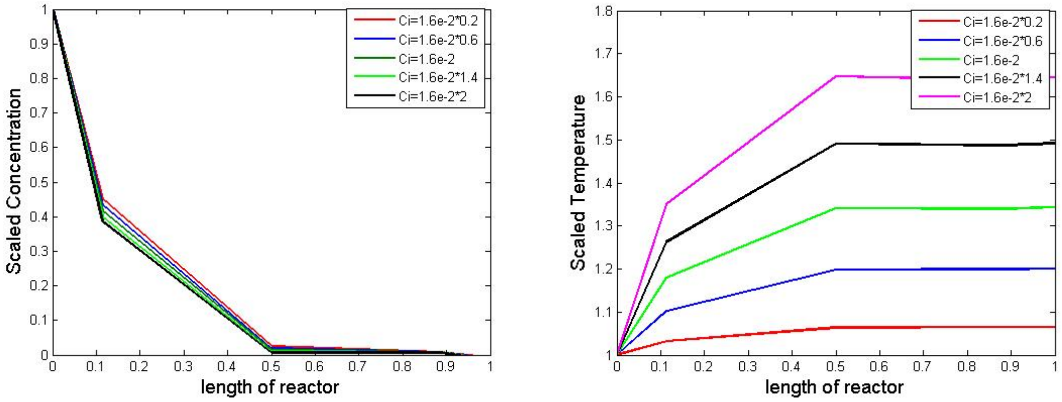

Fig. 15 Steady state reactor concentration (left) and temperature (right) sensitivity vs. C_i. (Nominal value is C_i=1.6E-2 mol/m^3)

From the figure above, we observe there is a decrease of concentration profile as we increase the inlet concentration. Such phenomenon is the result of scaling. We do observe actual decrease in concentration as we multiply the scaled concentration with its respective inlet concentration. At the same time, we observe an increase in temperature profile as we increase inlet concentration. We may attempt to explain the behavior for that the reaction is exothermic, as the concentration of reactant increases, reactions will release more heat.

The reactor profile sensitivity against inlet temperature is shown in Fig. 16.

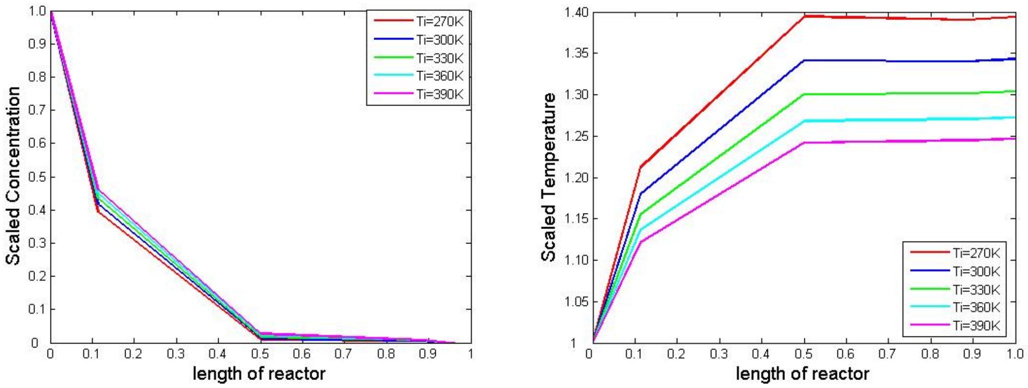

Fig. 16 Steady state reactor concentration (left) and temperature (right) sensitivity vs. T_i. (Nominal value T_i=550K)

From Fig. 16, we observe an increase in reactor concentration profile as we increase the inlet temperature. For an exothermic reaction, reaction rate declines as temperature increases, thus causing the reactant concentration to increase inside the reactor. We also observe a decrease in reactor temperature profile as we increase the inlet temperature, again caused by scaling.

4.4 Reactor Design at Dynamic State

With steady state reactor profiles successfully solved, dynamic profiles can be obtained using MATLAB solvers. We notice that the convergence of solution is especially sensitivity to the choices of initial conditions (T_o and C_o), therefore the choices of initial conditions are not completely at random. We programmed the steady state results to be implanted into the dynamic model as initial guess. With this mechanism, convergence is much better guaranteed. By changing the inlet conditions (T_i, C_i), we observe how reactor responds to the fluctuation in Fig. 17. The results are generated with steady state initial guess at T_i=300K, C_i=1.6E-2 mol/m^3, and new inlet conditions at T_i=300K, C_i=5E-2 mol/m^3.

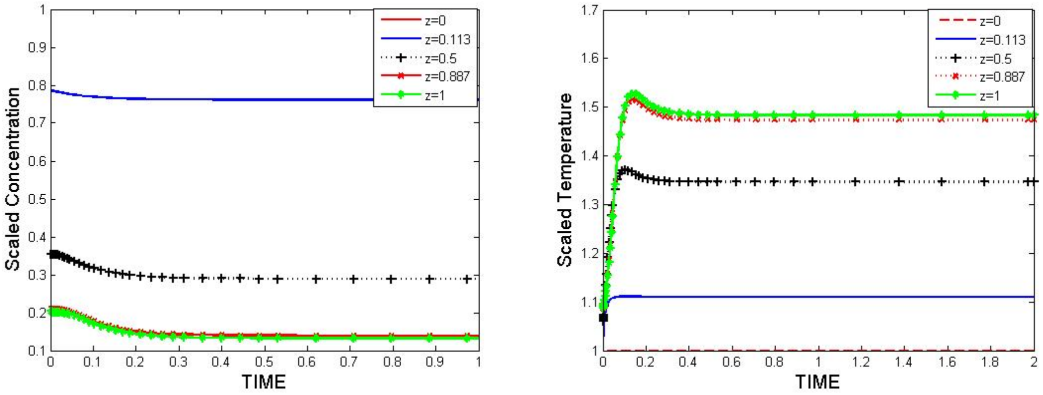

Fig. 17 Dynamic concentration profile (left) and temperature profile (right) for coupled reactor design.

We observe an increase in temperature profile and decrease in concentration profile for given collocation points along the reactor as we increase the inlet concentration. The behavior can be predicted by studying the sensitivity of steady state reactor with respect to inlet conditions. Indeed, the increase in temperature and decrease in concentration profile was already predicted by Fig. 15.

4.5 Multiplicity

In studying the relationship between η and Thiele modulus Φ defined in Eq.(11), multiple solutions of η of steady state pellets have been observed in highly exothermic or endothermic reactions. Shown in Fig. 18, for a γ value of 30, multiple solutions exist for a single Thiele modulus at β=0.4. The x-axis shows normalized Thiele modulus. Multiple solutions indicate multiple steady states may exist depending on the initial condition of the pellet. Further study on the multiplicity behavior and pellet initial condition shows that for a single Φ, multiple concentration profiles inside the pellet could be obtained. In Fig. 19, three different concentration profiles are observed for the three values of η marked on Fig. 18.

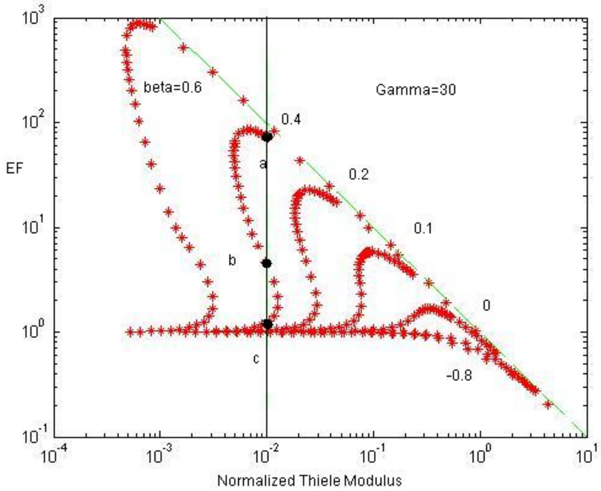

Fig. 18 Effectiveness factor multiplicity behavior

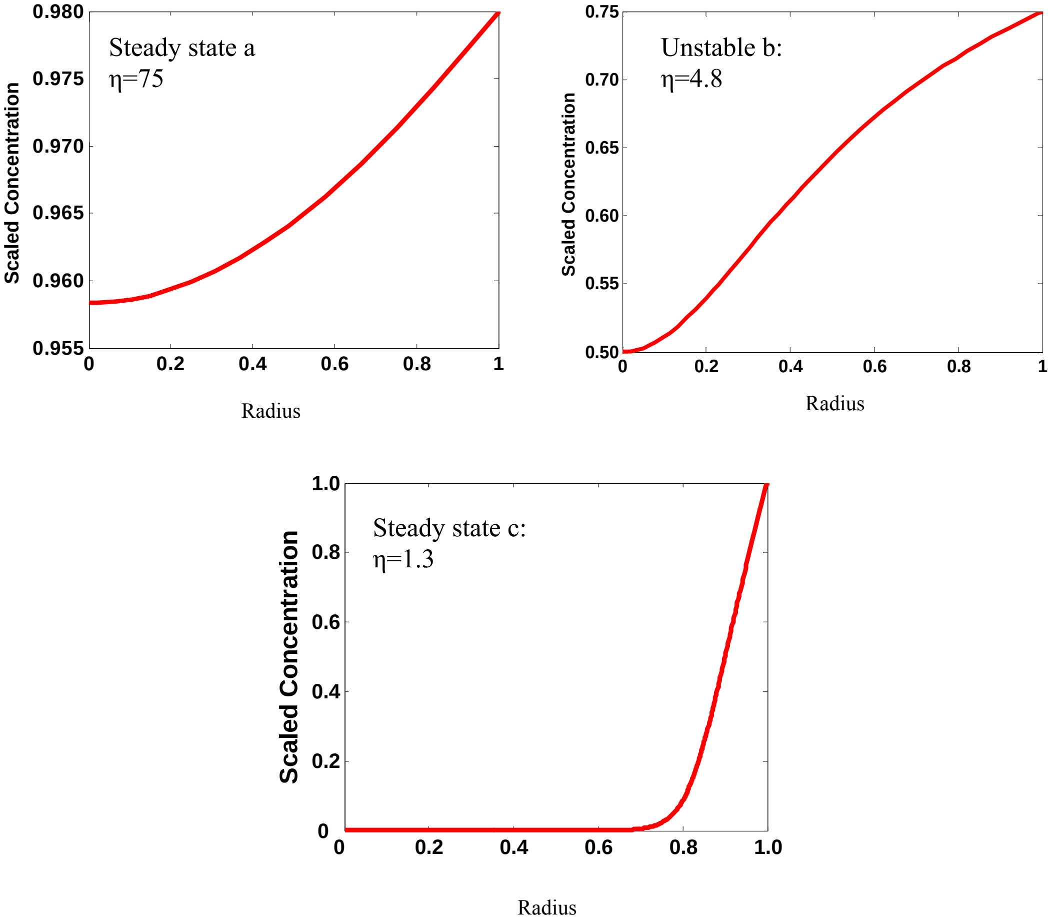

Fig. 19 Different concentration profiles correspond to the same Thiele modulus (Φ=0.430)

There are two distinct steady states for this particular pellet: steady state *a* and *c*. The transition point *b* indicates an unstable region where slight change in reaction condition may disturb the system, causing the system to slide into either steady state *a* or *c*.

Mathematically, multiplicity occurs due to the nonlinearity and coupling of differential equations. However, the discovery of multiple steady states is crucial in terms of the design safety. If the design is scaled at an initial condition different from the realistic initial condition, the pellets may choose to approach a different steady state than predicted.

Furthermore, we observe along a particular reactor with conditions C_i=1.6E-6 mol/m^3 and T_i=550K, pellets also exhibit multiplicity behavior based on the changing bulk conditions along the reactor axial direction. At the beginning section of the reactor, pellets may converge to three possible steady states. At this section of the reactor, uncertainty in parameters may disturb the value of Thiele Modulus, consequently causing the reactor running at one steady state to switch to another (refer to Fig. 20 left figure). As the reaction condition changes, pellets fall into one steady state region in the later section of the reactor. The results are shown in Fig. 20 and Fig. 21.

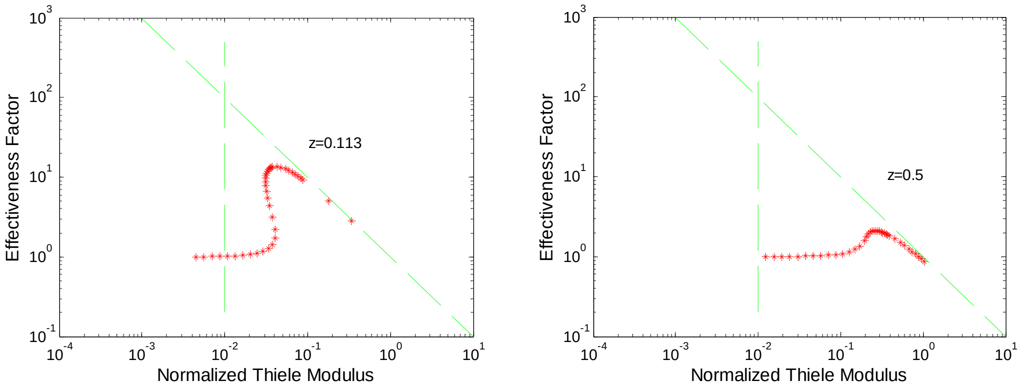

Fig. 20 Multiplicity in reactor, *z*=0.113 (left) and *z*=0.5 (right)

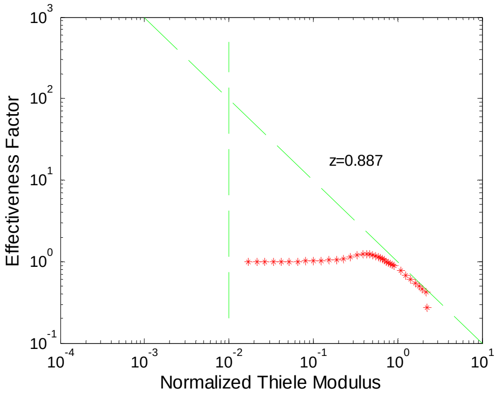

Fig. 21 Multiplicity in reactor, *z*=0.887

We also wish to study how uncertainty in different parameters may cause a dynamic reactor operating at different steady states to behave differently. This study is combined with the study of reactor "hotspot", which is shown in the next section.

4.6 Reactor Hotspot

"Hotspot" for fixed bed reactors can be observed for strongly exothermic processes. By varying the heat of reaction, length of reactor and the amount of heat removed by cooling, we observe a hotspot for the designed reactor at steady state. Shown in Fig. 22, "hotspot" for the reactor approximately happens at *z*=0.12 of the scaled length.

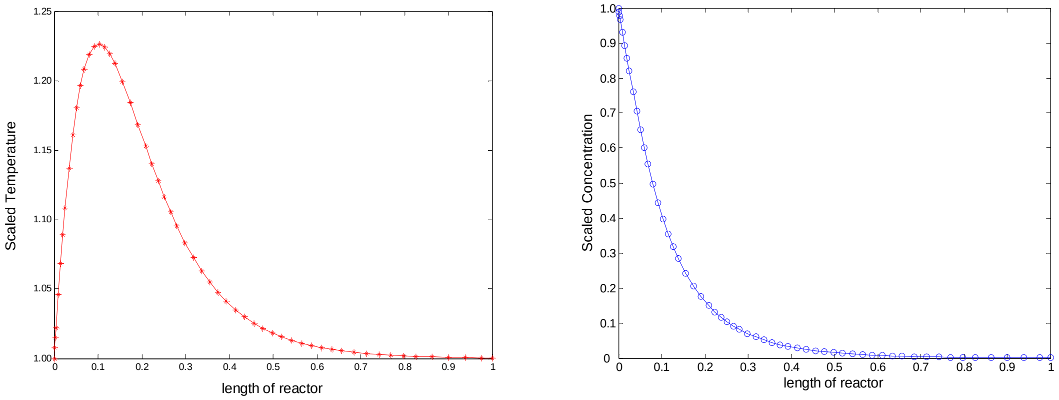

Fig. 22 "Hotspot" temperature (left) and concentration (right) profile. Figure produced at T_i=300K, C_i=1E-2 mol/m^3

"Hotspot" phenomenon can be observed when the concentration of reactant decreases along the axial direction of the reactor, the decreasing heat of reaction is compromised by heat removed by cooling, and thus creating a maximum in temperature profile along the reactor. The sensitivity of the "hotspot" varying with inlet conditions and reaction rate is shown in Fig. 23 and Fig. 24.

