Introduction to Reactor Design, 3K4

General heat balance equation:

$$\frac{d\,E_{\text{system}}}{d\,t} = 0 = \dot{Q} - \dot{W}_s - F_{A0}\sum_{i=1}^{n}\Theta_i Cp_i\,(T_i - T_0) - \Big[\Delta H_R^{\circ}(T_R) + \Delta Cp(T - T_R)\Big]F_{A0}X$$

- $\dot{Q}$ = rate of heat added or removed from the system [J/s]
- $\dot{W}_s$ = rate of done on the environment by the system (often ≈ 0) [J/s]
- F_{A0} = molar flow of basis species A [mol/s]
- X = conversion of basis species A [-]
- n = number of species in the system
- $\Theta = \dfrac{F_{i0}}{F_{A0}}$ [-]
- Cp_i = heat capacity of species i, [J.mol^{-1}.K^{-1}]
- $T_i = T$ = temperature of the system (assuming it is well mixed) [K]
- T_0 = entry temperature of the system [K]
- T_R = reference temperature = 298 K
- $\Delta H_R^{\circ}(T_R)$ = heat of reaction occurring in the system at T_R [J.(mol of A reacted)$^{-1}$]
- $\Delta Cp = \dfrac{d}{a}Cp_D + \dfrac{c}{a}Cp_C - \dfrac{b}{a}Cp_B - \dfrac{a}{a}Cp_A$ = change in heat capacity [J.(mol of A reacted)$^{-1}$.K^{-1}]

Adiabatic operation

With adiabatic operation we have $\dot{Q} = 0$. Additionally, if we assume that $\dot{W}_s \approx 0$, then, solving the above equation for T gives:

$$T = \frac{X\Big[-\Delta H_R^{\circ}(T_R)\Big] \;+\; \sum \Theta_i Cp_i T_0 \;+\; X\Delta Cp T_R}{\sum \Theta_i Cp_i \;+\; X\Delta Cp}$$

Example 8–3 Liquid-Phase Isomerization of Normal Butane

Normal butane, C_4H_{10}, is to be isomerized to isobutane in a plug-flow reactor. Isobutane is a valuable product that is used in the manufacture of gasoline additives. For example, isobutane can be further reacted to form iso-octane. The 2004 selling price of *n*-butane was 72 cents per gallon, while the price of isobutane was 89 cents per gallon.

The reaction is to be carried out adiabatically in the liquid phase under high pressure using essentially trace amounts of a liquid catalyst which gives a specific reaction rate of 31.1 h^{-1} at 360 K. Calculate the PFR and CSTR volumes necessary to process 100,000 gal/day (163 kmol/h) at 70% conversion of a mixture 90 mol % *n*-butane and 10 mol % *i*-pentane, which is considered an inert. The feed enters at 330 K.

Additional information:

$$\Delta H_{Rx} = -6900 \text{ J/mol} \cdot \text{butane}, \quad \text{Activation energy} = 65.7 \text{ kJ/mol}$$

$$K_C = 3.03 \text{ at } 60°\text{C}, \quad C_{A0} = 9.3 \text{ kmol/dm}^3 = 9.3 \text{ kmol/m}^3$$

Butane

$$C_{P_{n\text{-B}}} = 141 \text{ J/mol} \cdot \text{K}$$

i-Pentane

$$C_{P_{i\text{-P}}} = 161 \text{ J/mol} \cdot \text{K}$$

$$C_{P_{i\text{-B}}} = 141 \text{ J/mol} \cdot \text{K} = 141 \text{ kJ/kmol} \cdot K$$

How to handle temperature as a function of X; first assume

$$T = \frac{X\left[-\Delta H_R^\circ(T_R)\right] + \sum \Theta_i Cp_i T_0 + X\Delta CpT_R}{\sum \Theta_i Cp_i + X\Delta Cp}$$

