Reaction Engineering Basics Example 8.4.2

Matlab Work-Up

Before summarizing the assignment, the file, `ex_8_4_2.m`, was created for performing the calculations. Within it a function of the same name was declared. Each of the blocks of code listed below are added to this file to complete the function for performing the calculations.

```
function ex_8_4_2()
% Calculations for Example 8.4.2 from REB, The Book
```

Assignment Summary

Reactions

$$A + B \rightarrow D \tag{1}$$

$$2B \rightarrow U \tag{2}$$

Rate Expressions

$$r_1 = k_1 C_A C_B \tag{3}$$

$$r_2 = k_2 C_B^2 \tag{4}$$

Reactor: Adiabatic, liquid-phase SBSTR with a 2 stage operational protocol

Given Constants: $k_{0,1}$ = 1.83 x 10^{12} L mol^{-1} h^{-1}, E_1 = 18.0 kcal mol^{-1}, $k_{0,2}$ = 5.08 x 10^{13} L mol^{-1} h^{-1}, E_2 = 20.5 kcal mol^{-1}, ΔH_1 = -9000 cal mol^{-1}, ΔH_2 = -7800 cal mol^{-1}, $\breve{C}_p$ = 863 cal L^{-1} K^{-1}, $C_{A,0}$ = 2 M, $C_{B,0} = C_{D,0} = C_{U,0} = 0$, $C_{B,in}$ = 0.5 M, $C_{A,in} = C_{D,in} = C_{U,in} = 0$, $T_0 = T_{in}$ = 40 °C, P = 1 atm, V_0 = 2000 L, V_B = 8000 L, and t_f = 8 h.

The given and known constants are converted to consistent units and addded to `ex_8_4_2`.

```
% global constants available to all functions
% given
k0_1 = 1.83E12; % L/mol/h
E_1 = 18000.0; % cal/mol
k0_2 = 5.08E13; % L/mol/h
E_2 = 20500.0; % cal/mol
dH_1 = -9000.0; % cal/mol
dH_2 = -7800.0; % cal/mol
Cp = 863.0; % cal/L/K
CA_0 = 2.0; % mol/L
CB_in = 0.5; % mol/L
T_0 = 40 + 273.15; % K
T_in = T_0;
P = 1.0; % atm
V_A = 2000.0; % L
V_B = 8000.0; % L
tf = 8.0; % h
t1_range = [1, 3, 5, 7];
% known
Re = 1.987; % cal/mol/K
Rw = 0.08206; % L-atm/mol/K
% calculated
nA_0 = CA_0*V_A;
```

Task: Parametric Analysis

- Parameter: $t_1 = [1, 3, 5, 7]$ h
- Items of Interest: f_B, $S_{D/U}$, and $Y_{D/B}$

Preliminary Analysis

Based upon the rate expressions, the instantaneous selectivity is favored by a high concentration of A and a low concentration of B, equation (5). Assuming negligible temperature effects, starting the process with all of the A in the reactor keeps the concentration of A a large as possible at all times, and adding the B slowly can keep its concentration as low as desired. Both of those factors favor high selectivity for D over U.

$$S_{D/U,inst} = \frac{r_D}{r_U} = \frac{r_1}{r_2} = \frac{k_D}{k_U}\frac{C_A}{C_B} \tag{5}$$

Presumably the reactions are not extremely fast because the reactor is allowed to operate in batch mode after semi-batch processing ends. As such, the final conversion of B is expected to decrease the longer the addition time because reagent added near the end of semi-batch processing will have less time to react during the batch stage.

As the addition time increases, the selectivity for D over U is expected to increase because the slower the addition of B, the lower its concentration and the higher the selectivity. Yield is affected by both conversion and selectivity, so a trade-off is expected. The decrease in conversion with semi-batch processing time will tend to decrease the yield while the increase in selectivity will tend to increase it. Quantitative analysis is necessary to determine how the yield will vary with semi-batch processing time.

