ChE-403 Computer Problem Set

Week 10

Problem 1

Use one of the ODE solvers in Matlab (e.g. *bvp4c*) to solve the real tubular reactor equation for a reaction of order 3:

$$D_A \frac{d^2 C_i}{dz^2} - u \frac{dC_i}{dz} - kC_i^3 = 0$$

Start by putting the equation in dimensionless form.

Compute the results for L/d_p=100 and 5.

You can assume $Pe_A = d_p u/D_a$=2, d_p=0.04 m and k/u= 25 $\frac{m}{mol^2}$.

Compare what you get (i.e. C'(Z)) to the result of a PFR between Z=0 and Z=1.

Hint: in order to use the ODE solver bvp4c, the 2nd order ODE must be converted into a system of 1st order ODEs. To do so, define y=[y1 y2] with y1=Y and y2=$\frac{dY}{dZ}$ and substitute them in your 2nd order equation.

Solution

Axially dispersed PFR

Dimensionless formulation: Z=z/L, $y=C_i/C_{i,0}$

$$0 = \frac{L^2}{C_{i,0}^3} D_A \frac{d^2 C_i}{dz^2} - \frac{L^2}{C_{i,0}^3} u \frac{dC_i}{dz} - \frac{L^2}{C_{i,0}^3} kC_i^3 = \frac{D_A}{{C_{i,0}}^2} \frac{d^2 y}{dZ^2} - \frac{L}{C_{i,0}^2} u \frac{dy}{dZ} - L^2 k y^3$$

$$= \frac{1}{d_p u} \frac{D_A}{{C_{i,0}}^2} \frac{d^2 y}{dZ^2} - \frac{1}{d_p u} \frac{L}{C_{i,0}^2} u \frac{dy}{dZ} - \frac{1}{d_p u} L^2 k y^3 = \frac{1}{{C_{i,0}}^2 P_{eA}} \frac{d^2 y}{dZ^2} - \frac{L}{C_{i,0}^2 d_p} \frac{dy}{dZ} - \frac{L^2 k}{d_p u} y^3$$

Transformation from one 2nd order ODE ⇔ system of 1st order ODEs: by substituting

$$\begin{array}{ccccc} \frac{d^2 y}{dZ^2} & \rightarrow & \frac{dy}{dZ} & \rightarrow & y(Z) \\ \downarrow & & \downarrow & & \downarrow \\ y_2' & \rightarrow & y_2 = y_1' & \leftarrow & y_1 \end{array}$$

we obtain the following system of equations,

$$\begin{cases} g_1' = g_2 \\ g_2' = \frac{P_{eA}L}{d_p} g_2 + \frac{P_{eA}L^2 C_{i,0}^2 k}{d_p u} g_1^3 \end{cases}$$

For a closed-closed system, the boundary conditions are:

$$\begin{cases} z = 0: \quad uC_{i,0} = -D_A \left.\frac{dC_i}{dz}\right|_{z=0_+} + uC_i(z = 0_+) \\ z = L: \quad 0 = \left.\frac{dC_i}{dz}\right|_{z=L} \end{cases}$$

In adimensional form:

$$\begin{cases} Z = 0: \quad 0 = -\frac{D_A}{Lu} \left.\frac{dy}{dZ}\right|_{Z=0_+} + y(Z = 0_+) - 1 = 1 - \frac{D_A}{Lu} g_2|_{Z=0_+} + g_1|_{Z=0_+} \\ Z = 1: \quad 0 = \left.\frac{dy}{dZ}\right|_{Z=L} = g_2|_{Z=L} \end{cases}$$

The system of 1^{st} order ODEs along with the boundary conditions can now be solved using the function bvp4c in Matlab.

