# A First Course on Kinetics and Reaction Engineering

## Example 39.1

### Problem Purpose

This example illustrates the analysis, using the two film model, of a reactor wherein a gas-liquid reaction occurs.

### Problem Statement

A gas mixture containing 15 % A is fed to an isothermal steady state CSTR at a rate of 0.002 m$^3$ min$^{-1}$. A solution containing B at a concentration of 3.0 mol L$^{-1}$ is also fed to the reactor at a rate of 0.06 L min$^{-1}$. The reactor system operates at a constant temperature of 20 °C and a constant pressure of 5.0 atm, and the liquid density is essentially constant. An agitator/gas recirculation system perfectly mixes each of the two phases, separately. The liquid phase volume is equal to 1.0 L, and the gas is dispersed as bubbles in the liquid such that the bubble area per liquid volume is constant and equal to 445 m$^{-1}$. The partial pressure mass transfer coefficient for A is equal to 0.83 mol m$^{-2}$ atm$^{-1}$ s$^{-1}$, the liquid phase mass transfer coefficient for A is equal to 0.053 cm s$^{-1}$ and the Henry's law constant for A is equal to 50.0 L atm mol$^{-1}$. The diffusion coefficients for A and B may be taken to be constant and equal to 2.1 x 10$^{-5}$ and 7.25 x 10$^{-6}$ cm$^2$ s$^{-1}$, respectively. A is the only gas phase component that is soluble in the liquid phase, and none of the components of the liquid are volatile. Thus reaction (1) takes place only within the liquid phase where the rate expression is given by equation (2) with the rate coefficient equal to 605 L mol$^{-1}$ min$^{-1}$. Ignoring the curvature of the bubble surface and assuming that the amount of reaction taking place in the liquid film is significant, calculate the conversion of A.

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

$$r = k[A][B] \tag{2}$$

### Problem Analysis

The problem statement describes a steady state system wherein the gas phase component, A, is transferred to the liquid phase where it reacts with B. The problem statement alludes to the liquid film, so this mass transfer will be modeled using the two film model described in the informational reading for this unit. A schematic representation of the system is presented in Figure 1. The bulk gas, on the left side of the left dashed line is perfectly mixed; the partial pressure of A in this region is denoted as $P_{A,g}$. The gas-liquid interface is represented by the vertical solid line; the region between the left dashed line and the gas-liquid interface is the gas film. The interface is defined as the origin of the $x$ axis with increasing $x$ being in the direction perpendicular to the interface and into the liquid (to the right of the interface as drawn). At the interface, the gas phase pressure of A is denoted as $P_{A,i}$, and the liquid phase concentration of A as $C_{A,i}$; these two quantities are assumed to be related to each other through Henry's

law. The region between $x = 0$ and $x = x_L$ (right dashed line) represents the liquid film. Within this region, the concentrations of A and B are denoted as $C_{A,f}$ and $C_{B,f}$, and both of these quantities vary with $x$. The region on the right, where $x > x_L$ represents the bulk liquid which is also perfectly mixed. The bulk liquid concentrations of A and B are denoted as $C_{A,l}$ and $C_{B,l}$, and they do not depend on $x$. A gas mixture containing A as one component flows into the reactor and a gas mixture containing less A flows out of the reactor. These flows are denoted as $\dot{n}_{A,g}^0$ and $\dot{n}_{A,g}$, respectively. A liquid feed containing B enters the reactor, with the inlet molar flow rate of B denoted as $\dot{n}_{B,l}^0$. Since the density of the liquid is constant, a liquid stream with the same volumetric flow rate, $\dot{V}_L$, leaves the reactor; the molar flow rates of A and B in this stream are denoted as $\dot{n}_{A,l}$ and $\dot{n}_{B,l}$.