SBSTR Model Function

SBSTR Design Equations

$$\frac{dn_A}{dt} = -r_1 V \tag{6}$$

$$\frac{dn_B}{dt} = \dot{n}_{B,in} - (r_1 + 2r_2)\, V \tag{7}$$

$$\frac{dn_D}{dt} = r_1 V \tag{8}$$

$$\frac{dn_U}{dt} = r_2 V \tag{9}$$

$$V\breve{C}_p \frac{dT}{dt} - P\frac{dV}{dt} = -\dot{V}_{in}\breve{C}_p\,(T - T_{in}) - r_1 V \Delta H_1 - r_2 V \Delta H_2 \tag{10}$$

$$\frac{dV}{dt} = \dot{V}_{in} \tag{11}$$

SBSTR Model Variables: $\underline{t}$, $\underline{n}_A$, $\underline{n}_B$, $\underline{n}_D$, $\underline{n}_U$, $\underline{T}$, and $\underline{V}$

Additional Computable Unknowns: r_1, r_2, k_1, k_2, C_A, C_B, $\dot{n}_{B,in}$, and $\dot{V}_{in}$

$$k_i = k_{0,i} \exp\left(\frac{-E_i}{RT}\right) \qquad i = 1 \text{ and } 2 \tag{12}$$

$$C_i = \frac{n_i}{V} \qquad i = A \text{ and } B \tag{13}$$

$$\dot{n}_{B,in} = \dot{V}_{in} C_{B,in} \tag{14}$$

$$\begin{aligned} \dot{V}_{in} &= \tfrac{V_B}{t_1}; \qquad \text{stage 1} \\ \dot{V}_{in} &= 0; \qquad \text{stage 2} \end{aligned} \tag{15}$$

Uncomputable Unknowns (Received as Arguments): t_1

If a single SBSTR derivatives is used, it needs t_1 and the stage number (see equation (15)), so the SBSTR model function must make them available as global variables. Alternatively separate derivatives functions could be written for each stage, in which case global variables would not be needed.

IVODE Solver Requirements

1. Stage 1

   a. Initial values

$$t_{iv,1} = t_1 \tag{16}$$

$$n_{A,iv,1} = n_{A,0} = C_{A,0} V_A \tag{17}$$

$$n_{i,iv,1} = 0 \qquad i = B, D, \text{ and } U \tag{18}$$

$$T_{iv,1} = T_0 \tag{19}$$

$$V_{iv,1} = V_A \tag{20}$$

   b. Stopping criterion

$$t = t_1 \tag{21}$$

2. Stage 2

   a. Initial values

$$t_{iv,2} = t_1 \tag{22}$$

$$n_{i,iv,2} = n_i\big|_{t=t_1} \qquad i = A, B, D, \text{ and } U \tag{23}$$

$$T_{iv,2} = T\big|_{t=t_1} \tag{24}$$

$$V_{iv,2} = V\big|_{t=t_1} \tag{25}$$

   b. Stopping criterion

$$t = t_f \tag{26}$$

3. (both stages) SBSTR derivatives function that

    a. receives the reactor model variables at the start of an integration step
    b. can access the current value of t_1 and the stage number
    c. calculates the additional unknowns using equations (3), (4), and (12) - (15)
    d. evaluates and returns the SBSTR design equation SBSTR derivatives using equations (6) - (11)