PFR

$$u\frac{dC_i}{dz} + kC_i^3 = 0$$

Dimensionless formulation: Z=z/L, $y=C_i/C_{i,0}$

$$\frac{uL}{C_{i,0}^3}\frac{dC_i}{dz} + \frac{kL}{C_{i,0}^3}C_i^3 = 0 \quad \Leftrightarrow \quad \frac{dy}{dZ} = -\frac{kLC_{i,0}^2}{u}y^3$$

This equation can be solved analytically:

$$\int_{y_0}^{y(z)} \frac{dy}{y^3} = -\frac{kLC_{i,0}^2}{u}\int_0^1 dZ \quad \Rightarrow \quad \frac{1}{y(z)} = \frac{2kLC_{i,0}^2}{u}Z + \frac{1}{y_0^2}$$

Code:

Main script:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%   PROBLEM 1: PFR vs. AXIALLY-DISPERSED PFR    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clearvars
clear all
close all

global Pe L k_u dp n C0

% Parameters
L_dp=5;        % L/dp [-]
n=3;           % reaction order
Pe=2;          % axial Peclet number [-]
dp=0.004;      % particle diameter [m]
```

```
L=L_dp*dp;      % reactor length[m]
k_u=25;         % [m/mol^2] if n=3
C0=2;           % initial concentration [M]

% % % AXIALLY DISPERSED PFR % % %
%The system of 1st order ODEs describing the fluid behaviour in a axially dispersed %tubular reactor is a boundary-value problem that can be solved using the bvp4c function. %This function requires as input the system of 1st order ODE (defined in the function %dstate_adpfr), the boundary conditions (defined in bc_adpfr) and an initial guess for %the desired solution (solinit_adpfr). Note all files .m in which the functions are %defined should be located in the same folder.

%The initial guess can be generated using the function bvpinit which require as input %the mesh (discretisation of Z, here from 0 to 1 with step = 0.01) and guessed values for  %[g1 g2] = [1 0].
Z_discretisation=linspace(0,1,101); % discretisation of the integration domain; linspace creates a vector form 0 to 1 with increment (1-0)/(101-1)=0.01
init_guess=bvpinit(Z_discretisation,[1 0]);

%Resolution of the system; the output is stored in the structure solution that contains %the discretisation mesh Z = z/L in solution.x, the concentration g1 = y = C(z)/C0 = %solution.y(1,:), the approximate jacobian dy/dZ solution.yp aswell as the computational %cost statistics.
Solution=bvp4c(@dstate_adpfr,@bc_adpfr,init_guess);

% We can now extract the solution g1, g2 and the Z mesh from the structure Solution
%Z_mesh=Solution.x;
g1=Solution.y(1,:);
g2=Solution.y(2,:);

% For comparison, we compute here the solution for the PFR analytically and
% numerically.

% % %   PFR: analytical  % % %
% Z-discretisation: specify the values Z for which we want to compute the solution zspan_pfr_an=linspace(0,1,101);
% Initial condition, i.e. at Z=0: y=1
y0=1;

% Analytical solution for different order of reaction
if (n>1)
   y_pfr=1./(((n-1).*k_u.*L.*zspan_pfr_an.*C0.^(n-1))+(1/y0.^(n-1))).^(1/(n-1));
elseif(n==1)
   y_pfr=y0*exp(-k_u*L*zspan_pfr_an);
end

% % %   PFR: numerical  % % %
% By contrast, the 1st order ODE describing the fluid behaviour in a PFR is a initial-%value problem and can be solved using the ODE solver ode45. This function requires as %input the ODE equation (defined in dstate_pfr),the Z domain (specified in zspan) and the %initial condition y0 = C(Z=0)/C0 = 1.
zspan=[0 1];
[Z,y]=ode45(@dstate_pfr,zspan,y0);

% % %   PLOT   % % %
fig=figure; hold on;
plot(zspan_pfr_an,y_pfr,'-','Color','k','Linewidth',2.5);
plot(Z,y,'--','Color','g','Linewidth',2.5);
plot(Z_mesh,g1,'--','Color','b','Linewidth',2.5);
xlabel('z/L [-]','Fontsize',20);
ylabel('C(z,t)/C_0 [-]','Fontsize',19);
set(gca,'Units','normalized',...
   'YTick',0:0.2:1,'XTick',0:0.2:1,...
   'Position',[.15 .2 .75 .7],...
      'FontUnits','points','FontWeight','normal','FontSize',15,'FontName','Times');
ylim([0 1])
xlim([0 1])
title('PFR vs AXIALLY DISPERSED PFR');
legend('PFR (analytical)','PFR (numerical)','Axially dispersed PFR','Location','Best');
print('pfr.pdf','-dpdf');
```

Functions (in separated files for versions of matlab <2016b):

```
%The system of 1st order ODEs is written in the vectorial form dg/dz=[dg1/dz dg2/dz]. %This function, called
```

dstate_adpfr, takes as argument Z (the discretisation of the Z %domain)and g=[g1 g2], the solution of the ODE and return the vector dg/dz=[dg1/dz %dg2/dz].