The system can be modeled by writing and solving separate mole balances on each of the two bulk phases present. Since the system is isothermal, the mole balances can be solved independent of energy balances. Since reaction only occurs in the liquid phase, only the liquid phase mole balance will include a rate term, but the balances for both phases will include terms representing the transfer of the reagent A from the gas phase into the gas film or from the liquid film into the bulk liquid, as appropriate. The mole balances on the two bulk phases will be coupled to mole balances on the liquid film such that analysis of the reactor will require solution of all of the mole balances.
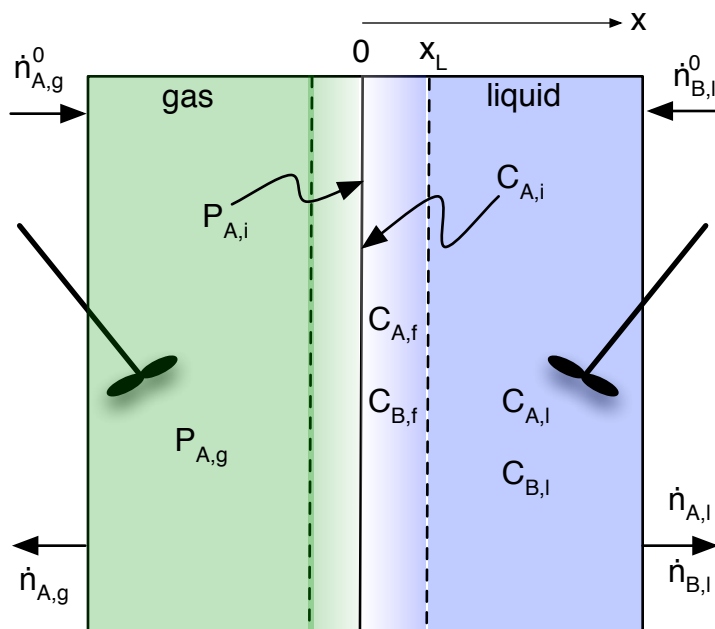


*Figure 1. Schematic representation of the gas-liquid reactor (not to scale).*

**Problem Solution**

The following quantities are specified in the problem statement: $y_{A,g}^0 = 0.15$, $\dot{V}_g^0 = 0.002$ m³ min⁻¹,
$C_{B,l}^0 = 3.0$ mol L⁻¹, $\dot{V}_L = 0.06$ L min⁻¹ (the liquid density is constant, so the inlet and outlet liquid phase

volumetric flow rates are equal), $T$ = 20 °C, $P$ = 5.0 atm, $V_L$ = 0.001 m³, $S_V$ = 445 m⁻¹, $k_{G,P}$ = 0.83 mol m⁻² atm⁻¹ s⁻¹, $k_L$ = 0.053 cm s⁻¹, $h_A$ = 50.0 L atm mol⁻¹, $D_A$ = 2.1 x 10⁻⁵ cm² s⁻¹, $D_B$ = 7.25 x 10⁻⁶ cm² s⁻¹, and $k$ = 605 L mol⁻¹ min⁻¹. Since the diffusion coefficient of A in the liquid and the liquid mass transfer coefficient are specified, the liquid film thickness can be calculated according to equation (3). The problem asks that the conversion of A, $f_A$, be calculated.

$$x_L = \frac{D_A}{k_L} \tag{3}$$

The problem statement indicates that the gas phase may be assumed to be perfectly mixed. As such, a gas phase mole balance on the reactant A will resemble an ideal CSTR mole balance except there will be no reaction rate term. Instead, there will be a term representing the A that is transferred to the liquid phase. If $N_A$ is used to represent the flux of A in the $x$ direction, the bulk gas phase mole balance on A takes the form given in equation (4). In writing this equation it was noted that the flux of A from the bulk gas into the gas film must equal the flux of A from the gas film into the liquid because the system operates at steady state and no reaction occurs in the gas phase.

