Example Homework Assignment

Matrim Cauthon - ASUID#1000000

January 5, 2019

Contents

1 Problem 20.2-1

Solute A is diffusing at unsteady state into a semi-infinite medium of pure B and undergoes a first-order reaction with B. Solute A is dilute. Calculate the concentration c_A at points z = 0, 4, and 10 mm from the surface for $t = 1 \times 10^5$ s. Physical property data are $D_{AB} = 1 \times 10^{-9}$ m^2/s, $k' = 1 \times 10^{-4}$ s^{-1}, $c_{A0} = 1.0$ kg mol/m^3. Also calculate the kg mol absorbed/m^2. The concentration of A at point z is given by equation (1) (corrected version),

$$\begin{aligned}\frac{C_A}{C_{A0}} =& \frac{1}{2}\exp\left(-z\sqrt{k'/D_{AB}}\right) * \operatorname{erfc}\left(\frac{z}{2\sqrt{tD_{AB}}} - \sqrt{k't}\right) \\ &+ \frac{1}{2}\exp\left(z\sqrt{k'/D_{AB}}\right) * \operatorname{erfc}\left(\frac{z}{2\sqrt{tD_{AB}}} + \sqrt{k't}\right)\end{aligned} \tag{1}$$

Equation (1) was incorrect in the textbook, and has been modified according to the instructor's email.

```
u = symunit;
t = 1e5 * u.s;
D_AB = 1e-9 * u.m^2 / u.s;
k_prime = 1e-4 * u.s^-1;
```

```
C_A0 = 1.0 * (u.kg * u.mol) / (u.m^3);
C_A = @(z) C_A0 * ...
    (0.5*exp(-z*sqrt(k_prime/D_AB))*erfc((z/(2*sqrt(t*D_AB))) - sqrt(k_prime*t)) + ...
    0.5*(exp(z*sqrt(k_prime/D_AB)))*erfc((z/(2*sqrt(t*D_AB))) + sqrt(k_prime*t)));
disp(unitString(C_A(0), 'C_A (0 mm)'))
disp(unitString(C_A(0.004 * u.m), 'C_A (4 mm)'))
disp(unitString(C_A(0.010 * u.m), 'C_A (10 mm)'))
```

```
C_A (0 mm): 1 (kg*mol)/m^3
C_A (4 mm): 0.28226 (kg*mol)/m^3
C_A (10 mm): 0.042328 (kg*mol)/m^3
```

The amount of A absorbed per square meter is given by equation (2) (corrected version),

$$Q = C_{A0}\sqrt{D_{AB}/k'}\left[\left(k't + \frac{1}{2}\right)\mathrm{erf}\sqrt{k't} + \sqrt{k't/\pi}e^{-k't}\right] \tag{2}$$

```
Q = C_A0*sqrt(D_AB/k_prime)*((k_prime*t+0.5)*erf(sqrt(k_prime*t) + ...
    sqrt((k_prime*t)/pi)*exp(-k_prime*t)));
disp(unitString(Q))
```

```
Q: 0.033204 (kg*mol)/m^2
```

2 Problem 21.1-2

Prove or show the following relationships, starting with the flux equations:

2.1 Part a

Convert k_c' to k_y and k_G.

The flux equations involving these coefficients are,

$$N_A = k_c(C_{A1} - C_{A2}) = k_G(p_{A1} - p_{A2}) = k_y(y_{A1} - y_{A2})$$

the conversion between k_c' and k_c is given in the flux equation for A diffusing through stagnant, non-diffusing B,

$$N_A = \frac{k_c'}{x_{BM}}(C_{A1} - C_{A2}) = k_c(C_{A1} - C_{A2})$$
$$k_c = \frac{k_c'}{x_{BM}} = \frac{k_c' P}{p_{BM}}$$
$$k_y = \frac{k_c' P}{p_{BM}} \cdot \frac{C_{A1} - C_{A2}}{y_{A1} - y_{A2}} = \frac{k_c' P}{RT \cdot y_{BM}}$$
$$k_G = \frac{k_c' P}{p_{BM}} \cdot \frac{C_{A1} - C_{A2}}{p_{A1} - p_{A2}} = \frac{k_c' P}{RT \cdot p_{BM}}$$

Published with Matlab 9.4.0.813654 (R2018a) and AutoReport

2.2 Part b

Convert k_L to k_x and k'_x.