```
function dgdz=dstate_adpfr(Z,g)
   global Pe L k_u n dp C0

   dgdz(1)=g(2);
   dgdz(2)=((Pe*L)/dp)*g(2)+((Pe*k_u*L^2*C0^(n-1))/(dp))*g(1)^n;
end
```

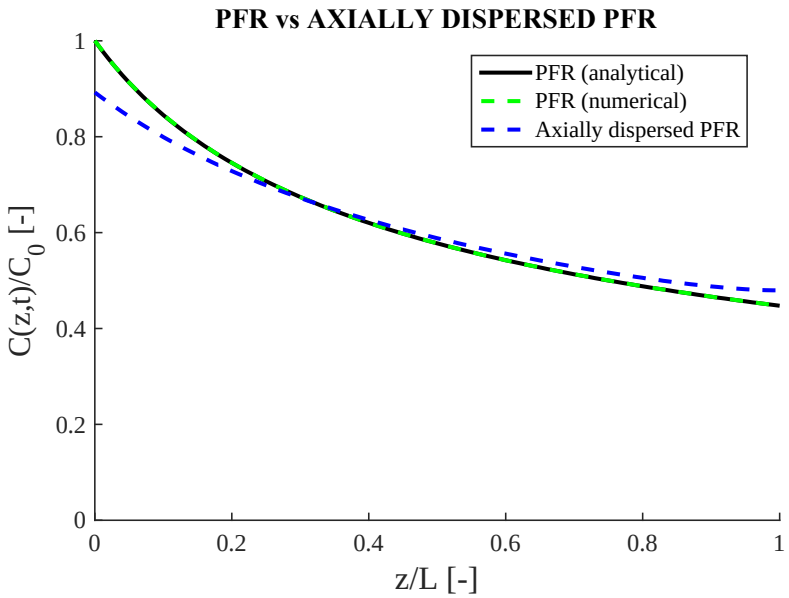
% Boundary conditions (BC) for the axially dispersed PFR system. We have here two %boundary conditions: the first one for Z=0 (bc1) and the second one (bc2) for Z=1. These %are stored in the vector BC = [bc1 bc2]. Here, the function called bc_adpfr takes as %argument g0, representing the solution g=[g1 g2] evaluated at the Z=0, and the solution %g1, representing the solution g=[g1 g2] evaluated at Z=1. This function returns the %vector BC=[bc1 bc2] containing the two boundary conditions.

```
function BC=bc_adpfr(g0,g1)
   global Pe L dp

   BC(1)=1-g0(1)+(dp/(Pe*L))*g0(2);
   BC(2)=g1(2);

end
```

```
function yz=dstate_pfr(z,y)
   global L k_u n C0

   yz=-(k_u*L*C0^(n-1))*y^n;

end
```

```
function res=bc_pfr(ya,yb)

   res(1)=ya(2);
   res(2)=yb(2);