$$\dot{n}_{A,g}^0 - S_V V_l N_A\big|_{x=0} - \dot{n}_{A,g} = 0 \tag{4}$$

A mole balance on A in the bulk liquid can also be written. Again, since the liquid is perfectly mixed, this will look like a CSTR mole balance with two exceptions. First, there is no A present in the liquid feed, so that term is absent, but, second, A is transferred into the liquid phase from the liquid film at the boundary of that film, $x = x_L$. Again letting $N_A$ represent the flux of A in the $x$ direction, the liquid phase mole balance on A is given in equation (5). The mole balance on B in the bulk liquid phase, equation (6), is also similar to a CSTR mole balance, with the exception that an additional term is needed to represent the B that is transferred from the bulk liquid into the liquid film.

$$S_V V_l N_A\big|_{x=x_L} - kV_l C_{A,l} C_{B,l} - \dot{n}_{A,l} = 0 \tag{5}$$

$$\dot{n}_{B,l}^0 + S_V V_l N_B\big|_{x=x_L} - kV_l C_{A,l} C_{B,l} - \dot{n}_{B,l} = 0 \tag{6}$$

The mole balances on the bulk gas and liquid constitute a set of three non-linear algebraic equations. Hence, these three equations can be solved numerically to find the values of three unknowns. The unknowns can be taken to be the gas phase outlet molar flow rate of A , $\dot{n}_{A,g}$, and the liquid outlet molar flow rates of A and B, $\dot{n}_{A,l}$ and $\dot{n}_{B,l}$. In order to solve the equations numerically for these three quantities, it will be necessary to provide guesses for each of their values along with code that evaluates the left hand sides of equations (4) through (6) given values for the unknowns. Of the quantities appearing in the equations, $S_V$, $V_L$ and $k$ are known from the problem statement. Therefore, in order to

evaluate the left-hand sides of equations (4) through (6), the code we provide must first calculate $\dot{n}^0_{A,g}$,

$N_A\big|_{x=0}$, $N_A\big|_{x=x_L}$, $C_{A,l}$, $C_{B,l}$, $\dot{n}^0_{B,l}$ and $N_B\big|_{x=x_L}$.

The inlet gas phase molar flow rate of A can be calculated from the specified inlet gas phase volumetric flow rate and composition by use of the ideal gas law, equation (7). The inlet liquid phase molar flow rate of B can be computed from the liquid volumetric flow rate and the feed concentration of B, equation (8). Recalling that the code we must provide will be given values of $\dot{n}_{A,g}$, $\dot{n}_{A,l}$ and $\dot{n}_{B,l}$, the ideal gas law can be used to calculate the partial pressure of A in the gas phase, equation (9). In order to do so, the outlet gas volumetric flow rate must be calculated; it won't equal the inlet volumetric flow rate because some A will have been transferred to the liquid phase, equation (10). Application of the ideal gas law gives the outlet gas volumetric flow rate, equation (11). Again recalling that the code we must provide will be given values of $\dot{n}_{A,g}$, $\dot{n}_{A,l}$ and $\dot{n}_{B,l}$, the liquid phase concentrations of A and B can be calculated using the specified liquid volumetric flow rate, equations (12) and (13).

$$\dot{n}^0_{A,g} = \frac{\dot{V}^0_g P}{RT} y^0_{A,g} \tag{7}$$

$$\dot{n}^0_{B,l} = \dot{V}_L C^0_{B,l} \tag{8}$$

$$P_{A,g} = \frac{\dot{n}_{A,g} RT}{\dot{V}_g} \tag{9}$$

$$\dot{n}_{total,g} = \dot{n}^0_{total,g} - \dot{n}^0_{A,g} + \dot{n}_{A,g} = \frac{P\dot{V}^0_g}{RT} - \dot{n}^0_{A,g} + \dot{n}_{A,g} \tag{10}$$