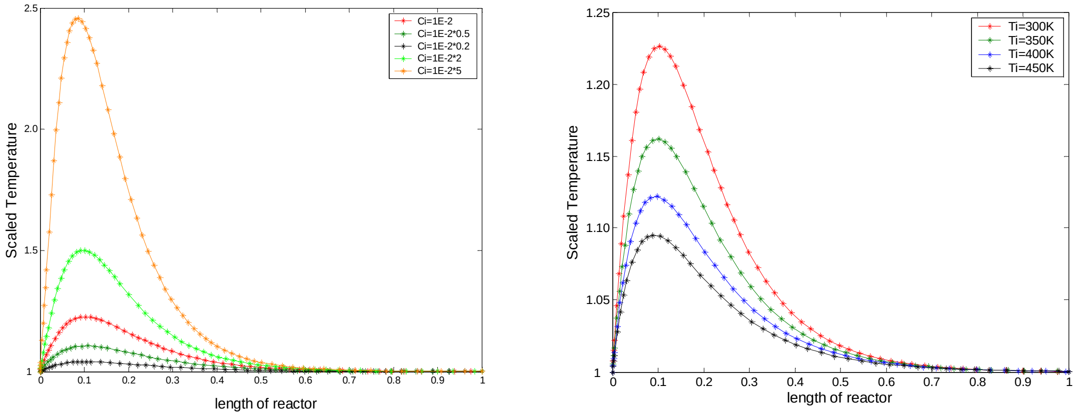

Fig. 23 "Hotspot" Sensitivity vs. inlet concentration (left) and inlet temperature (right). Nominal values are T_i=300K, C_i=1E-2 mol/m^3.

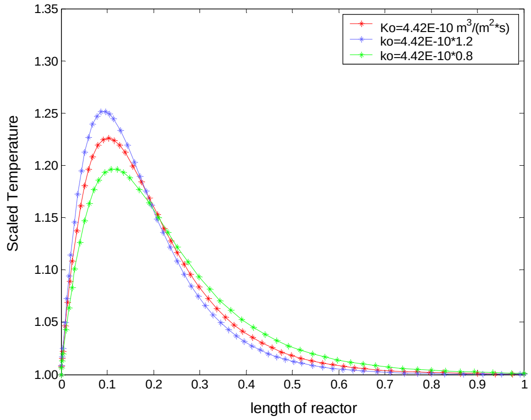

Fig. 24 “Hotspot” sensitivity vs. reference reaction rate (k_o). Nominal value: k_o=4.42E-10 $m^3/(m^2*s)$.

As shown in Fig. 23, the location of the “hotspot” is stationary when inlet temperature and concentration changes. On the other hand, Fig. 24 shows that the magnitude and the location both change as we vary reference reaction rate. Fix inlet temperature, the magnitude of “hotspot” increase dramatically as inlet concentration increases. While fixing inlet concentration, the magnitude of “hotspot” decreases as inlet temperature decreases. In the design of this reactor, we should avoid fluctuating inlet concentration as the reactor temperature seems to be much more sensitivity toward the inlet concentration. As shown in Fig. 21, if we have C_i=0.05 mol/m^3 instead of C_i=0.01 mol/m^3, the highest temperature inside the reactor increases to approximately 750K (instead of 360K at nominal inlet concentration). If we prepare our reactor at the nominal value of C_i=1E-2mol/m^3 without the information of uncertainty, the reactor safety is certainty jeopardized.

We wish to combine the knowledge of pellet multiplicity and reactor hotspot to discover if disturbance of parameter will cause reactor with pellets operating at different steady states to behave differently. From Fig. 18, we observe there are two possible steady state *a* and *c* in pellets, with unstable region *b* as transition. At these steady states, we adjusted our reactor to run dynamic models. We chose the temperature profile at *z*=0.5 and initial steady state at T_r=1.2, C_r=0.7, then we implemented a new inlet temperature T_i=300K, C_i=1E-2mol/m^3. This allows us to see how the reactor at *z*=0.5 approach distinct steady states with respect to time as shown in Fig. 25.

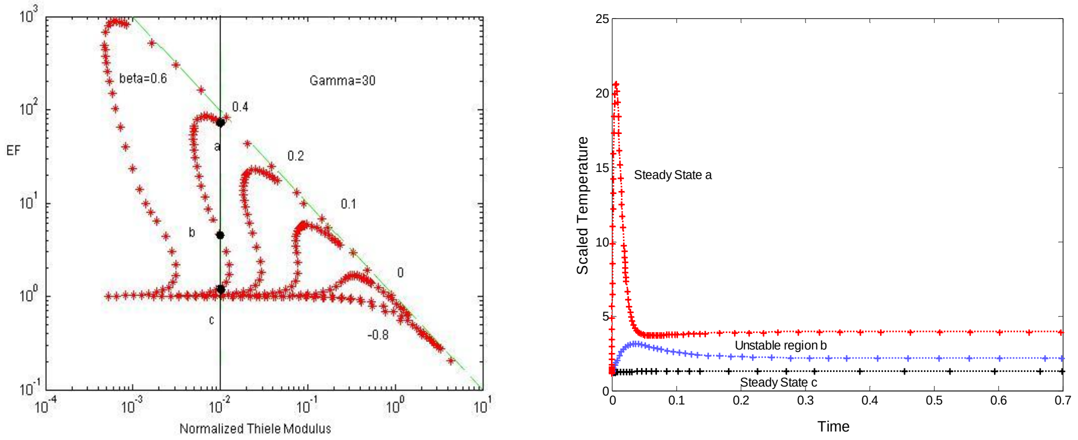

Fig. 25 Dynamic reactor running at multiple steady states (right). z=0.5

If the reactor is operating at the pellet steady state *c*, then we will not observe an obvious "hotspot" for the disturbance we implemented. However, if the reactor is running at pellet steady state *a*, such a disturbance will cause the reactor to have a "hotspot" with a temperature twenty times higher than the inlet temperature. To design a safe reactor, we certainly need to conduct uncertainty study in order to fully understand how the reactor behaves against possible disturbance in various parameters. To prevent such disastrous "hotspot" in this specific reactor, we will have to pay special attention to the initial condition of the pellets to force them to approach steady state c.

5. Conclusion

Designing a catalytic pellet reactor requires engineers to accurately predict the mass transfer and heat transfer phenomenon occurring both inside an individual pellet and within the entire reactor. To guarantee the safety condition of the process, it is also important to study the degrees of uncertainties in the design parameters.

Collocation method has proven to accurately solve systems of non-linear second order differential equations. Using this method, we are able to numerically solve for the concentration and temperature profile for an individual pellet under steady and dynamic state. With the information for individual pellets known, effectiveness factor η can then be evaluated. The sensitivity of effectiveness factor against steady state surface conditions and pellet physical properties are studied. Behaviors of effectiveness factor against these parameters satisfy the physical picture we predicted.

We then proceed to study the concentration and temperature profiles of the entire reactor. Solving pellet and reactor governing equations simultaneously, numerical solutions of reactor profiles at steady state and dynamic state are obtained. Reactor steady state sensitivity is done targeting inlet conditions as uncertain parameters.