   res=res';
end
```

Output:



Problem 2

Consider a second order reaction taking place in a porous catalyst pellet (this is the next part of the course but I want to show you how to solve this type of math). For a spherical particle of radius R=0.005 m, the equation describing the simultaneous diffusion and reaction inside the catalyst is given by:

$$\frac{\partial C(r,t)}{\partial t} = D_{eff}\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial C(r,t)}{\partial r}\right] - kC(r,t)^2$$

where D_{eff} = 5e-8 m^2/min is the effective diffusivity in the pores and k = 1e-2 $\frac{L}{mol\cdot s}$ is the rate constant.
Assuming a constant concentration C_0 = 0.1 M in the bulk solution surrounding the catalyst and that there is no flux at the center of the particle, compute the concentration profile inside the pellet as a function of time.

Start by putting the equation in a dimensionless form by substituting u=C/C_0, x=r/R. Use then the PDE solver *pdepe* in Matlab to solve it.

Solution
Dimensionless formulation: x=r/R, u(r,t)=C(r,t)/C_0

$$\frac{R^4}{C_0^4}\frac{\partial C(r,t)}{\partial t} = \frac{R^4}{C_0^4}D_{eff}\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial C(r,t)}{\partial r}\right] - \frac{R^4}{C_0^4}kC(r,t)^2$$

$$\Longrightarrow \quad \frac{\partial u(x,t)}{\partial t} = \frac{1}{R^2}D_{eff}\frac{1}{x^2}\frac{\partial}{\partial x}\left[x^2\frac{\partial u(x,t)}{\partial x}\right] - kC_0u(x,t)^2$$

The boundary conditions are given by

$$\begin{cases} r = 0: \quad 0 = \left.\dfrac{dC(r,t)}{dr}\right|_{r=0} & no\ flux\ at\ the\ center\ of\ the\ particle \\ r = R: \quad C_0 = u(r=R,t) & constant\ concentration\ in\ the\ bulk \end{cases}$$

In a dimensionless form

$$\begin{cases} x = 0: \quad 0 = \left.\dfrac{du(x,t)}{dx}\right|_{x=1} \\ x = 1: \quad 0 = u(x=0,t) - 1 \end{cases}$$

Initially, the concentration inside the particle is 0: $u(x, t = 0) = 0$. The dimensionless PDE along with the boundary and initial conditions can now be solved using the function pdepe in Matlab.

<u>Code:</u>

Main script:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%   PROBLEM 2: DIFFUSION AND REACTION INSIDE SPHERICAL CATALYST PELLET    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clearvars
clear all
close all

%%%   PARAMETERS   %%%
global solution m x t R De k C0

R=0.002;
De=5e-8;
k=1e-2;
C0=0.1;

% Specify the symmetry of the system: m=0 (slab), m=1 (cylindrical), m=2 (spherical). %Here, we consider the diffusion-reaction into a spherical catalyst pellet --> m=2
m=2;

% The time and space domain must be discretised. Define a vector specifying the %discretisation in the x direction and the time discretisation
x=linspace(0,1,101); % linspace creates a vector starting from 0 to 1 with increment of (1-0)/(101-1)=0.01
t=linspace(0,60,61); % vector from 0 to 60 with increment of 1

%%%   DIFFUSION AND REACTION   %%%
% Execute the file performing the integration
pde_diff_reac

% Extract the solution
u = solution(:,:,1);

% Plot
fig=figure;
surf(t,x,u')
ylabel('r/R [-]','FontSize',20)
xlabel('t [s]','FontSize',20)
zlabel('C(r,t)/C_0 [-]','FontSize',19)
set(gca,'Units','normalized',...
    'YTick',0:0.2:1,'XTick',0:20:60,...
    'Position',[.15 .2 .75 .7],...
        'FontUnits','points','FontWeight','normal','FontSize',15,'FontName','Times');
zlim([0 1])
title('DIFFUSION AND REACTION');
print('cat_spherical_diff_reac.pdf','-dpdf');
```