$$\dot{V}_g = \frac{\dot{n}_{total,g} RT}{P} \tag{11}$$

$$C_{A,l} = \frac{\dot{n}_{A,l}}{\dot{V}_l} \tag{12}$$

$$C_{B,l} = \frac{\dot{n}_{B,l}}{\dot{V}_l} \tag{13}$$

At this point, the code we provide to solve equations (4) through (6) still needs to calculate $N_A\big|_{x=0}$, $N_A\big|_{x=x_L}$ and $N_B\big|_{x=x_L}$. In order to do so, it will first need to solve mole balances on A and B in the liquid film (between $x = 0$ and $x = x_L$). The mole balance on A in the liquid film is given in equation (14), with the boundary conditions given in equations (15) and (16). The first boundary condition specifies that Henry's law is obeyed at the interface, and the second boundary condition simply requires the concentration of A to equal the bulk liquid concentration at the interface between the bulk liquid and the liquid film. The mole

balance on B in the liquid film and its boundary conditions are given in equations (17) through (19). The first boundary condition requires the flux of B at the interface to equal zero since B is non-volatile. The second boundary condition is like that for A; it requires the concentration of B to equal the bulk liquid concentration at the interface between the bulk liquid and the liquid film.

$$D_A \frac{d^2 C_{A,f}}{dx^2} = k C_{A,f} C_{B,f} \quad \Rightarrow \quad \frac{d^2 C_{A,f}}{dx^2} = \frac{k}{D_A} C_{A,f} C_{B,f} \tag{14}$$

$$C_{A,f}(0) = C_{A,i} = \frac{P_{A,i}}{h_A} \tag{15}$$

$$C_{A,f}(x_L) = C_{A,l} \tag{16}$$

$$D_B \frac{d^2 C_{B,f}}{dx^2} = k C_{A,f} C_{B,f} \quad \Rightarrow \quad \frac{d^2 C_{B,f}}{dx^2} = \frac{k}{D_B} C_{A,f} C_{B,f} \tag{17}$$

$$N_B\big|_{x=0} = -D_B \frac{dC_{B,f}}{dx}\bigg|_{x=0} = 0 \tag{18}$$

$$C_{B,f}(x_L) = C_{B,l} \tag{19}$$

Equations (14) through (19) constitute a set of boundary value ODEs, and as noted, the code that is provided to solve equations (4) through (6) will need to solve equations (14) through (19) numerically. Supplemental Unit S6 provides an overview of how this is done and the information and code that must be provided. For present purposes it will simply be noted that the only additional quantity that is needed is the interfacial concentration of A or the interfacial partial pressure of A, since they are related as given in equation (15). The interfacial partial pressure can be calculated using the gas phase mass transfer coefficient as shown in equation (20).

$$k_{G,P}\left(P_{A,g} - P_{A,i}\right) = N_A\big|_{x=0} \quad \Rightarrow \quad k_{G,P}\left(P_{A,g} - P_{A,i}\right) - N_A\big|_{x=0} = 0 \tag{20}$$

At first glance it may seem as if we are confronted with a chicken and egg problem. We need the value of $N_A\big|_{x=0}$ in order to solve the bulk phase mole balances. We can calculate $N_A\big|_{x=0}$ upon solving the film mole balances, but in order to solve the film mole balances, we need $N_A\big|_{x=0}$ so we can calculate $P_{A,i}$ using equation (20). It turns out that when the equations are being solved numerically this isn't actually a problem. There are a few different ways to formulate the numerical solution, but some are more confusing than others. Hopefully the approach to be described here is not too confusing.

Let's go back, and instead of solving the just the bulk phase mole balances, equations (4) through (6) for $\dot{n}_{A,g}$, $\dot{n}_{A,l}$ and $\dot{n}_{B,l}$, let's add in equation (20), giving us four equations and let's solve equations (4), (5), (6) and (20) numerically to find values of $\dot{n}_{A,g}$, $\dot{n}_{A,l}$, $\dot{n}_{B,l}$ and $P_{A,i}$. To do so, we'll need to provide