To facilitate making t_1 and the stage number available to the SBSTR derivatives function, the global variables, `g_t1` and `g_stage`, are declared and set to `nan` initially. Then the SBSTR model and SBSTR derivatives functions are added to `ex_8_4_2`.

```
% define a global variables for the current value of t1 and the current protocol stage
g_t1 = nan;
g_stage = -1;

% SBSTR reactor function
function [t, nA, nB, nD, nU, T, V] = sbstr_model_variables(t1)
    % make t1 available to the derivatives function
    g_t1 = t1;

    % set the current operational protocol stage;
    g_stage = 1;

    % set the initial values
    ind_0 = 0;
    dep_0 = [nA_0, 0, 0, 0, T_0, V_A];

    % set the stopping criterion
    stopVar = 0;
    stopVal = t1;

    % solve the IVODE design equations
    odes_are_stiff = false;
    [t1, dep1, success, message] = solve_ivodes(ind_0, dep_0, stopVar...
        , stopVal, @sbstr_derivatives, odes_are_stiff);

    % check for solver issues
    if ~success
        disp('')
        disp(["SBSTR model stage 1 issue: ", message])
        disp('Press any key to continue or CTRL-C to exit')
        pause
```

```
        end

        % repeat for the second stage
        g_stage = 2;
        ind_0 = t1(end);
        dep_0 = [dep1(end,1), dep1(end,2), dep1(end,3), dep1(end,4)...
            , dep1(end,5), dep1(end,6)];
        stopVal = tf;
        [t2, dep2, success, message] = solve_ivodes(ind_0, dep_0, stopVar...
            , stopVal, @sbstr_derivatives, odes_are_stiff);
        if ~success
            disp('')
            disp(["SBSTR model stage 2 issue: ", message])
            disp('Press any key to continue or CTRL-C to exit')
            pause
        end

        % concatenate the stages
        t = [t1; t2];
        nA = [dep1(:,1); dep2(:,1)];
        nB = [dep1(:,2); dep2(:,2)];
        nD = [dep1(:,3); dep2(:,3)];
        nU = [dep1(:,4); dep2(:,4)];
        T = [dep1(:,5); dep2(:,5)];
        V = [dep1(:,6); dep2(:,6)];
    end

    % SBSTR derivatives function
    function ddt = sbstr_derivatives(~, dep)
        % extract the dependent variables
        nA = dep(1);
        nB = dep(2);
        T = dep(5);
        V = dep(6);

        % calculate the additional unknowns
        if g_stage == 1
            Vdot_in = V_B/g_t1;
        else
            Vdot_in = 0.0;
        end
        k_1 = k0_1*exp(-E_1/Re/T);
```

```
        k_2 = k0_2*exp(-E_2/Re/T);
        CA = nA/V;
        CB = nB/V;
        r_1 = k_1*CA*CB;
        r_2 = k_2*CB^2;
        nDotB_in = Vdot_in*CB_in;

        % evaluate the derivatives
        dnAdt = -V*r_1;
        dnBdt = nDotB_in - V*(r_1 + 2*r_2);
        dnDdt = V*r_1;
        dnUdt = V*r_2;
        dTdt = (-Vdot_in*Cp*(T - T_in) - V*r_1*dH_1 - V*r_2*dH_2...
            + P*Vdot_in*Re/Rw)/(V*Cp);
        dVdt = Vdot_in;

        % return the derivatives as a column vector
        ddt = [dnAdt; dnBdt; dnDdt; dnUdt; dTdt; dVdt];
    end
```

Items of Interest Function

1. Loop through the specified values of t_1
    a. Call the SBSTR reactor model function to get $\underline{t}$, $\underline{n}_A$, $\underline{n}_B$, $\underline{n}_D$, $\underline{n}_U$, $\underline{T}$, and $\underline{V}$
    b. Calculate and save the corresponding conversion, selectivity, and yield

$$f_B = \frac{V_B C_{B,in} - \underline{n}_B\big|_{t_f}}{V_B C_{B,in}} \tag{27}$$

$$S_{D/U} = \frac{\underline{n}_D\big|_{t_f}}{\underline{n}_U\big|_{t_f}} \tag{28}$$