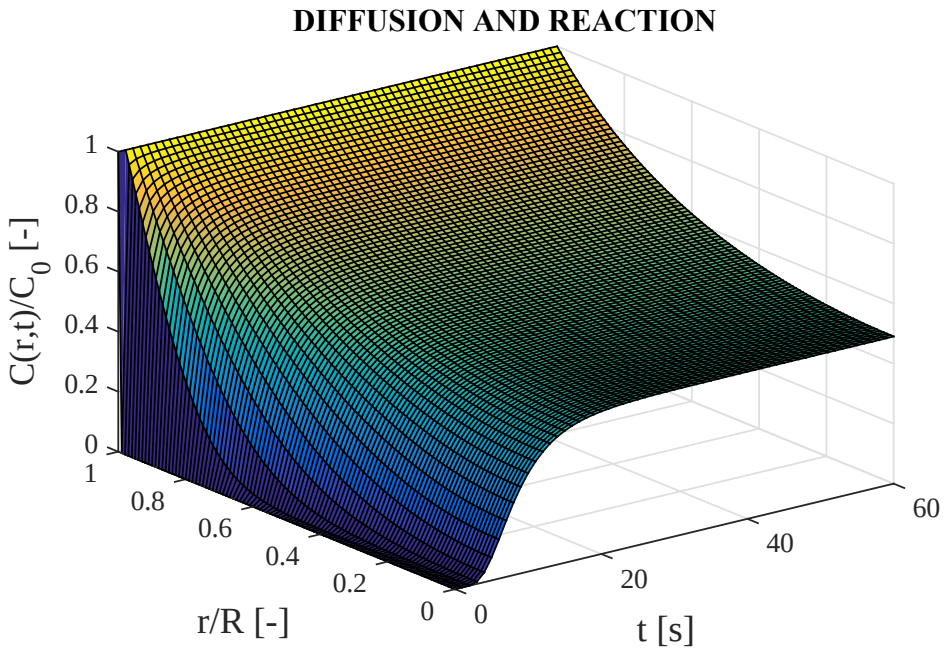
Functions (in separated files for versions of matlab <2016b):

```
% For sake of simplicity, all the functions required to integrate our PDE are stocked in %the same file. The first function pde_diff_reac calls thematlab internal function pdepe %to integrate our pde equation; pdepe requires as inputs the symmetry of the system, the %expression of the pde to solve (defined in dstate_pde), the initial condition (defined %in ic_pde), the boundary conditions (in bc_pde) and the integration variables x,t. The %solution is stored in the matrix of size x*t called here solution.

function pde_diff_reac

global solution m x t

solution = pdepe(m,@dstate_pde,@ic_pde,@bc_pde,x,t);

% ------------------------------------------------------------
function [c,f,s]=dstate_pde(x,t,u,dudx)

global R De k C0

% pdede solves pde of the form: c*du/dt = x^(-m)*d(f*x^m)/dt + s where c, f and s can be %function of x,t,u and du/dx.
```

The equation has to be put into this specific form and then %c, f and s can be specify in the dstate_pde function. The function dstate_pde takes as %arguments (we do not choose, the pdepe algorithm is designed like this) x and t values, the solution u and its derivative du/dx, here written as dudx.

```
c=1;
f=(De/R^2)*dudx;
s=-(k/C0)*u^2;
% ---------------------------------------------------------------
function u0=ic_pde(x)
% Initially, the concentration inside the particle is 0. Again, as specified by the pdede %algorithm, the function defining the initial conditions takes as argument x.
u0=0;
% ---------------------------------------------------------------
function [pl,ql,pr,qr]=bc_pde(xl,ul,xr,ur,t)

% The boundary conditions should be specified in this specific form: q*du/dx + p = 0. p %can be function of (x,t,u) while q can be function of (x,t). Here, the function defining %the boundary conditions takes as argument the value xl of x at the left boundary (in %this case xl would be 0) and its solution ul, the value xr of x at the right boundary %(here, xr would be 1) and its solution ur as well as the time.

% p and q for the left boundary condition (here, x=0)
pl=0;
ql=1;

% p and q for the right boundary condition (here, x=1)
pr=ur-1;
qr=0;
```

Output:



Problem 3

G. Thodor and C.F. Stutzman investigated the reaction of ethylene and HCl in the presence of methane over Zirconium oxide-silica gel catalysts. The reaction is an equilibrium reaction that has the following stoichiometric form:

$$C_2H_4 + HCl = C_2H_5Cl$$

Because the reaction is a reversible reaction at reaction conditions, it must mean that all steps are reversible. The equilibrium constant for the reaction is equal to 35 at reaction conditions.

Can you find a reaction rate expression that describes the following data:

If the equilibrium constant for the reaction is 35 at reaction conditions, find a reaction rate expression that describes the following data:

	Partial pressure (atm)			
$r \times 10^4$ (mol/h/(lb cat))	CH_4	C_2H_4	HCl	C_2H_5Cl
2.66	7.005	0.300	0.370	0.149
2.61	7.090	0.416	0.215	0.102
2.41	7.001	0.343	0.289	0.181
2.54	9.889	0.511	0.489	0.334
2.64	10.169	0.420	0.460	0.175
2.15	8.001	0.350	0.250	0.150
2.04	9.210	0.375	0.275	0.163
2.36	7.850	0.400	0.300	0.208

Solution:

The following reaction is studied over a zirconium oxide-silica gel catalyst:

$C_2H_4 + HCl = C_2H_5Cl$ (4)

Step 1: Assume a plausible reaction mechanism.