guesses for each of their values along with code that evaluates the left hand sides of equations (4), (5), (6) and (20) *given values for the unknowns*: $\dot{n}_{A,g}$, $\dot{n}_{A,l}$, $\dot{n}_{B,l}$ **and** $P_{A,i}$. The code we provide can still use equations (7) through (13) to calculate $\dot{n}_{A,g}^0$, $C_{A,l}$, $C_{B,l}$, $\dot{n}_{B,l}^0$, and $P_{A,g}$. The problem specifies values for $S_V$, $V_L$, $k$ and $k_{G,P}$, again leaving $N_A\big|_{x=0}$, $N_A\big|_{x=x_L}$ and $N_B\big|_{x=x_L}$ as the only unknown quantities appearing in equations (4), (5), (6) and (20).

As previously noted, these fluxes can be calculated after first solving the film mole balances, equations (14) through (19). Thus the code we provide to evaluate equations (4), (5), (6) and (20) will first have to call a mixed boundary ODE solver. This time, however, the code we provide to evaluate equations (4), (5), (6) and (20) will be given a value for $P_{A,i}$, so it will have all the values that are needed to solve the film mole balances. The numerical solution of equations (14) through (19) will yield values of $C_{A,f}$, $C_{B,f}$, $\dfrac{dC_{A,f}}{dx}$ and $\dfrac{dC_{B,f}}{dx}$ at discrete values of $x$ (mesh points) between 0 and $x_L$. The fluxes required to evaluate the left-hand sides of equations (4), (5), (6) and (20) can then be calculated using equations (21) through (23).

$$N_A\big|_{x=0} = -D_A \frac{dC_{A,f}}{dx}\bigg|_{x=0} \tag{21}$$

$$N_A\big|_{x=x_L} = -D_A \frac{dC_{A,f}}{dx}\bigg|_{x=x_L} \tag{22}$$

$$N_B\big|_{x=x_L} = -D_B \frac{dC_{B,f}}{dx}\bigg|_{x=x_L} \tag{23}$$

Once those fluxes have been calculated, the left-hand sides of equations (4), (5), (6) and (20) can be evaluated. In this way, everything that is needed to solve equations (4), (5), (6) and (20) numerically and obtain the values of $\dot{n}_{A,g}$, $\dot{n}_{A,l}$, $\dot{n}_{B,l}$ and $P_{A,i}$ is available. The problem asks for the conversion of A which can be computed using equation (24).

$$f_A = \frac{\dot{n}_{A,g}^0 - \dot{n}_{A,g} - \dot{n}_{A,l}}{\dot{n}_{A,g}^0} \tag{24}$$

Upon performing the calculations as described, one finds the conversion to be 80%. If the liquid film had been ignored (the *bulk* liquid concentration of A assumed to be in equilibrium with the partial pressure of A at the interface) the conversion would incorrectly be found to equal 99%. It should be noted that this problem makes a number of simplifying assumptions that probably would not be valid in the analysis of a real reactor system. The purpose of the problem is to illustrate the coupling between the bulk gas, liquid film and bulk liquid balance equations and the mathematics needed to solve them.