The flux equations involving these coefficients are,

$$N_A = k_L(c_{A1} - c_{A2}) = k_x(x_{A1} - x_{A2})$$

A relationship for k_x can be found by substituting $x = c_A/c$,

$$k_x = k_L \frac{c_{A1} - c_{A2}}{x_{A1} - x_{A2}} = k_L \frac{c_{A1} - c_{A2}}{\frac{c_{A1}}{c} - \frac{c_{A2}}{c}} = k_L \cdot c$$

$$k'_x = k_x \cdot x_{BM} = k_L \cdot c \cdot x_{BM}$$

2.3 Part c

Convert k_G to k_y and k_c.

The flux equations involving these coefficients are,

$$N_A = k_G(p_{A1} - p_{A2}) = k_y(y_{A1} - y_{A2}) = k_c(C_{A1} - C_{A2})$$

By substituting $p_A = y_A P$,

$$k_y = k_G \frac{p_{A1} - p_{A2}}{y_{A1} - y_{A2}} = k_G \frac{y_{A1}P - y_{A2}P}{y_{A1} - y_{A2}} = k_G \cdot P$$

Finally, substituting $C = P/RT$,

$$k_c = k_G \frac{p_{A1} - p_{A2}}{C_{A1} - C_{A2}} = k_G \frac{p_{A1} - p_{A2}}{(p_{A1} - p_{A2})/RT} = k_G \cdot RT$$

3 Problem 21.1-3

In a wetted-wall tower an air- H_2S mixture is flowing by a film of water that is flowing as a thin film down a vertical plate. The H_2S is being absorbed from the air to the water at a total pressure of 1.50 atm abs and 30 C. A value for k'_c of 9.567×10^{-4} m/s has been predicted for the gas-phase mass-transfer coefficient. At a given point, the mole fraction of H_2S in the liquid at the liquid-gas interface is $2.0(10^{-5})$ and p_A of H_2S in the gas is 0.05 atm. The Henry's law equilibrium relation is $p_A(\text{atm}) = 609 x_A$ (mole fraction in liquid). Calculate the rate of absorption of H_2S. (Hint: Call point 1 the interface and point 2 the gas phase. Then, calculate p_{A1} from Henry's law and the given x_A. The value of p_{A2} is 0.05 atm.)

The mass transfer coefficient given is k'_c, which can be converted to k_G for a flux equation based on partial pressures.

$$k_G = \frac{k'_c P}{RT \cdot p_{BM}}$$

$$N_A = k_G(p_{A1} - p_{A2})$$

```
u = symunit;
P = 1.5 * u.atm;
T = rewrite(30 * u.Celsius, u.K, 'Temperature', 'absolute');
k_c_prime = 9.567e-4 * u.m / u.s;
x_A1 = 2.0e-5;
p_A1 = 609 * x_A1 * u.atm;
p_A2 = 0.05 * u.atm;
p1 = separateUnits(p_A1); p2 = separateUnits(p_A2); p = separateUnits(P);
p_BM = ((p-p1)-(p-p2))/log((p-p1)/(p-p2)) * u.atm;
R = 8.2057338e-5 * (u.m^3 * u.atm) / (u.mol * u.K);
k_G = (k_c_prime * P) / (R * T * p_BM);
N_A = k_G * (p_A1 - p_A2) * 1e-3 * u.kg;  % convert to kg-mol
disp(unitString(N_A))
```

```
N_A: -1.4854e-06 (kg*mol)/(m^2*s)
```

4 Problem 21.2-1

A fluid is flowing in a vertical pipe and mass tranfer is occurring from the pipe wall to the fluid. Relate the convective mass-transfer coefficient k_c' to the variables D, ρ, μ, v, D_{AB}, g, and $\Delta\rho$, where D is pipe diameter, L is pipe length, and $\Delta\rho$ is the density difference.

According to Buckingham's pi theorem, given nine independent variables in three physical dimensions, there are six $\pi-$ groups to construct.