- $\Delta Cp =$
- $\Theta_A =$
- $\Theta_B =$
- $\Theta_I =$
- $\sum \Theta_i Cp_i T_0 =$
- $\sum \Theta_i Cp_i =$
- $T =$

pfr_example.m:

```
function [d_depnt__d_indep, CA] = pfr_example(indep, depnt, param)

% Dynamic balance for the reactor
%
%    indep: the independent ODE variable, such as time or length
%    depnt: a vector of dependent variables
%    Returns d(depnt)/d(indep) = a vector of ODEs

% Assign some variables for convenience of notation
X = depnt(1);

% Constants. Make sure to use SI for consistency
FT0 = 163000/3600;% mol/s (was kmol/hour originally)
FA0 = 0.9 * FT0;  % mol/s
T1 = 360;         % K
T2 = 333;         % K
E = 65700;        % J/mol
R = 8.314;        % J/(mol.K)
HR = -6900;       % J/(mol of n-butane)
CA0 = 9300;       % mol/m^3
k_1 = 31.1/3600;  % 1/s (was 1/hr originally)
K_Cbase = 3.03;   % [-]

% Equations
T = 43.3*X + param.T_0;                % derived in class, from the heat balance
k1 = k_1 * exp(E/R*(1/T1 - 1/T));      % temperature dependent rate constant
KC = K_Cbase * exp(HR/R*(1/T2 - 1/T)); % temperature dependent equilibrium constant
k1R = k1 / KC;                         % reverse reaction rate constant
CA = CA0 * (1 - X);                    % from the stoichiometry
CB = CA0 * (0 + X);                    % (differs from Fogler, but same result)
r1A = -k1 * CA;                        % rate expressions derived in class
r1B = -r1A;
r2B = -k1R * CB;
r2A = -r2B;
rA = r1A + r2A;                        % total reaction rate for species A

n = numel(depnt);
d_depnt__d_indep = zeros(n,1);
d_depnt__d_indep(1) = -rA / FA0;
```

---

driver.m:

```
% The independent variable always requires an initial and final value:
indep_start = 0.0;  % m^3
indep_final = 5.0;  % m^3

% Set initial condition(s): for integrating variables (dependent variables)
X_depnt_zero = 0.0;   % i.e. X(V=0) = 0.0
```

```
% Other parameters
param.T_0 = 330;      % feed temperature [K]

% Integrate the ODE(s):
[indep, depnt] = ode45(@pfr_example, [indep_start, indep_final], ...
                   [X_depnt_zero], optimset(), param);

% Deal with the integrated output to show interesting plots
X = depnt(:,1);
T = 43.3.*X + param.T_0;          % what was the temperature profile?
rA_over_FA0 = zeros(numel(X), 1); % what was the rate profile?
C_A = zeros(numel(X), 1);         % what was the concentration profile?
for i = 1:numel(X)
    [rA_over_FA0(i), C_A(i)] = pfr_example([], X(i), param);
end

% Plot the results
f=figure;
set(f, 'Color', [1,1,1])
subplot(2, 2, 1)
plot(indep, X); grid
xlabel('Volume, V [kg]', 'FontWeight', 'bold')
ylabel('Conversion, X [-]', 'FontWeight', 'bold')

subplot(2, 2, 2)
plot(indep, T); grid
xlabel('Volume, V [kg]', 'FontWeight', 'bold')
ylabel('Temperature profile [K]', 'FontWeight', 'bold')

subplot(2, 2, 3)
plot(indep, C_A); grid
xlabel('Volume, V [kg]', 'FontWeight', 'bold')
ylabel('Concentration C_A profile [K]', 'FontWeight', 'bold')

subplot(2, 2, 4)
plot(indep, rA_over_FA0); grid
xlabel('Volume, V [kg]', 'FontWeight', 'bold')
ylabel('(Reaction rate/FA0) profile [1/m^3]', 'FontWeight', 'bold')

% Now plot one of the most important figures we saw earlier in the course:
% F_A0 / (-rA) on the y-axis, against conversion X on the x-axis. This plot
% is used to size various reactors.

% The material leaves the reactor at equilibrium; let's not plot
% that far out, because it distorts the scale. So plot to 95% of
% equilibrium
f = figure; set(f, 'Color', [1,1,1])
index = find(X>0.95 * max(X), 1);
plot(X(1:index), 1./rA_over_FA0(1:index)); grid
xlabel('Conversion, X [-]', 'FontWeight', 'bold')
ylabel('FA0/(-r_A) profile [m^3]', 'FontWeight', 'bold')
```