**Calculation Details Using MATLAB**

The code for performing the calculations for this example was divided into three MATLAB functions. The first, Example_39_1.m stores the quantities specified in the problem statement, along with other constants that can be calculated from them, in a structure array named `given` (see the calculation details for Example 38.1 for a discussion of structure arrays). It then makes a guess for the four unknowns in equations (4), (5), (6) and (20) and calls a second MATLAB function, Example_39_1_bulk.m which solves those equations. Once the equations have been solved, it concludes by calculation of the conversion. The code is straightforward, as can be seen in Listing 1, and won't be discussed further.

```matlab
function Example_39_1
%Example_39_1 Code used in the solution of Example 39.1 of AFCoKaRE

    % Known quantities and constants entered in a structure array
    given = struct;
    given.yA0 = 0.15;
    given.VFRg0 = 2.0E-3; % m^3/min
    given.CBl0 = 3.0*1000; % mol/m^3
    given.VFRl = 0.06E-3; % m^3/min
    given.T = 20+273.15; % K
    given.P = 5.0; % atm
    given.Vl = 1.0E-3; % m^3
    given.SV = 445.; % /m
    given.kGP = 0.83*60; % mol/m^2/atm/min
    given.kL = 0.053/100*60; % m/min
    given.hA = 50.0E-3; % atm m^3/mol
    given.DA = 2.1E-5/1.0E4*60; % m^2/min
    given.DB = 7.25E-6/1.0E4*60; % m^2/min
    given.k = 6.05E2/1000; % m^3/mol/min
    given.R = 0.08206/1000; % m^3atm/mol/K
    given.xL = given.DA/given.kL; % eq 3
    given.nAg0 = given.VFRg0*given.P/given.R/given.T*given.yA0; % eq 7
    given.nBl0 = given.VFRl*given.CBl0; % eq 8

    % Solve the bulk liquid and gas mole balances
    guess = [
        0.1*given.nAg0 % nAg guess
        0.001*given.nBl0 % nAl guess
        given.nBl0/3. % nBl guess
        given.P*given.yA0/2. % PAi guess
        ];
    [nAg,nAl,nBl,PAi] = Example_39_1_bulk(given,guess);

    % Calculate and report the conversion
    pct_conv = 100*(given.nAg0 - nAg - nAl)/given.nAg0 % eq 24

end % of Example_39_1
```

*Listing 1. MATLAB function Example_39_1*

Equations (4), (5), (6) and (20) are non-linear algebraic equations, so they can be solved numerically by modifying the template file SolvNonDif.m as described in Supplemental Unit S2. By this point in the course, the use of this template file to solve algebraic equations should be quite familiar, so the description of the modifications will be succinct. The only difference here is that the internal function that evaluates the equations must call a boundary value ODE solver. A copy of SolvNonDif.m was saved as Example_39_1_bulk.m. The long initial comment was replace with a short one describing the purpose of the function and the function declaration was changed so that the structure array, $\texttt{given}$, and a vector named $\texttt{guess}$ (containing guesses for the value of the unknowns) are passed in as arguments and the final values of the unknowns are returned. Next, the internal function, $\texttt{evalEqns}$, was modified to return the values of the left-hand sides of equations (4), (5), (6) and (20). Specifically, the values of the four unknowns provided to $\texttt{evalEqns}$ were extracted and given meaningful names. The total outlet molar gas flow rate, outlet volumetric gas flow rate, bulk partial pressure of A, liquid phase concentrations of A and B and the rate were then calculated using equations (2) and (10) through (13).

A third MATLAB function, Example_39_1_film.m, was then called to solve the liquid film mole balances and calculate the fluxes of A and B at $x = 0$ and $x = x_L$. That function is described below. Once the fluxes have been calculated, the left hand sides of equations (4), (5), (6) and (20) are evaluated and stored in the vector $f$ to be returned by $\texttt{evalEqns}$, and that completes the modification of the internal function $\texttt{evalEqns}$. The final modification to the template file SolvNonDif.m involved extracting the values of the unknowns from the solution vector, $z$, so that they would be returned. All these modifications are shown in Listing 2.

```matlab
function [nAg,nAl,nBl,PAi] = Example_39_1_bulk(given,guess)
%Example_39_1_bulk Solve the bulk phase mole balances in Example 39.1 of
AFCoKaRE

    % Function that evaluates the bulk mole balance equations
    function f = evalEqns(z)
        % Extract the unknowns
        nag = z(1);
        nal = z(2);
        nbl = z(3);
        pai = z(4);

        % Calculate quantities appearing in the bulk mole balance equations
        ntotalg = given.P*given.VFRg0/given.R/given.T - given.nAg0 + nag; %
eq 10

        VFRg = ntotalg*given.R*given.T/given.P; % eq 11
        PAg = nag/ntotalg*given.P;
        CAl = nal/given.VFRl; % eq 12
        CBl = nbl/given.VFRl; % eq 13
        rate = given.k*given.Vl*CAl*CBl;
        if (rate < 0)
            rate = 0.;
        end

        % Solve the liquid film mole balances
        [NA0, NAxL, NBxL] = Example_39_1_film(given,pai,CAl,CBl);

        % Evaluate the bulk mole balance equations
        f = [
            given.nAg0 - given.SV*given.Vl*NA0 - nag; % eq 4
            NAxL*given.SV*given.Vl - rate - nal; % eq 5
            given.nBl0 + NBxL*given.SV*given.Vl - rate - nbl; % eq 6
            given.kGP*(PAg - pai) - NA0; % eq 20
        ];
    end % of internal function evalEqns

    % Solve the bulk phase mole balance equations
    z = fsolve(@evalEqns, guess);
    nAg = z(1);
    nAl = z(2);
    nBl = z(3);
    PAi = z(4);

end % of Example_39_1_bulk.m
```