As we study the steady state pellets, we observe for one reaction condition, there exist multiple effectiveness factor. Such phenomenon indicates multiple steady states inside the pellet may exist depending on the initial conditions of the pellets. We are able to verify the existence of pellet multiple steady states by observing three distinct concentration profiles inside the pellet. As we observe multiple steady states in pellets, we continue to study how the reactor responds to such a phenomenon. This study is combined with the reactor behavior “hotspot” to give a sounding prove that uncertainty study strongly influences design safety. “Hotspots” may happen depending on the physical parameters of the reactor. In testing the location and magnitude of “hotspot”, we observe that the case study reactor is especially sensitive to reactor inlet concentration. Implementing multiplicity behavior, a reactor dynamic model is developed to run with the different steady states found in pellets and the resulting magnitude of “hotspot” is analyzed. Specific example shows that a reactor normally operates at one steady state may show no obvious “hotspot” as we disturb inlet conditions, while the same disturbance may cause intolerable “hotspot” appearing in the reactor running at a different steady state. This is certainty important to warn engineers designing reactors: uncertainty study should be done to prevent process failure and unsafe design for every design.

Future work on this design may include studying deeper into reactor multiplicity along axial direction. It may be convincing to show a case where pellets operate at different steady states as concentration and temperature change inside the reactor. A dynamic model will help to show if there is transition of steady states inside the reactor with respect to time.

7. Nomenclature

a	Dimensionless reaction rate coefficient
Bi_M	Bios number for mass transfer
C_A	Pellet concentration of reactant A
C_{As}/C_s, T_s	Pellet surface conditions
C_i, T_i	Reactor inlet conditions
C_o, T_o	Initial conditions
C_p	Heat capacity of gas reactant
C_r	Scaled reactor bulk concentration
D_{AB}	Mass diffusion coefficient
E_a	Arrhenius activation energy
k_b	Heat conductivity coefficient of the bed
k_e	Heat conductivity coefficient of gas reactant
r	Radial position along the pellet
R	Pellet radius
r'_A	Overall rate of reaction
S_a	Pellet internal surface area
T	Pellet temperature
T_r	Scaled reactor bulk temperature
U	Superficial velocity
β	Dimensionless variable in pellet equation
γ	Dimensionless variable in pellet equation
ΔH	Heat of reaction
ζ	Scaled pellet temperature
η	Internal effectiveness factor
θ	Scaled radius
ρ_b	Density of the bed

ρ_g	Density of reactant gas
φ	Scaled pellet concentration
Φ	Thiele modulus

8. References

[1] Villadsen John and Michelsen Michael L. *Solutions of Differential Equations Models by Polynomial Approximation*. Prentice-Hall, Inc., New Jersey, 1978.

[2] Fogler H. Scott. *Elements of Chemical Reaction Engineering. 3^{rd} Ed*. Prentice-Hall Inc., New Jersey, 1999

[3] Froment Gilbert F. and Bischoff Kenneth B. Chemical Reactor Analysis and Design, 2^{nd} Ed. John Wiley & Sons, Inc. Canada, 1979.

[4] Weisz P.B and Hicks J.S. *The behavior of porous catalyst particles in view of internal mass and heat diffusion effects.* Chemical Engineering Science, Vol. 17, pp. 265. 1962.

[5] Damkohler G., D.Chem.Ing. Vol. 3, pp.430. 1937.

[6] Thiele E. W., Industr. Engng. Chem. Vol. 31, pp.916. 1939.

[7] Zeldowitsch J.B., Acta Physicochim, U.R.S.S. Vol. 10, pp.583. 1939.

[8] Wheeler A. in Advances in Catalysis. Vol.3, pp. 249. Academic Press, New York 1951.

[9] Pita J., Balakotaiah V, and Luss Dan. *Thermoflow Multiplicity in a Packed-Bed Reactor: Condition and Cooling Effects.* AIChE Journal. Vol. 35, Issue 3, pp. 373-384. March 1989.

[10] Syed Faisal H., Datta Ravindra and Jensen Kyle L., *Thermodynamically Consistent Modeling of a Liquid-Phase Nonisothermal Packed-Bed Reactor.* AIChE Journal. Vol. 46, pp.380. Feb. 2000.

[11] Lee, J. P., Balakotaiah V., Luss Dan. *Thermoflow multiplicity in a packed-bed reactor Part I: Adiabatic case. AIChE Journal.* Vol.33, Issue 7, pp.1136-1154. July 1987.

9. Appendix

6.1 Physical Properties of the Pellet

$\varepsilon=0.4$ (porosity)
$D_e=5.2E\text{-}2\ cm^2/s$
$K_e=7E\text{-}4$ cal/s cm ^oK
$R=0.3$ cm
$\rho_c=2.8E6\ g/m^3$
$S_a=530\ m^2/g$

6.2 Physical Properties of the Bed

$\varepsilon_b=0.4$
$\rho_b=\rho_c(1-\varepsilon_b)$

6.3 Physical Properties of the Inlet Gas

To=773 ^{o}K
U=100cm/s
C_{SO2}=C_{bo}=1.6E-6 mol/cm^3
D_{AB}=2.0E-8 m^2/s

6.4 Reaction Conditions

ΔH= 23100 cal/mol
k_o=4.42E-10 m/s

6.5 Analytical Method Solving Isothermal Pellet

```
function PelletDiffusion_AnalyticalSolution

% Case study of 1.3.5 P33, Determine Thiele Number (sita in this case)
De=5.2*10^(-2);  % cm^2/s
R=0.3;   % cm
To=500;  % Kelvin
Rg=8.314;    % J/mol*K
E= 2000;     % J/mol, approximated
a=0.5;      % 1/s, approximated
Sita=R*(1/De*(a*exp(-E/(Rg*To))))^0.5;

% roots=[0,0.3,0.5,0.7,1];
roots=0:0.1:1;
C=besseli(0,Sita*roots)/besseli(0,Sita);
C
plot(roots,C,'-r','LineWidth',2);

%%%%%Effectiveness factor
ef=2/Sita*besseli(1,Sita)/besseli(0,Sita);
display(ef);
```

6.6 Collocation Method Solving Non-Isothermal Pellet at steady state

Main Program to Solve for Concentration/Temperature Profile and η

```
function SolveDiffusion
%%% Collocation Method to solve the diffusion equations.
clear;
clc;
%%% Choose the  arbitrary and orthgoanal nodes
%%% Compare the accuarrcy
ReactorLength=1;    % m
roots=[0,0.113,0.5,0.887,1].*ReactorLength;
mark='--g.';
m=length(roots);

%Initial Guess:
x0=[0.8785 0.9213 0.9370 0.9553 1.0000 7.35 6.37 5.60 3.70 1.0000];
```

```
%Definining Constants
Co=1E-6;
To=773;
De=5.2*10^(-2); % cm^2/s
E= 5000;     % J/mol
DeltaH=-96558;               %J/mol
Ke=2.926e-5;                 %W/m*K, heat conductivity
beta=DeltaH*De*Co/(Ke*To);
Rg=8.314;    % J/mol*K, gas constant
gamma=E/(Rg*To);
R=0.3;   % cm

%Start Solving 2nd order differential equations with Residuals
Minimized

%options = optimset('Display','iter','Maxiter',150,'MaxFunEvals',1500);
options = optimset('Display','iter','NonlEqnAlgorithm','gn');
x=fsolve(@diffusionfunction,x0,options,roots,Reaction_rate,To,Co)
C=x(1:m);
T=x(m+1:2*m);

%Generate Plot for Concentration and Temperature Profile
hold on
plot(roots,C,'-b','LineWidth',2);
xlabel('radius');
ylabel('concentration ratio');
legend ('Concentration');
plot(roots,T,'-r','linewidth',1);
xlabel('radius');
ylabel('Temperature Ratio');
legend('Temperature');
hold off;

%Integrate to get effectiveness factor
tol=1e-6;
trace=[];
EffectivenessFactor=quadl(@ReactionRateFunction,0,1,tol,trace,C,T,roots
,gamma);
EffectivenessFactor

%Define Reactions used in script above
function f=ReactionRateFunction(t,C,T,roots,gamma)
m=length(t);
for i=1:m
    a_Con=PolynomialFunction(t(i),C,roots);
    a_Temp=PolynomialFunction(t(i),T,roots);
    f(i)=3*t(i)^2*a_Con*exp(gamma*(1-1/a_Temp));
end;


function f=PolynomialFunction(t,weight,roots)
sum=0;
order=length(roots);
for i=1:order
    sum=sum+weight(i)*getLagrangian(i,order,t,roots);
```

```
end;
f=sum;
```