$$Y_{D/B} = \frac{\underline{n}_D\big|_{t_f}}{V_B C_{B,in}} \tag{29}$$

2. Tabulate, show, and save the results

To complete `ex_8_4_2`, the items of interest function was added along with a line of code to call it and perform the calculations.

```
    % items of interest function
    function items_of_interest()
        % allocate storage for the results
        nParams = length(t1_range);
        f_B = ones(1, nParams)*nan;
        S_DU = ones(1, nParams)*nan;
        Y_DB = ones(1, nParams)*nan;

        % loop through the t1 values
        for i = 1:nParams
            % solve the design equations
            [~, ~, nB, nD, nU, ~, ~] = sbstr_model_variables(t1_range(i));

            % calculate and save the quantities of interest
            f_B(i) = 100*(V_B*CB_in - nB(end))/(V_B*CB_in);
            S_DU(i) = nD(end)/nU(end);
            Y_DB(i) = 100*nD(end)/(V_B*CB_in);
        end

        % tabulate, show and save the results
        resultsTable = table(t1_range.', f_B.', S_DU.', Y_DB.');
        resultsTable.Properties.VariableNames = ["Stage 1 Duration (h)",...
            "Conversion (%)", "Selectivity D/U", "Yield D-B (%)"];
        disp('')
        disp(resultsTable)
        disp('')
        writetable(resultsTable, 'ex_8_4_2_results.csv')
    end

    % perform the calculations
    items_of_interest()
end
```

Results

The calculations were performed by calling the function, `ex_8_4_2`.

```
>> ex_8_4_2
    Stage 1 Duration (h)    Conversion (%)    Selectivity D/U    Yield D-B (%)
    --------------------    --------------    ---------------    -------------
```

```
1   84.344   3.163   51.671
3   80.651   3.7418  52.558
5   75.428   4.3865  51.807
7   67.649   5.3322  49.196
```

That output was used to generate Table 1.

Table 1: Effect of the duration of stage 1 of the operational protocol on the overall conversion, selectivity, and yield.

Stage 1 Duration (h)	Conversion (%)	Selectivity D/U	Yield D-B (%)
1	84.3	3.17	51.7
3	80.6	3.75	52.6
5	75.4	4.39	51.8
7	67.6	5.33	49.2

Discussion

Table 1 shows that the conversion of B decreases as its addition gets spread over longer times. This is reasonable because as noted in the preliminary analysis, the later in the process that the B is added, the less time it has to react. Separate calculations show that if the reaction is run in a batch reactor the conversion is 85.7% with a selectivity for D over U of 2.88 and a yield of D from B of 50.6%.

As previously noted, adding the B to A over time keeps the concentration of A as large as possible at all times and keeps the concentration of B lower than if it was all added at the start. The expression for the instantaneous selectivity for D over U for these reactions, equation (5), suggests that this should result in greater selectivity, and indeed, Table 1 shows that the slower the B is added (i. e. the longer the time during which it is added), the greater the selectivity for D over U.

However, while the selectivity is increasing, the conversion of B is decreasing. As a consequence, the yield of D from B is nearly constant at approximately 50%, passing through a weak maximum when B is added over 3 h. The processing time in all cases is the same, 8 h. Hence, the operating costs, are basically the same. The net rate of production of D is essentially the same, too, because the yield of D is effectively constant.

In effect, the same amount of B is converted to D in all four cases, but as the addition time increases, the amount of B converted to U decreases while the amount of unreacted B increases. Comparing batch processing to semi-batch processing where the B is added during the first 7 h of the 8 h process, the same amount of D is produced at the same net rate. The difference is

that in semi-batch mode there is more unreacted B while in batch mode there is more undesired U.

Semi-batch processing where the B is added during the first 7 h of the 8 h process has an advantage over batch processing in two scenarios. First, if reagent B is easily and inexpensively separable from the product mixture, not converting it to U, but instead recovering it and re-using it in a subsequent semi-batch run mitigates the low conversion.

The other scenario is when U is a valueless hazardous waste. The disposal of such a waste can add significantly to the cost of the process, thereby decreasing profits. In this scenario, even if the unconverted B cannot be separated and re-used, the cost of wasting reagent B (i. e. low conversion) may be more than offset by significantly lower expenses associated with disposal of U (because less U is generated).

Summary

Table 1 summarizes the effect of varying the rate of addition of B upon conversion, selectivity and yield. As the rate of addition increases, the conversion increases and the selectivity decreases with the yield remaining approximately constant.

As the semi-batch processing time is increased, holding the total processing time constant, the conversion of B decreases from 84.3 to 67.6%, the selectivity for D over U increases from 3.17 to 5.33, and the yield of D is close to constant at 50%. In consequence, as the semi-batch processing time increases the amount of undesired product decreases and the amount of unreacted B increases.