*Listing 2. MATLAB function Example_39_1_bulk*

The liquid film mole balances are second order boundary value ODEs, equations (14) and (17). The MATLAB solver that will be used, bvp4c, requires that these ODEs be converted into a set of first order ODEs. Letting $y_1$ represent $C_{A,f}$ and $y_3$ represent $C_{B,f}$ followed by defining $y_2$ according to equation (25)

gives one first order ODE. Substitution into equation (14) gives the second first order ODE, equation (26) where equations (25) and (26) are completely equivalent to equation (14). Similarly defining $y_4$ according to equation (27) and substituting into equation (17) gives two first order ODEs, equations (27) and (28), that are completely equivalent to equation (17). Substitution of the new variables into the boundary conditions, equations (15), (16), (18) and (19), results in expressions for the boundary conditions in terms of the new variables, equations (29) through (32).

$$\frac{dy_1}{dx} = y_2 \tag{25}$$

$$\frac{dy_2}{dx} = \frac{k}{D_A} y_1 y_3 \tag{26}$$

$$\frac{dy_3}{dx} = y_4 \tag{27}$$

$$\frac{dy_4}{dx} = \frac{k}{D_B} y_1 y_3 \tag{28}$$

$$y_1(0) = C_{A,i} = \frac{P_{A,i}}{h_A} \tag{29}$$

$$y_1(x_L) = C_{A,l} \tag{30}$$

$$y_4(0) = 0 \tag{31}$$

$$y_3(x_L) = C_{B,l} \tag{32}$$

Having re-written the liquid film mole balances in this form, they can be solved by modification of the MATLAB template file SolvBVDif.m as described in Supplemental Unit S6. A copy of SolvBVDif.m was saved as Example_39_1_film.m and modified accordingly. The initial comment was deleted and replaced with a succinct comment stating the purpose of the modified file. The function declaration was changed to match the filename. In order to solve equations (25) through (28), values for the interfacial partial pressure of A and the bulk liquid concentrations of A and B are needed along with constant quantities given in the problem statement. Therefore the function declaration was changed so that these quantities are passed into the function as arguments. Solving equations (25) through (28) will yield the values of

$C_{A,f}$, $C_{B,f}$, $\dfrac{dC_{A,f}}{dx}$ and $\dfrac{dC_{B,f}}{dx}$ at discrete values of $x$ (mesh points) between 0 and $x_L$. However, we are

interested in the flux of A at $x = 0$ and $x = x_L$ and the flux of B at $x = x_L$. Therefore the function declaration was additionally changed so that these quantities are returned.

The next things in the template file are definitions of two internal functions named bvodes and bvs. The first of these functions, bvodes, must be modified so that given values for the dependent ($y_1$ through

$y_4$) and independent ($x$) variables, it returns a vector containing the values of the right-hand sides of equations (25) through (28). In this case, everything needed in order to do that is available in the structure array, `given`. The internal function, `bvs`, is passed the values of the dependent variables at the two ends of the range of the independent variable. Hence, in the present case, `y_at_start` is a vector containing the values of $y_1$ through $y_4$ at $x = 0$ and `y_at_end` is a vector containing the values of $y_1$ through $y_4$ at $x = x_L$. This function must be modified so that it returns the corresponding error in each of the boundary conditions. In the present case, if the boundary condition in equation (29) is satisfied, $y_1(0) - C_{A,i}$ should equal zero. If that term does not equal zero, its value is the error in the first boundary condition. If the boundary condition in equation (30) is satisfied, $y_2(x_L) - C_{A,l}$ should equal zero, and if it does not, its value is the error in the second boundary condition. Similarly, $y_4(0)$ should equal zero and $y_3(x_L) - C_{B,l}$ should equal zero when the boundary conditions of equations (31) and (32) are satisfied. Thus, the internal function, `bvs`, is modified to return these values. All of the modification described to this point are shown in Listing 3.

```
function [NA0, NAxL, NBxL] = Example_39_1_film(given,PAi,CAl,CBl)
%Example_39_1_film Solve the liquid film mole balances in Example 39.1 of
AFCoKaRE

    % Function that evaluates the derivatives
    function dydx = bvodes(x,y)
        rate = given.k*y(1)*y(3);
        if (rate < 0)
            rate = 0;
        end
        dydx = [
            y(2)
            rate/given.DA
            y(4)
            rate/given.DB
        ];
    end % of internal function bvodes

    % Function that calculates the errors at the boundaries
    function res = bvs(y_at_start,y_at_end)
        res = [
            y_at_start(1) - PAi/given.hA
            y_at_end(1) - CAl
            y_at_start(4)
            y_at_end(3) - CBl
        ];

    end % of internal function bvs
```

Listing 3. Modifications to the internal functions bvodes and bvs.

The template file must next be modified to discretize the range of the independent variable by specifying its lower limit and its upper limit along with the number of elements to divide the range into. Here I arbitrarily chose to divide the range into 20 parts. Following that, a guess for each of the dependent variables must be provided. Even though the $x$ direction has been divided into 20 sections, we only need to provide a single guess for each of the dependent variables. I used the average of the interfacial and bulk concentrations as a guess for $C_{A,f}$; the change in $C_{A,f}$ across the film divided by the film thickness as a guess for $\dfrac{dC_{A,f}}{dx}$; one-half of the bulk concentration of B as a guess for $C_{B,f}$ and the bulk concentration of B divided by the film thickness as a guess for $\dfrac{dC_{B,f}}{dx}$.

The final modification appears after equations (25) through (28) have been solved. It simply involves using the results to calculate the fluxes in equations (21) through (23) so that their values will be returned. All of these modifications are shown in Listing 4. At that point, the calculations can be performed by simply typing "Example_39_1" at the MATLAB command prompt (assuming all three files are in the current working directory or the MATLAB search path). Doing so generates the output shown in Listing 5.

```matlab
% Set up the initial mesh
x_range_low = 0.0;
x_range_high = given.xL;
n_mesh_points = 20;

% The next line creates an array for the independent variable
x = linspace(x_range_low,x_range_high,n_mesh_points);

% Guesses
yinit = [
    (PAi/given.hA + CAl)/2
    (PAi/given.hA - CAl)/given.xL
    CBl/2
    -CBl/given.xL
];

% Create a structure containing the mesh and guesses
solinit=bvpinit(x,yinit);

% Solve the odes
result = bvp4c(@bvodes,@bvs,solinit);

% Calculate any other desired quantities from the results
i_last_mesh = length(result.x);
NA0 = -given.DA*result.y(2,1);
NAxL = -given.DA*result.y(2,i_last_mesh);
NBxL = -given.DB*result.y(4,i_last_mesh);

end % of file Example_39_1_film.m
```

*Listing 4. Modifications to set up the mesh, provide guesses and calculate the desired fluxes.*

```
>> Example_39_1

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


pct_conv =

    79.9336
```

*Listing 5. Results generated upon execution of the function.*