Define Steady State Models used in the Main Program

```
function f=Diffusionfunction(x,root,rate,To,Co)
m=length(root);
C=x(1:m);
T=x(m+1:2*m);

% Case study of 1.3.5 P33, Determine Thiele Number
De=5.2*10^(-2); % cm^2/s
R=0.3;   % cm
To=773;  % Kelvin
Rg=8.314;    % J/mol*K, gas constant
E= 5000;     % J/mol
a=0.5;      % 1/s
Thiele=R*(1/De*(a*exp(-E/(Rg*To))))^0.5;

% Case study of 1.3.5 P33, Determine beta and gamma value for
temperature profile
DeltaH=-96558;               %J/mol
Co=1*10^(-6);   %Reference Concentration
Ke=2.926e-5;                %W/m*K, heat conductivity
beta=DeltaH*De*Co/(Ke*To);
gamma=E/(Rg*To);

f(1)=GetDiffusionFirstOrderDerivative(C,1,m,root); %% starting boundary
dC/dx=0
f(m)=C(m)-1;
for i=2:m-1
    f(i)=GetDiffusionSecondOrderDerivative(C,i,m,root)+...
        2.0/root(i)*GetDiffusionFirstOrderDerivative(C,i,m,root)-
Thiele^2*exp(gamma*(1-1.0/T(i)))*C(i); %% interior points dC2/dx2=0
end;

f(1+m)=GetDiffusionFirstOrderDerivative(T,1,m,root); %% starting
boundary dC/dx=0
f(2*m)=T(m)-1;         %%%%% ending boundary C=1;
for i=2:m-1;
    f(i+m)=GetDiffusionSecondOrderDerivative(T,i,m,root)+...
        2.0/root(i)*GetDiffusionFirstOrderDerivative(T,i,m,root)-
beta*exp(gamma*(1-1.0/T(i)))*C(i); %% interior points dC2/dx2=0
end;

%%%%%%%%%%%%%%%%%%%%%Frist order derivative for state variables y
(dy/dx)%%%%%%%%%%%%%%%%%%
function
f=GetDiffusionFirstOrderDerivative(weights,aNodeIndex,order,nodes)
sum = 0;
for i= 1 : order
    d = getLagrangianDerivative(aNodeIndex,i,order,nodes);
    sum = sum + d * weights(i);
end;
f = sum;
```

```
%%%%%%%%%%%%%%%%%%%%%Second order derivative for state variables y
(dy2/dx2)%%%%%%%%%%%%%%%%%%%
function
f=GetDiffusionSecondOrderDerivative(weights,aNodeIndex,order,nodes)
sum = 0;
for i= 1 : order
    d = getLagrangianSecondOrderDerivative(aNodeIndex,i,order,nodes);
    sum = sum + d * weights(i);
end;
f = sum;

%%%%%%%%%%%%%%%%%%%%%Second order derivative for laganrage polynomial
L(x)%%%%%%%%%%%%%%%%%%%

function
la=getLagrangianSecondOrderDerivative(aNodeIndex,anOrderIndex,order,nod
es)
 if (aNodeIndex == anOrderIndex)
    xx=nodes(aNodeIndex);

la=1.0/3.0*getThirdOrderDerivNodesPol(xx,order,nodes)/getFirstOrderDeri
vNodesPol(xx,order,nodes);
    return;
  end;
  xj=nodes(aNodeIndex);
  xi=nodes(anOrderIndex);
  la = 1.0/(xj-
xi)*(getSecondOrderDerivNodesPol(xj,order,nodes)/getFirstOrderDerivNode
sPol(xi,order,nodes)...
  -2.0*getLagrangianDerivative(aNodeIndex,anOrderIndex,order,nodes));

%%%%%%%%%%%%%%%%%%%%%Frist order derivative for laganrage polynomial
L(x)%%%%%%%%%%%%%%%%%%%

function
la=getLagrangianDerivative(aNodeIndex,anOrderIndex,order,nodes)
 if (aNodeIndex == anOrderIndex)
    xx=nodes(aNodeIndex);

la=1.0/2.0*getSecondOrderDerivNodesPol(xx,order,nodes)/getFirstOrderDer
ivNodesPol(xx,order,nodes);
    return;
  end;
  xj=nodes(aNodeIndex);
  xi=nodes(anOrderIndex);
  la = 1.0/(xj-
xi)*getFirstOrderDerivNodesPol(xj,order,nodes)/getFirstOrderDerivNodesP
ol(xi,order,nodes);

 %%%%%%%%%% Polynomials %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f=nodePolynomials(x,order,nodes)
  enumerator  = 1.0;
  if (order == 0)
      f=1;
      return;
  end;
```

```
%     for i= 1 : order
%       enumerator  = enumerator * ( x- nodes(i) );
%     end;
    enumerator = (x-nodes(order))*nodePolynomials(x,order-1,nodes);
    f = enumerator;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%for the derivative of nodes polynomials
%
function g=getFirstOrderDerivNodesPol(x,order,nodes)
  if (order == 0)
      g=0;
      return;
  end;
      g=(x-nodes(order))*getFirstOrderDerivNodesPol(x,order-1,nodes) +
nodePolynomials(x,order-1,nodes);
%   g=(x-nodes(order))+nodePolynomials(x,order-1,nodes);

function h=getSecondOrderDerivNodesPol(x,order,nodes)
  if (order == 0)
      h=0;
      return;
  end;
  h=(x-nodes(order))*getSecondOrderDerivNodesPol(x,order-
1,nodes)+2.0*getFirstOrderDerivNodesPol(x,order-1,nodes);

function h=getThirdOrderDerivNodesPol(x,order,nodes)
  if (order == 0)
      h=0;
      return;
  end;
  h=(x-nodes(order))*getThirdOrderDerivNodesPol(x,order-
1,nodes)+3.0*getSecondOrderDerivNodesPol(x,order-1,nodes);
```

Define the LaGrange Polynomial and its Derivatives

```
function La=getLagrangian(aNodeIndex,order,x,root)

  enumerator  = 1.0;
  denominator = 1.0;
  for i= 1 : order

     if (i ~= aNodeIndex)
       enumerator  = enumerator * ( x- root(i) );
       denominator = denominator * ( root(aNodeIndex) - root(i) );
     end;
  end;
La = enumerator/denominator;

function la=getLagrangianDerivative(aNodeIndex,anOrderIndex,order,nodes)
 if (aNodeIndex == anOrderIndex)
   xx=nodes(aNodeIndex);
```

```
la=1.0/2.0*getSecondOrderDerivNodesPol(xx,order,nodes)/getFirstOrderDerivNodesPol
(xx,order,nodes);
    return;
  end;
  xj=nodes(aNodeIndex);
  xi=nodes(anOrderIndex);
  la = 1.0/(xj-
xi)*getFirstOrderDerivNodesPol(xj,order,nodes)/getFirstOrderDerivNodesPol(xi,order,n
odes);


function f=nodePolynomials(x,order,nodes)
  enumerator  = 1.0;
  if (order == 0)
      f=1;
      return;
  end;
%     for i= 1 : order
%       enumerator  = enumerator * ( x- nodes(i) );
%     end;
    enumerator = (x-nodes(order))*nodePolynomials(x,order-1,nodes);
    f = enumerator;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%
%
%for the derivative of nodes polynomials
%
function g=getFirstOrderDerivNodesPol(x,order,nodes)
  if (order == 0)
      g=0;
      return;
  end;
      g=(x-nodes(order))*getFirstOrderDerivNodesPol(x,order-1,nodes) +
nodePolynomials(x,order-1,nodes);
%   g=(x-nodes(order))+nodePolynomials(x,order-1,nodes);

function h=getSecondOrderDerivNodesPol(x,order,nodes)
  if (order == 0)
      h=0;
      return;
  end;
  h=(x-nodes(order))*getSecondOrderDerivNodesPol(x,order-
1,nodes)+2.0*getFirstOrderDerivNodesPol(x,order-1,nodes);
```

```
function
la=getLagrangianSecondOrderDerivative(aNodeIndex,anOrderIndex,order,nodes)
 if (aNodeIndex == anOrderIndex)
    xx=nodes(aNodeIndex);

la=1.0/3.0*getThirdOrderDerivNodesPol(xx,order,nodes)/getFirstOrderDerivNodesPol(x
x,order,nodes);
    return;
  end;
  xj=nodes(aNodeIndex);
  xi=nodes(anOrderIndex);
  la = 1.0/(xj-
xi)*(getSecondOrderDerivNodesPol(xj,order,nodes)/getFirstOrderDerivNodesPol(xi,orde
r,nodes)...
  -2.0*getLagrangianDerivative(aNodeIndex,anOrderIndex,order,nodes));


function la=getLagrangianDerivative(aNodeIndex,anOrderIndex,order,nodes)
 if (aNodeIndex == anOrderIndex)
    xx=nodes(aNodeIndex);

la=1.0/2.0*getSecondOrderDerivNodesPol(xx,order,nodes)/getFirstOrderDerivNodesPol
(xx,order,nodes);
    return;
  end;
  xj=nodes(aNodeIndex);
  xi=nodes(anOrderIndex);
  la = 1.0/(xj-
xi)*getFirstOrderDerivNodesPol(xj,order,nodes)/getFirstOrderDerivNodesPol(xi,order,n
odes);

function f=nodePolynomials(x,order,nodes)
  enumerator  = 1.0;
  if (order == 0)
      f=1;
      return;
  end;
%     for i= 1 : order
%       enumerator  = enumerator * ( x- nodes(i) );
%     end;
    enumerator = (x-nodes(order))*nodePolynomials(x,order-1,nodes);
    f = enumerator;
```

6.7 Collocation Method Solving Non-Isothermal Pellet at dynamic state

Main Program to Solve Concentration/Temperature Profile

```
function SolveDiffusion
%%% Collocation Method to solve the diffusion equations.
clear;
clc;
%%% Choose the  arbitrary and orthgoanal nodes
%%% Compare the accuarrcy
%roots=[0,0.3,0.5,0.7,1];
% roots=[0,0.113,0.5,0.887,1];
To=550; %This value corresponds to the value in "SolveDiffusion".
Co=2.46e-4;% This value corresponds to the value in "SolveDiffusion".
Reaction_rate=1e-5;
ReactorLength=1;    % m
roots=[0,0.113,0.5,0.887,1].*ReactorLength;
m=length(roots);
x0=[0 0 0 0 1 1 1 1 1 1];
%options = optimset('Display','iter');
% x=fsolve(@diffusionfunction,x0,options,roots,Reaction_rate,To,Co);
% C=x(1:m);
% T=x(m+1:2*m);
% plot(roots,C,'-rs');
% Plot(roots,T,'--g');
% x0(1:2*m)=1;
% x0(m)=1;
% x0(2*m)=1;
options = odeset('RelTol',1e-4);
[t,x]=ode45(@DynamicDiffusionfunction,[0,1],x0,options,roots,Reaction_rate,To,Co);
C=x(:,1:m)
T=x(:,m+1:2*m)
t
FinalTime=length(t)
%plot(roots,x(FinalTime,:));
surf(C,roots,t);
surf(T,roots,t);
```

Define Dynamic Model used in the Main Program

```
function dc=DynamicDiffusionfunction(t,x,root,rate,To,Co)
f=Diffusionfunction(x,root,rate,To,Co);
dc=f';
```

6.8 Testing Sensitivity of steady state η

```
% function SolveDiffusion(roots,mark)
function SolveDiffusion
%%% 04/18/2005, Libin Zhang
%%% Collocation Method to solve the diffusion equations.
```

```
clear;
clc;

%%% Choose the  arbitrary and orthgoanal nodes
%%% Compare the accuarrcy
% roots=[0,0.3,0.5,0.7,1];
ReactorLength=1;    % m
roots=[0,0.113,0.5,0.887,1].*ReactorLength;
mark='--g.';
Reaction_rate=1e-5;     %mol/(m^3*s)
m=length(roots);

%Initial Guess, sensitive to convergence
x0=[0.8756    0.8772    0.9066    0.9736    1.0000   33.9194   33.5085   25.7212    7.9898
1.0000];

%Start for loop for varying Co and To
% k=0;
% for Co=1E-6:5E-6:300E-6
%     k=k+1;
%     z=0;
%     for To=100:50:1000
%         z=z+1;

% Start for loop for varying Reaction_rate
% k=0;
% for a=0.1:0.1:1
%     k=k+1;

% Start for loop for varying Ke
% k=0;
% for Ke=0.2926e-5:1e-5:12e-5  % W/m*K
%     k=k+1;

% Start for loop for varying De
% k=0;
% for De=1e-2:1e-2:12e-2
%     k=k+1;

%Definining Constants
Co=2.46E-4;
To=550; %Kelvin
De=5.2*10^(-2); % cm^2/s
E=5000;     % J/mol
DeltaH=-96558;              %J/mol
Ke=2.926e-5;                %W/m*K, heat conductivity
```

```
beta=DeltaH*De*Co/(Ke*To);
Rg=8.314;    % J/mol*K, gas constant
gamma=E/(Rg*To);
R=0.3;   % cm
a=0.5;

%Start Solving 2nd order differential equations with Residuals Minimized

%options = optimset('Display','iter','Maxiter',150,'MaxFunEvals',1500);
options = optimset('Display','off','NonlEqnAlgorithm','gn');
x=fsolve(@diffusionfunction,x0,options,roots,Reaction_rate,To,Co,a,Ke,De,beta,gamma)
;
C=x(1:m);
T=x(m+1:2*m);

%
% r=0:0.1:1;
% for i=1:length(r)
%     y(i)=PolynomialFunction(r(i),C,roots);
% end;
% Generate Plot for Concentration and Temperature Profile
%hold on
%plot(r,y,'-b','LineWidth',2);
%plot(roots,C,mark,'LineWidth',2);
%plot(roots,T,'-r','linewidth',1);
%hold off;

%Integrate to get effectivess factor
tol=1e-6;
trace=[];
Factor=quadl(@ReactionRateFunction,0,1,tol,trace,C,T,roots,gamma);
Factor

%Store Values in different arrays
% Array_To(k,z)=To
% Array_Co(k,z)=Co
% Array_a(1,k)=a;
% Array_Ke(1,k)=Ke;
% Array_De(1,k)=De;
% Array_Factor(k,z)=Factor
end;
end;

% save Co_data.dat Array_Co -ascii;
% save To_data.dat Array_To -ascii;
% save Factor_data.dat Array_Factor -ascii;
```

```
%Plot 3D surface, Co, To are varaibles, Factor is the result
%surf(Array_beta,Array_gamma,Array_Factor);

%Plot 2D plot, a/Ke/De as variable, factor is the result
% plot(Array_Ke,Array_Factor);
% XLABEL('Ke');
% YLABEL('Factor');
% legend ('Factor');
```

6.9 Generate Multiplicity Graph and Concentrations for each η

Main Program

```
function Caculate_EffetiveFactor
clc;
clear;
roots=[0,0.113,0.5,0.887,1];
% roots=[0,0.4,0.8,1];
Beta=[0.6,0.4,0.3,0.2,0.1,0,-0.8];
%Beta=[0.6];
Gamma=30;
InitialCon_1=logspace(-11,-3,10);
InitialCon_2=logspace(-3,log10(0.5),15);
InitialCon_3=1-logspace(log10(0.5),-3,15);
 InitialCon=[1e-200,1e-100,1e-50,1e-25,InitialCon_1,InitialCon_2,InitialCon_3];
% InitialCon=linspace(0.8,0.87,20);
% InitialCon=[0.0000000000160];
m=length(Beta);
n=length(InitialCon);
T_span=[0,10];
tol=1e-10;
trace=[];
p1=1;

options = odeset('AbsTol',1E-30,'RelTol',1e-9,'Event',@StopCondition);
for i=1:m
   for j=1:n
      I_Factor=quadl(@ReactionFunction,0,1,tol,trace,Gamma,Beta(i));
      I_Factor=sqrt(2*I_Factor);
      abs_tol=(1e-5)*InitialCon(j);
      options = odeset('AbsTol',abs_tol,'RelTol',1e-9,'Event',@StopCondition);
      xx0=[InitialCon(j),0];
      [t,x]=ode15s(@TransformDiffusionfunction,T_span,xx0,options,Gamma,Beta(i));
%        if exitflag~=1
%           fprintf(1,'Waring: Does not convergence; exit flag= %3d\n',exitflag);
%        end;
```

```
        No_Step=length(t);
%           hold on
         plot(t,x(:,1));
         xlabel ('radius');
         ylabel ('Scaled Concentration');
%           plot([0, t(No_Step)],[1,1],'--r');
%           hold off
        a=t(No_Step)/3;
        ThieleM=a*sqrt(p1)
        Con_Deriv_surface=x(No_Step,2);
        factor=a*Con_Deriv_surface/(ThieleM^2);
        Thiele_M(i,j)=ThieleM/I_Factor;
        EFactor_M(i,j)= factor
    end;

end;
figure
save Thiele_Data.dat Thiele_M -ascii;
save EFactor_Data.dat EFactor_M -ascii;
load Thiele_Data.dat;
load EFactor_Data.dat;
Thiele_M=Thiele_Data;
EFactor_M=EFactor_Data;
pts=[0.001 1000 0.01 0.2;
     10    0.1  0.01 500];
loglog(Thiele_M',EFactor_M','r*');
hold on
loglog(pts(:,1)',pts(:,2)','--g');
loglog(pts(:,3)',pts(:,4)','--g');

hold off
axis([1e-4 10 0.1 1000]);
% plot(Thiele_M',EFactor_M','-r*');
% axis([0.76 0.92 0.8 5]);

function f=ReactionFunction(c,Gamma,beta)
f=c.*exp(Gamma*beta*(1-c)./(1+beta*(1-c)));
```

Define Functions used in Main Program

```
function dy=TransformDiffusionfunction(t,y,varargin)
Gamma=varargin{1};Beta=varargin{2};
C=y(1);
DCDt=y(2);
f(1)=DCDt; %% starting boundary
if (t==0)
```

```
   f(2)=1/3*C*exp(Gamma*Beta*(1-C)/(1+Beta*(1-C)));
else
   f(2)=-2/t*DCDt+C*exp(Gamma*Beta*(1-C)/(1+Beta*(1-C))); %% interior points
dC2/dx2=0
end
dy=f';

function [value,isterminal,direction]=StopCondition(t,y,varargin)
Gamma=varargin{1};Beta=varargin{2};
value(1)=1.0-y(1);
isterminal(1) = 1;
direction(1) = 0;
```

6.10 Solve Steady State Reactor Concentrations and Temperature Profile

Main Program

```
% function SolveDiffusion(roots,mark)
function SolveDiffusion
%warning off MATLAB:nearlySingularMatrix
%%% Collocation Method to solve the diffusion equations.
clear;
clc;

ReactorLength=1;    % m
roots=[0,0.113,0.5,0.887,1].*ReactorLength;
%roots=[0,0.113,0.5,0.887,1].*ReactorLength;

mark='--g.';
%Reaction_rate=1e-5;     %mol/(m^3*s)
m=length(roots);
% x0(1:2*m)=0.5

% xr0=[1  0.5   0.4   0.3   0.2   0.1     0.05   1   1.01    1.05    1.06  1.07   0.1  1.15];
% xr0=[1.0000 0.1285 0.0507 0.0204 0.0103 0.0092 1.0000 1.2617 1.3130 1.3087
1.3180 1.3900];
xr0=[1  0.5   0.4   0.3   0.05   1   1.01    1.05  0.1  1.15];
%Definining Constants
To=300;          %K
Co=1E-2;        %mol/cm^2       %K
De=5.2*10^(-6); % cm^2/s
E=5000;    % J/mol
DeltaH=-96558*100;              %J/mol
U=1E-3;    %m/s, Superficial velocity
Sa=580000*0.025;            % m^2/kg, used P764 Fogler case study
Dab=0.00655;              %m^2/s
```

```
rowg=4.57E-1;          %kg/m^3
rowb=1400;          % kg/m^3
Cp=1.0032e+003;                  % for gas, J/kg*K
ko=4.42E-10;          % Reference reaction rate
Kb=0.0384;       % W/(m*K)
%Kb=2.926e-5;
R=0.003;   % cm
Rg=8.314;    % J/mol*K, gas constant
a=1.4280;      % 1/s
Ke=2.926e-5;               %W/m*K, heat conductivity

% k=0;
% for To=270:30:400
%     k=k+1;
   beta=DeltaH*De*Co/(Ke*To);
   gamma=E/(Rg*To);
   options = optimset('Display','iter','NonlEqnAlgorithm','gn');
   xr=fsolve(@DiffusionCoupled,xr0,options,roots,To,Co,beta,gamma)
%     %Co(k,:)=Co;
%     To(k,:)=To;
%     Cr(k,:)=xr(1:m);
%     Tr(k,:)=xr(m+1:2*m);
% end;
Cr=xr(1:m);
Tr=xr(m+1:2*m);

figure
hold on
plot(roots,Cr(1,:),'-r','linewidth',2);
% plot(roots,Cr(2,:),'-b','linewidth',2);
% plot(roots,Cr(3,:),'-g','linewidth',2);
% plot(roots,Cr(4,:),'-c','linewidth',2);
% plot(roots,Cr(5,:),'-m','linewidth',2);
% legend('Ti=270K','Ti=300K','Ti=330K','Ti=360K','Ti=390K');
xlabel('length of reactor');
ylabel('Scaled Concentration');
hold off

figure
hold on
plot(roots,Tr(1,:),'--r','linewidth',2);
% plot(roots,Tr(2,:),'--b','linewidth',2);
% plot(roots,Tr(3,:),'--g','linewidth',2);
% plot(roots,Tr(4,:),'--c','linewidth',2);
% plot(roots,Tr(5,:),'--m','linewidth',2);
% legend('Ti=270K','Ti=300K','Ti=330K','Ti=360K','Ti=390K');
```

```
xlabel('length of reactor');
ylabel('Scaled Temperature');
hold off
```

Define functions used above

```
unction f=DiffusionCoupled(xr0,root,To,Co,beta,gamma)

m=length(root);
Cr=xr0(1:m);
Tr=xr0(m+1:2*m);

%Define All the Varaibles

% To=300;          %K
% Co=1E-2;        %mol/cm^2       %K
De=5.2*10^(-6); % cm^2/s
E=5000;     % J/mol
DeltaH=-96558*100;              %J/mol
U=1E-3;     %m/s, Superficial velocity
Sa=580000*0.025;            % m^2/kg, used P764 Fogler case study
Dab=0.00655;                %m^2/s
rowg=4.57E-1;          %kg/m^3
rowb=1400;          % kg/m^3
Cp=1.0032e+003;                 % for gas, J/kg*K
ko=4.42E-10;          % Reference reaction rate
Kb=0.0384;       % W/(m*K)
%Kb=2.926e-5;
R=0.003;   % cm
Rg=8.314;    % J/mol*K, gas constant
a=1.4280;      % 1/s
Ke=2.926e-5;                %W/m*K, heat conductivity
Thiele=R*(1/De*(a*exp(-E/(Rg*To))))^0.5;
Bio=1;
beta=DeltaH*De*Co/(Ke*To);
gamma=E/(Rg*To);

% Getting effectiveness factor
%xx0=[0.8756   0.8772   0.9066   0.9736   1.0000   0.85   0.87   0.95   0.97   1.0000];

%options = optimset('Display','iter','NonlEqnAlgorithm','gn');
options=optimset('Display','iter');
for i=2:m-1
   %avg_theta(i)=(Tr(i)*To+To)/2;
   avg_theta(i)=Tr(i);
   Thiele1(i)=(Thiele^2*exp(gamma*(1-1/avg_theta(i))))^0.5;
```

```
% xx=fsolve(@Diffusionfunction,xx0,options,root,a,Tr(i),Cr(i));
%xx=MatSolNewt(xx0,3,'Diffusionfunction');
% P_C=xx(1:m);
% P_T=xx(m+1:2*m);
% xx0=xx;
% tol=1e-6;
% trace=[];
% Factor(i)=quadl(@ReactionRateFunction,0,1,tol,trace,P_C,P_T,root,gamma);
Factor(i)=3*Bio/(Thiele1(i)^2)*(Thiele1(i)*coth(Thiele1(i))-
1)/(Thiele1(i)*coth(Thiele1(i))+Bio-1);
end;
Factor;

% Concentration Profile For a Reactor (Dimensionless)
f(1)=Cr(1)-1; % starting boundary, Cr(z=0)=Co
f(m)=GetDiffusionFirstOrderDerivative(Cr,m,m,root);
for i=2:m-1
   f(i)=GetDiffusionSecondOrderDerivative(Cr,i,m,root)-...
       U/Dab*GetDiffusionFirstOrderDerivative(Cr,i,m,root)-
rowb*Factor(i)*Sa/Dab*Cr(i)*ko*exp(gamma*(1-1/Tr(i))); %% interior points
dC2/dx2=0
end;

%Temperature Profile For a Reactor (Dimensionless)
f(m+1)=Tr(1)-1;        %%%%% ending boundary C=1;
f(2*m)=GetDiffusionFirstOrderDerivative(Tr,m,m,root); %% starting boundary dT/dx=0
for i=2:m-1;
   f(i+m)=GetDiffusionSecondOrderDerivative(Tr,i,m,root)-...
       rowg*Cp*U/Kb*GetDiffusionFirstOrderDerivative(Tr,i,m,root)-
rowb*DeltaH*Factor(i)*Sa/Kb*Cr(i)*Co/To*ko*exp(gamma*(1-1/Tr(i)))...
       -4*1*(Tr(i)-1)*To; %% interior points dC2/dx2=0
end;
```

6.11 Solve Dynamic Reactor Concentration and Temperature Profile

Main Program

```
function SolveDiffusion
%%% 04/18/2005, Libin Zhang
%%% Collocation Method to solve the diffusion equations.
clear;
clc;
%%% Choose the  arbitrary and orthgoanal nodes
%%% Compare the accuarrcy
%roots=[0,0.3,0.5,0.7,1];
% roots=[0,0.113,0.5,0.887,1];
```

```
To=550; %This value corresponds to the value in "SolveDiffusion".
Co=4e-4;% This value corresponds to the value in "SolveDiffusion".
Reaction_rate=1e-5;
ReactorLength=1;    % m
roots=[0,0.113,0.5,0.887,1].*ReactorLength;
m=length(roots);
x0=[0.8756    0.8772    0.9066    0.9736    1.0000   33.9194   33.5085   25.7212    7.9898
1.0000];
%options = optimset('Display','iter');
% x=fsolve(@diffusionfunction,x0,options,roots,Reaction_rate,To,Co);
% C=x(1:m);
% T=x(m+1:2*m);
% plot(roots,C,'-rs');
% Plot(roots,T,'--g');
% x0(1:2*m)=1;
% x0(m)=1;
% x0(2*m)=1;
options = odeset('RelTol',1e-4);
[t,x]=ode45(@DynamicDiffusionfunction,[0,1],x0,options,roots,Reaction_rate,To,Co);
C=x(:,1:m)
T=x(:,m+1:2*m)
t
FinalTime=length(t)

hold on;
plot(t,C(:,1),'--r');
plot(t,C(:,2),'-b');
plot(t,C(:,3),'k+:');
plot(t,C(:,4),'rx:');
plot(t,C(:,5),'-g*');
XLABEL ('TIME');
YLABEL ('CONCENTRATION');
legend ('r=0','r=0.113','r=0.5','r=0.887','r=1');
hold off;

% hold on;
% plot(t,T(:,1),'--r');
% plot(t,T(:,2),'-b');
% plot(t,T(:,3),'k+:');
% plot(t,T(:,4),'rx:');
% plot(t,T(:,5),'-g*');
% XLABEL ('TIME');
% YLABEL ('TEMPERATURE');
% legend ('r=0','r=0.113','r=0.5','r=0.887','r=1');
% hold off;
```

Define Functions used Above

```
function dc=DynamicDiffusionfunction(t,x,root,rate,To,Co)
f=Diffusionfunction(x,root,rate,To,Co);
dc=f';
```

6.12 Generate Hotspot and Testing Hotspot Sensitivity

Main Program

```
function Solve_OneD_Convection_Reator
clc;
clear;
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%

C=1;
T=1;
x0=[C,T]';
zspan=[0,1];
options=odeset('RelTol',1e-4);
[z,x]=ode15s('OneDimensionReactor',zspan,x0,options);
C=x(:,1)'
T=x(:,2)'
%
% figure
% hold on
% plot(z,C,'-ob');
% xlabel('length of reactor');
% ylabel('scaled concentration');
% hold off

%figure
hold on
plot(z,T,'-*r');
xlabel('length of reactor');
ylabel('scaled temperature');
hold off

end;
```

Define Functions above

```
function f=OneDimensionReactor(t,x)
```

```
To=300;          %K
Co=1E-2;        %mol/cm^2       %K
C=x(1);
T=x(2);
De=5.2*10^(-6); % cm^2/s
E=5000;    % J/mol
DeltaH=-96558*100;              %J/mol
U=1E-3;    %m/s, Superficial velocity
Sa=580000*0.025;            % m^2/kg, used P764 Fogler case study
Dab=0.00655;               %m^2/s
rowg=4.57E-1;          %kg/m^3
rowb=1400;          % kg/m^3
Cp=1.0032e+003;                 % for gas, J/kg*K
ko=4.42E-10*0.9;          % Reference reaction rate, m^3/(m^2*s)
Kb=0.0384;       % W/(m*K)
%Kb=2.926e-5;
R=0.003;   % m
Rg=8.314;    % J/mol*K, gas constant
a=1.4280;      % 1/s
Ke=2.926e-5;                %W/m*K, heat conductivity


%beta=DeltaH*De*Co/(Ke*To);
beta=DeltaH*De*Co/(Ke*To);
gamma=E/(Rg*To);
Thiele=R*(1/De*(a*exp(-E/(Rg*To))))^0.5;
Bio=0.5;
Thiele1=(Thiele^2*exp(gamma*(1-1/T)))^0.5;


Factor=3*Bio/(Thiele1^2)*(Thiele1*coth(Thiele1)-1)/(Thiele1*coth(Thiele1)+Bio-1);
ff(1)=-rowb*Factor*Sa*C*ko/U*exp(gamma*(1-1/T));
%ff(2)=-Kb/rowg/Cp/U*rowb*DeltaH*Factor*Sa/Kb*C*Co/To*ko*exp(gamma*(1-
1/T))-4*0.001*(T-To);
%ff(2)=-1/rowg/Cp/U*rowb*DeltaH*Factor*Sa*C*Co/To*ko*exp(gamma*(1-1/T))-
4*0.001*(T-To);
ff(2)=-DeltaH*Factor*Sa*rowb*Co/(To*Cp*U*rowg)*C*ko*exp(gamma*(1-1/T))-
4*0.01*(T-1)*To;
f=ff';
```

6.13 Generate Dynamic Hotspot

Use the same program to solve dynamic reactor, implement specific effectiveness factor, instead of solving it.