

Process System Engineering #2

#03150796 Amane Suzuki

1 Theory

Given reaction rate constant k as a function of temperature. Constant temperature condition therefore k is common in all reactor.

mole balance on reactor 1, with flow rate v , reactor size V , and monomer concentration C_n ,

$$C_0 v - k C_1 V = C_1 v, \quad (1)$$

$$\frac{C_1}{C_0} = \frac{v}{v + kV}, \quad (2)$$

mole balance on reactor 2,

$$C_1 v - k C_2 V = C_2 v, \quad (3)$$

$$\frac{C_2}{C_0} = \frac{C_2}{C_1} \frac{C_1}{C_0} = \left(\frac{v}{v + kV} \right)^2, \quad (4)$$

therefore,

$$\frac{C_n}{C_0} = \left(\frac{v}{v + kV} \right)^n. \quad (5)$$

In terms of weight fraction of monomer γ_n ,

$$\frac{\gamma_n}{\gamma_0} = \left(\frac{v}{v + kV} \right)^n, \quad (6)$$

moreover,

$$1 - \zeta = \frac{\gamma_N}{\gamma_0} = \left(\frac{v}{v + kV} \right)^N, \quad (7)$$

$$\frac{v}{v + kV} = (1 - \zeta)^{\frac{1}{N}}, \quad (8)$$

$$(9)$$

therefore,

$$\gamma_n = (1 - \zeta)^{\frac{n}{N}} \gamma_0. \quad (10)$$

2 Algorithm

Power consumption P of each reactor is defined by γ_{in} , γ_{out} , and invariable conditions (e.g. ρ , ΔH , etc.)

therefore the structure of the source program is,

```
class Plant
  initialize(N, gamma_0, T2)
  calc()
  reactor(gamma_in, gamma_out)
end

plant = Plant.new(N, gamma_0, T2)
plant.calc()
```

In `initialize()` function, we setup

- viscosity μ
- flow rate v
- reactor size V
- diameter of reactor D
- height of reactor H

based on the method of Home Task #1.
Given reactor size V is calculated by,

$$V = \frac{v\tau}{N}, \quad (11)$$

In `reactor()` function, we calculate

- heat transfer rate h
- dimensionless numbers Pr, Nu, Re
- revolution number n
- power consumption P

In `calc()` function,

1. calculate γ_n by equation (8)
2. call `reactor`(γ_{n-1}, γ_n) from $n = 1$ to $n = N$
3. output total power consumption P_{tot}

3 Result

(1) $N = 1, \gamma_0 = 0.05$

Use listing 1,

```
$ ruby plant.rb
input n[-], gamma_0[wt%], T2[K]
1
0.05
258
Conditions:
(N, gamma_0, T2)=(1, 0.05, 258)
Reactor Size
V = 514.706[m3]
D = 7.959[m]
H = 10.346[m]
Result:
#1
Re = 6893.441
n = 0.316[rps]
P = 32745.985[W]
Total:
Ptot = 32745.985[W]
```

volume	514.706 m ³
diameter	7.959 m
height	10.346 m
total power consumption	32,745.985 W

Table 1: Result in $(N, \gamma_0, T_2) = (1, 0.05, 258)$

(2) $N = 3, \gamma_0 = 0.04$

Use listing 1,

```
$ ruby reactor.rb
input n[-], gamma_0[wt%], T2[K]
3
0.04
258
Conditions:
(N, gamma_0, T2)=(3, 0.04, 258)
Reactor Size:
V = 214.461[m3]
D = 5.944[m]
H = 7.728[m]
Results:
#1
Re = 3446.108
n = 0.229[rps]
P = 3542.243[W]
#2
Re = 2179.51
n = 0.145[rps]
P = 1018.779[W]
#3
Re = 1378.443
n = 0.092[rps]
P = 293.009[W]
Total:
Ptot = 4854.031[W]
```

volume	214.461 m ³
diameter	5.944 m
height	7.728 m
total power consumption	4854.031 W

Table 2: Result in $(N, \gamma_0, T_2) = (3, 0.04, 258)$

(3) $N = 1$ to 5 , $\gamma_0 = 0.02$ to 0.10

Use listing 2,

```
$ ruby plant-advanced.rb
2.47  1.25 0.75 0.52 0.38
6.92  3.5  2.11 1.45 1.07
15.9  8.04 4.85 3.33 2.46
32.75 16.55 9.99 6.85 5.07
62.92 31.8 19.21 13.16 9.75
115.28 58.26 35.19 24.11 17.86
203.95 103.08 62.25 42.65 31.61
351.32 177.56 107.23 73.47 54.45
592.59 299.51 180.87 123.93 91.84
```

γ_0	N				
	1	2	3	4	5
0.02	2.47	1.25	0.75	0.52	0.38
0.03	6.92	3.5	2.11	1.45	1.07
0.04	15.9	8.04	4.85	3.33	2.46
0.05	32.75	16.55	9.99	6.85	5.07
0.06	62.92	31.8	19.21	13.16	9.75
0.07	115.28	58.26	35.19	24.11	17.86
0.08	203.95	103.08	62.25	42.65	31.61
0.09	351.32	177.56	107.23	73.47	54.45
0.10	592.59	299.51	180.87	123.93	91.84