$C_2H_4 + * \rightleftharpoons C_2H_4*$

$HCl + * \rightleftharpoons HCl*$

$C_2H_4* + HCl* \rightleftharpoons C_2H_5Cl* + *$ (5)

$C_2H_5Cl* \rightleftharpoons C_2H_5Cl + *$

$CH_4 + * \rightleftharpoons CH_4*$

Step 2: Derive a rate expression for this mechanism.

The equilibrium relationships are as follows:

$$K_1 = \frac{[C_2H_4*]}{[C_2H_4][*]},\ K_2 = \frac{[HCl*]}{[HCl][*]},\ K_3 = \frac{[CH_4*]}{[CH_4][*]} \quad \text{and} \quad K_4 = \frac{[C_2H_5Cl*]}{[C_2H_5Cl][*]} \tag{6}$$

The rate of reaction is that of the RDS:

$$\mathrm{r} = \frac{k_1[C_2H_4*][HCl*] - k_{-1}[C_2H_5Cl*][*]}{[*]_o} \tag{7}$$

The site balance is

$$[*]_o = [*] + [C_2H_4*] + [HCl*] + [C_2H_5Cl*] + [CH_4*] \tag{8}$$

After substituting equation (6) into the site balance:

$$[*] = \frac{[*]_o}{1 + K_1[C_2H_4] + K_2[HCl] + K_3[CH_4] + K_4[C_2H_5Cl]} \tag{9}$$

The rate expression in terms of only the reactants and products is:

$$\mathrm{r} = \frac{k_1K_1[C_2H_4][*]_o[HCl]K_2 - k_{-1}K_4[C_2H_5Cl][*]_o}{\left(1 + K_1[C_2H_4] + K_2[HCl] + K_3[CH_4] + K_4[C_2H_5Cl]\right)^2} \tag{10}$$

From the overall equation, equation (4),

$$K_e = \frac{[C_2H_5Cl]}{[C_2H_4][HCl]} = 35 \tag{11}$$

After substituting equation (11) into equation (10) and rearranging

$$\mathrm{r} = \frac{k\left([C_2H_4][HCl] - {}^{[C_2H_5Cl]}\!/_{K_e}\right)}{\left(1 + K_1[C_2H_4] + K_2[HCl] + K_4[C_2H_5Cl] + K_3[CH_4]\right)^2}$$

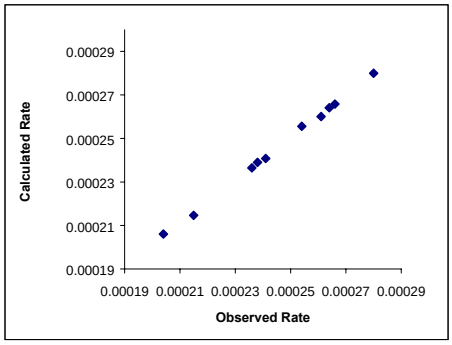
where k is a grouping of constants.

Use nonlinear regression to fit the data to this form.

Each program that is used to solve for the best fit parameters will give a slightly different answer depending on initial values. We obtained the following parameters:

k = 3.86
K_1 = 25.14
K_2 = 32.95
K_3 = 1.98
K_4 = 32.19

The following graph shows that how these parameters represent the data.



Now assume a different mechanism, namely that of Rideal-Eley. The mechanism is as follows:

$$C_2H_4 + * \rightleftharpoons C_2H_4*$$
$$C_2H_4* + HCl \rightleftharpoons C_2H_5Cl* \qquad (12)$$
$$C_2H_5Cl* \rightleftharpoons C_2H_5Cl + *$$
$$CH_4 + * \rightleftharpoons CH_4*$$

The rate is then the rate of the RDS,

$$r = k_2[C_2H_4*][HCl] - k_{-2}[C_2H_5Cl*] \qquad (13)$$

The other three steps are assumed to be quasi-equilibrated, and the equilibrium expressions are the following:

$$K_1[C_2H_4][*] = [C_2H_4*]$$
$$K_3[C_2H_5Cl][*] = [C_2H_5Cl*] \qquad (14)$$
$$K_4[CH_4][*] = [CH_4*]$$

The site balance is the following:

$$[*]_o = [*] + [C_2H_4*] + [C_2H_5Cl*] + [CH_4*] \qquad (15)$$

After substituting equations (14) and (15) into equation (13), the rate is given by the following expression:

$$r = \frac{k\left([C_2H_4][HCl] - {[C_2H_5Cl]}/{K_e}\right)}{\left(1 + K_1[C_2H_4] + K_4[C_2H_5Cl] + K_3[CH_4]\right)} \qquad (16)$$

However, after fitting this expression to the data, negative numbers are found for some of the constants. Since this is unreasonable, the Rideal-Eley mechanism does not fit the data.