Vector order: [Length Mass time]

```
D = [1 0 0];
L = [1 0 0];
rho = [-3 1 0];
mu = [-1 1 -1];
v = [1 0 -1];
D_AB = [2 0 -1];
g = [1 0 -2];
delta_rho = [-3 1 0];
k_c_prime = [1 0 -1];

pi_1 = ([D' rho' mu']\-k_c_prime')'
```

```
pi_1 =

     1     1    -1
```

$$\pi_1 = \frac{D\rho k_c'}{\mu}$$

```
pi_2 = ([D' rho' mu']\-v')'
```

```
pi_2 =

     1     1    -1
```

$\pi_2 = \frac{D\rho v}{\mu}$

```
pi_3 = ([D' rho' mu']\-D_AB')'
```

```
pi_3 =

     0     1    -1
```

$\pi_3 = \frac{\rho D_{AB}}{\mu}$

```
pi_4 = ([D' rho' mu']\-g')'
```

```
pi_4 =

     3     2    -2
```

$\pi_4 = \frac{D^3\rho^2 g}{\mu^2}$

```
pi_5 = ([D' rho' mu']\-delta_rho')'
```

```
pi_5 =

     0    -1     0
```

$\pi_5 = \frac{\Delta\rho}{\rho}$

```
pi_6 = ([D' rho' mu']\-L')'
```

```
pi_6 =

    -1     0     0
```

$\pi_6 = \frac{L}{D}$

These dimensionless groups can be combined to describe the system. This combination was done pencil-and-paper with a lot of frustration, and the results are shown here.

$$\frac{\pi_1}{\pi_3} = f\left(\pi_4\pi_5\pi_6^3, \pi_2, \pi_3^{-1}\right)$$
$$\frac{k_c' D}{D_{AB}} = f\left(\frac{gL^3\rho\Delta\rho}{\mu^2}, \frac{Dv\rho}{\mu}, \frac{\mu}{\rho D_{AB}}\right)$$

5 Problem 21.3-1

Using the data and physical properties of Example 21.3-2, calculate the flux for a water velocity of 0.152 m/s and a plate length of $L = 0.137$ m. Do not assume that $x_{BM} = 1.0$ but actually calculate its value.

The Schmidt and Reynolds numbers can be calculated from the given quantities,

```
u = symunit;
T = rewrite(26.1 * u.Celsius, u.K, 'Temperature', 'absolute');
L = 0.137 * u.m;
v = 0.152 * u.m / u.s;
solubility = 0.02948 * (u.kg * u.mol) / (u.m^3);
D_AB = 1.245e-9 * u.m^2 / u.s;
mu = 8.71e-4 * u.Pa * u.s;
rho = 996 * u.kg / u.m^3;
N_Sc = simplify(mu / (rho * D_AB));
N_Re = simplify((L * v * rho)/mu);
disp(unitString(N_Sc))
disp(unitString(N_Re))
```

```
N_Sc: 702.408 1
N_Re: 23812.5189 1
```

This value for the Reynolds number corresponds to this equation for mass flux,

$$J_D = 0.99 N_{Re,L}^{-0.5} = \frac{k_c'}{v} \left(N_{Sc}\right)^{2/3}$$

```
syms k_c_prime;
k_c_prime = solve(0.99*N_Re^-0.5 == (k_c_prime/v)*(N_Sc^(2/3)));
FW_water = 18.02* u.kg / (u.kg * u.mol);
c = rho / FW_water;
x_A1 = 0;
x_A2 = solubility / (solubility + c);
```

```
x_BM = (1 - (1 - x_A2))/log(1/(1 - x_A2));
k_c = k_c_prime / x_BM;
N_A = k_c * solubility;
disp(unitString(N_A))
```

```
N_A: 3.6391e-07 (kg*mol)/(m^2*s)
```

6 Referenced Functions

6.1 unitString.m

```
 function displayString = unitString(quantity, name)
%UNITSTRING Display a 1x2 sym with symbolic units
% USAGE: unitString(some_quantity, name)
% OUTPUT:
%   - displayString: char vector containing name, scalar, and units
if nargin < 2
    n = inputname(1);
else
    n = name;
end

[s, U] = separateUnits(quantity);
formatSpec = '%s: %s %s';
displayString = sprintf(formatSpec, n, num2str(double(s)), symunit2str(U));
end
```