Table 3: Power Consumption P [kW] in each condition

4 Source Program

Listing 1: plant.rb

```
1 include Math
2
3 GP = (63.0*1000/24) # product flow rate [kg h-1]
4 ML = 50.0 # polymer length
5 TAU = 5.0 # residence time
6 RHO = 850.0 # density [kg m-3]
7 ALPHA = 1.3 # height/diameter of reactor
8 HP = (72.8*1000) # heat of polymerization [J mol-1]
9 CV = 0.6 # conversion
10 T1 = (273+50) # [K]
11 TC = 0.128 # thermal conductivity [W m-1 K-1]
12 CP = (1.68*1000) # specific heat of toluene [J kg-1 K-1]
13 M = 54.0 # molecular weight of butadiene [g mol-1]
14
15 class Plant
16   def initialize(n, feed, coolant)
17     @n = n # number of reactors [-]
18     @feed = feed # feed [wt%]
19     @visc = ((ML)**1.7)*(CV**2.5)*exp(21.0*feed)*1e-3 # viscosity [
20       Pa s]
21     @rate = GP/(RHO*CV*feed) # feed speed [m3 h-1]
22     @volume = @rate*TAU/n # [m3]
23     @diameter = (@volume/(ALPHA*2*PI))**(1/3.0)*2 # [m]
24     @height = @diameter*ALPHA # [m]
25     @coolant = coolant # T2 [K]
26   end
27   def calc()
28     tpc = 0 # total power consumption [W]
29     prop = (1-CV)**(1.0/@n) # proportional constant [-]
30
31     # show conditions
32     puts "Conditions:"
33     puts "(N,γ0,T2)="+
34       "("+@n.to_s+",γ0"+"@feed.to_s+",γ0"+"@coolant.to_s+")"
35
36     # show reactor size
37     puts "Reactor Size:"
38     puts "V0"+"@volume.round(3).to_s+"[m3]"
39     puts "D0"+"@diameter.round(3).to_s+"[m]"
40     puts "H0"+"@height.round(3).to_s+"[m]"
41
42     # show result of each reactor
43     puts "Results:"
44     for n in 1..@n
45       puts "#"+n.to_s
46       tpc += reactor(@feed*(prop**(n-1)),@feed*(prop**n))
47     end
48
49     # show total power consumption
```

```

50     puts "Total:"
51     puts "Ptot_=" + tpc.round(3).to_s + " [W] "
52 end
53
54 def reactor(gamma_in, gamma_out)
55     # heat transfer rate
56     h = HP*(@rate*RHO*(gamma_in-gamma_out)*1000/3600)/M/(@diameter*
        PI*@height)/(T1-@coolant)
57
58     # dimensionless numbers
59     pr = @visc*CP/TC
60     nu = h*@diameter/TC
61     re = (2*nu/pr**(1/3.0))*1.5
62     puts "Re_=" + re.round(3).to_s
63
64     # revolution number
65     revnum = re*@visc/RHO/(@diameter/2)**2
66     puts "n_=" + revnum.round(3).to_s + " [rps] "
67
68     # power consumption
69     np = 14.6*re**(-0.28)
70     p = np*RHO*(revnum**3)*(@diameter/2)**5
71     puts "P_=" + p.round(3).to_s + " [W] "
72
73     return p
74 end
75 end
76
77 puts "input_n[-],_gamma_0[wt%],_T2[K] "
78 n = gets.to_i
79 feed = gets.to_f
80 coolant = gets.to_i
81
82 plant = Plant.new(n, feed, coolant)
83 plant.calc()

```

Listing 2: plant-advanced.rb

```

1 include Math
2
3 GP = (63.0*1000/24) # product flow rate [kg h-1]
4 ML = 50.0 # polymer length
5 TAU = 5.0 # residence time
6 RHO = 850.0 # density [kg m-3]
7 ALPHA = 1.3 # height/diameter of reactor
8 HP = (72.8*1000) # heat of polymerization [J mol-1]
9 CV = 0.6 # conversion
10 T1 = (273+50) # [K]
11 TC = 0.128 # thermal conductivity [W m-1 K-1]
12 CP = (1.68*1000) # specific heat of toluene [J kg-1 K-1]
13 M = 54.0 # molecular weight of butadiene [g mol-1]
14
15 class Plant
16   def initialize(n, feed, coolant)
17     @n = n # number of reactors [-]
18     @feed = feed # feed [wt%]
19     @visc = ((ML)**1.7)*(CV**2.5)*exp(21.0*feed)*1e-3 # viscosity [
20       Pa s]
21     @rate = GP/(RHO*CV*feed) # flow rate [m3 h-1]
22     @volume = @rate*TAU/n # [m3]
23     @diameter = (@volume/(ALPHA*2*PI))**(1/3.0)*2 # [m]
24     @height = @diameter*ALPHA # [m]
25     @coolant = coolant # T2 [K]
26   end
27
28   def calc()
29     tpc = 0 # total power consumption [W]
30     prop = (1-CV)**(1.0/@n) # proportional constant [-]
31
32     for n in 1..@n
33       tpc += reactor(@feed*(prop**(n-1)),@feed*(prop**n))
34     end
35
36     return ((tpc)/1000).round(2)
37   end
38
39   def reactor(gamma_in, gamma_out)
40     # heat transfer rate
41     h = HP*(@rate*RHO*(gamma_in-gamma_out)*1000/3600)/M/(@diameter*
42       PI*@height)/(T1-@coolant)
43
44     # dimensionless numbers
45     pr = @visc*CP/TC
46     nu = h*@diameter/TC
47     re = (2*nu/pr**(1/3.0))**1.5
48
49     # revolution number
50     revnum = re*@visc/RHO/(@diameter/2)**2
51
52     # power consumption
53     np = 14.6*re**(-0.28)

```

```
52     p = np*RH0*(revnum**3)*(@diameter/2)**5
53
54     return p
55 end
56 end
57
58 coolant = 258
59 for feed in 2..10
60     f = feed*0.01
61     for n in 1..5
62         plant = Plant.new(n, f, coolant)
63         print plant.calc().to_s + "\t"
64     end
65     print "\n"
66 end
```