Code:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%   PROBLEM 3: REACTION OF ETHYLENE WITH HCl    %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clearvars
clear all
close all

global Ke

%%%   EXPERIMENTAL DATA   %%%
% data = [P_CH4; P_C2H4; P_HCl; P_C2H5Cl; rate] [atm; mol/h/(lb cat)]
data=[7.005 7.090 7.001 9.889 10.169 8.001 9.210 7.850 10.010 8.503;
    0.300 0.416 0.343 0.511 0.420 0.350 0.375 0.400 0.470 0.500;
    0.370 0.215 0.289 0.489 0.460 0.250 0.275 0.300 0.400 0.425;
    0.149 0.102 0.181 0.334 0.175 0.150 0.163 0.208 0.256 0.272;
    2.66 2.61 2.41 2.54 2.64 2.15 2.04 2.36 2.38 2.80];
Ke=35;

%%%   DETERMINATION OF k, K1, K2, K3, K4 FROM NON-LINEAR FITTING   %%%

% Non-linear fit can be done using the lsqcurvefit function. This function requires as %inputs the expression of the fitting function y=f(K,x) (defined in fun), an initial %guess for the fitted parameters (specified in K0 = [k_0 K1_0 K2_0 K3_0 K4_0]) and %experimental values to be fitted (x=[P_CH4; P_C2H4; P_HCl; P_C2H5Cl], y=r).This function %returns then the fitted parameters K = [k K1 K2 K3 K4] and the residual norm. In this %case, for the algorithm to converge to a solution, the number of step must be increased % (default value: 200; here, we increased it to 500 in the options). Since the overall %syntax for lsqcurvefit is:[K,resnorm]=lsqcurvefit(@fun, K0, x, y, lb, ub, options), the %algorithmis expecting as input the lower and upper bound for the solution before %specifying the options. Thus, we also define two empty vectors to give to the function %lsqcurvefit

% Initial guess
K0=[35 35 35 35 35];

% Modify the convergence criteria
options=optimset('MaxFunEvals',500);

% Specify lower bound lb and upper bound ub for the solution
lb=[];
ub=[];

% Experimental data to be fitted
x=data(1:4,:);
y=data(5,:);
[K, resnorm]=lsqcurvefit(@fun_pb3,K0,x,y,lb,ub,options);

% Back calculate r
r_nonlinear=fun_pb3(K,x);

% Compute R^2
R2_nl=1-sum((r_nonlinear-y).^2)/sum((y-mean(y)).^2);

%%%   SUMMARY   %%%
prec=3;

fig3=figure; hold on;
scatter(data(5,:),r_nonlinear,80,'filled','r')
plot([2 3],[2 3],'--','Color','k')
set(gca,'Units','normalized',...
    'Position',[.15 .2 .75 .7],...
        'FontUnits','points','FontWeight','normal','FontSize',15,'FontName','Times');
xlabel('r_{exp} [1/s]','FontSize',20);
ylabel('r_{fit} [1/s]','FontSize',19);
ylim([2 3]);
legend(['Fit: k=',num2str(K(1),prec),', K_1=',num2str(K(2),prec) ,', K_2=', num2str(K(3),prec),', K_3=', num2str(K(4),prec),', K_4=', num2str(K(5),prec),', R^2=', num2str(R2_nl,prec)],'Location','Best');
title('Problem 3');
print('pb3_summary','-dpdf');
```

Functions (in separated files for versions of matlab <2016b):

```
function F=fun_pb3(K,P_data)

global Ke

F=(K(1).*(P_data(2,:).*P_data(3,:)-
(P_data(4,:)/Ke))./((1+K(2).*P_data(2,:)+K(3).*P_data(3,:)+K(5).*P_data(4,:)+K(4).*P_data(1,:)).^2);

end
```

Output: