

Scilab Textbook Companion for
Diffusion: Mass Transfer In Fluid Systems
by E. L. Cussler¹

Created by
Abhinay Korukonda
Chemical Engineering
Chemical Engineering
IIT Bombay
College Teacher
Na

Cross-Checked by
Mukul Kulkarni

May 24, 2016

¹Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website <http://scilab.in>

Book Description

Title: Diffusion: Mass Transfer In Fluid Systems

Author: E. L. Cussler

Publisher: Cambridge University Press

Edition: 2

Year: 1997

ISBN: 0521564778

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes	4
3 Diffusion in concentrated solutions	5
4 Dispersion	7
5 Values of Diffusion Coefficients	9
6 Diffusion of Interacting Species	15
8 Fundamentals of Mass Transfer	20
9 Theories of Mass Transfer	28
10 Absorption	32
11 Mass Transfer in Biology and Medicine	37
12 Differential Distillation	40
13 Staged Distillation	43
14 Extraction	47
15 Adsorption	50
16 General Questions and Heterogenous Chemical Reactions	53
17 Homogenous Chemical Reactions	55

18 Membranes	59
19 Controlled Release and Phenomena	65
20 Heat Transfer	67
21 Simultaneous Heat and Mass Transfer	72

List of Scilab Codes

Exa 3.2.4	Average velocity	5
Exa 3.3.1	Error calculation	5
Exa 4.2.1	Dispersion coefficient	7
Exa 4.2.2	Percent of pipe	8
Exa 5.1.1	Diffusion Coefficient	9
Exa 5.1.2	Diffusion coefficient and error	9
Exa 5.1.3	Diffusion coefficient	10
Exa 5.2.1	Diffusion coefficient stokes	11
Exa 5.2.2	a b calculation	12
Exa 5.2.3	Diffusion coefficient	12
Exa 5.5.1	Diffusion coefficient	13
Exa 5.5.2	Diffusion coefficient	13
Exa 6.1.1	Diffusion coefficient and transference	15
Exa 6.1.2	Diffusion coefficient	15
Exa 6.1.5	Diffusion coefficient	16
Exa 6.2.1	Diffusion coefficient	16
Exa 6.4.1	Tortuosity calculation	17
Exa 6.4.2	Diffusion coefficient	18
Exa 6.4.3	Diffusion flux	18
Exa 6.4.4	Diffusion coefficient in gel	19
Exa 8.1.1	Time taken	20
Exa 8.1.2	Mass transfer coefficient	20
Exa 8.1.3	Mass transfer coefficient	21
Exa 8.1.4	Mass transfer coefficient	21
Exa 8.2.1	Mass transfer coefficient	22
Exa 8.2.2	Mass transfer coefficient liquid and gas	23
Exa 8.3.1	Column length	23
Exa 8.3.2	Mass flux in air and water	24

Exa 8.5.1	Overall Mass transfer coefficient	25
Exa 8.5.2	Overall Mass transfer coefficient	26
Exa 8.5.3	Overall Mass transfer coefficient	26
Exa 8.5.4	Overall Mass transfer coefficient	27
Exa 9.1.1	Film thickness	28
Exa 9.2.1	Contact time and residence time	28
Exa 9.3.1	Apparent Mass transfer coefficient	29
Exa 9.4.1	Average Mass transfer coefficient	30
Exa 9.4.2	Local Mtc	30
Exa 10.2.1	Diameter of tower	32
Exa 10.3.1	Length of tower	32
Exa 10.3.2	Percentage of oxygen	33
Exa 10.4.1	Length and diameter of tower	34
Exa 11.1.1	Mass transfer coefficient	37
Exa 11.2.1	Mass transfer coefficient	37
Exa 11.2.2	Percentage of toxins	38
Exa 12.2.1	Mass transfer	40
Exa 12.2.2	NTU	40
Exa 12.4.1	length of tower	41
Exa 13.1.1	Column diameter	43
Exa 13.2.1	Number of stages	44
Exa 13.2.2	Number of stages	44
Exa 13.4.1	Murphree efficiency	45
Exa 13.4.2	Murphree efficiency	46
Exa 14.3.1	Length required	47
Exa 14.4.1	No of stages required	48
Exa 14.5.1	No of stages including feed	48
Exa 15.3.2	Length of bed	50
Exa 15.3.3	volume of adsorbent	50
Exa 15.4.1	Breakthrough time	51
Exa 15.4.2	rate constant	52
Exa 16.3.2	rate constant	53
Exa 16.3.3	rate of reaction	54
Exa 17.1.1	Diffusion coefficient	55
Exa 17.1.2	Reduction in reaction rate	55
Exa 17.1.3	rate constant	56
Exa 17.2.1	Increase in rate	56
Exa 17.4.1	Rate constant	57

Exa 17.5.1 relaxation time	58
Exa 18.1.1 Rapidness calculation	59
Exa 18.2.1 Permeability calculation	59
Exa 18.2.2 Time spent	60
Exa 18.3.1 Osmotic pressure difference	61
Exa 18.3.2 Transport coefficient	62
Exa 18.4.1 Membrane selectivity	62
Exa 18.5.2 total flux	63
Exa 19.1.1 Permeability	65
Exa 19.2.1 Membrane area calculation	66
Exa 20.1.1 Thermal diffusivity	67
Exa 20.3.1 Overall heat transfer coefficient	67
Exa 20.3.2 Time taken	68
Exa 20.3.3 Savings calculation	69
Exa 20.4.1 Overall heat transfer coefficient	69
Exa 20.4.2 heat transfer coefficient	70
Exa 21.1.2 Thermal diffusivity	72
Exa 21.3.1 Time taken	73
Exa 21.3.2 Mass transfer and time taken	73
Exa 21.4.1 Flow rate area and length	74
Exa 21.5.1 Time taken	75

Chapter 3

Diffusion in concentrated solutions

Scilab code Exa 3.2.4 Average velocity

```
1
2 clc
3 D = 0.1 // cm^2/sec
4 l = 10 // cm
5 C10 = 1
6 C11 = 0
7 C1 = 0.5
8 V1 = (D/l)*(C10 - C11)/C1 // Cm/sec
9 V2 = -V1
10 M1 = 28
11 M2 = 2
12 omeg1 = C1*M1/(C1*M1 + C1*M2)
13 omeg2 = C1*M2/(C1*M1 + C1*M2)
14 V = omeg1*V1 + omeg2*V2
15 printf("The mass average velocity is %.5f cm/s",V)
```

Scilab code Exa 3.3.1 Error calculation

```
1
2 clc
3 //initialization of variables
4 // At 6 degree centigrade
5 p1sat = 37 // Vapor pressure of benzene in mm Hg
6 p = 760 // atmospheric pressure in mm Hg
7 y11 = 0
8 y10 = p1sat/p
9 n1byDcby1 = log((1-y11)/(1-y10))// because flux n1 =
    D*c/l * ln(1-y11/1-y10)
10 n2byDcby1 = y10-y11 // Flux calculated assuming
    dilute solution as n1 = Dc/l*(y10-y11)
11 err1 = ((n1byDcby1-n2byDcby1)/n2byDcby1)*100 //
    Percentage error
12 printf("The error in measurement at 6 degree
    centigrade is %.1f percent",err1)
13 // At 60 degree centigrade
14 p1sat = 395 // Vapor pressure of benzene in mm Hg
15 p = 760 // atmospheric pressure in mm Hg
16 y11 = 0
17 y10 = p1sat/p
18 n1byDcby1 = log((1-y11)/(1-y10))// because flux n1 =
    D*c/l * ln(1-y11/1-y10)
19 n2byDcby1 = y10-y11 // Flux calculated assuming
    dilute solution as n1 = Dc/l*(y10-y11)
20 err1 = ((n1byDcby1-n2byDcby1)/n2byDcby1)*100 //
    Percentage error
21 printf("\\n The error in measurement at 60 degree
    centigrade is %.1f percent",err1)
```

Chapter 4

Dispersion

Scilab code Exa 4.2.1 Dispersion coefficient

```
1
2 clc
3 //initialization of variables
4 z = 80 // metres
5 c1 = 410 //ppm
6 c = 860 // ppm
7 d = 2 //km
8 v = 0.6 //km/hr
9 r = 3600 //sec/hr
10 //Calculations
11 t1 = (d/v)*r//sec
12 E = (-((z^2)/(4*t1))/(log(410/860)))*10^4 // cm^2/sec
    //answer in textbook is wrong
13 d2 = 15 //km
14 c2 = c*(sqrt(d/d2))//ppm
15 //Results
16 printf("The value of dispersion coefficient is %.f cm
    ^2/sec",E)
17 printf("\\n The value of maximum concentration at 15
    km downstream is %.f ppm",c2)
```

Scilab code Exa 4.2.2 Percent of pipe

```
1
2 clc
3 //initialization of variables
4 d = 10 //cm
5 s = 3 // km
6 v = 500 //cm/sec
7 nu = 0.15 // cm2/sec
8 //Calculations
9 E = 0.5*d*v // cm2/sec
10 c1 = 1000 // m/km
11 c2 = 1/100 // m/cm
12 z = sqrt(4*E*c1*c2*s/v)
13 percent = z*100/(s*c1)
14 //Results
15 printf(" The percent of pipe containing mixed gases
    is %.1f percent",percent)
```

Chapter 5

Values of Diffusion Coefficients

Scilab code Exa 5.1.1 Diffusion Coefficient

```
1
2 clc
3 //initialization of variables
4 m = 20/(6*10^23)//wt of each molecule
5 kb = 1.38*10^-16 // g-cm^2/sec-K
6 T = 298 // Kelvin
7 dou = 0.04*10^-7 // cm
8 //Calculations
9 v = sqrt(kb*T*2/m) //cm/sec
10 D = (dou*v/6)*10^5 // in x*10^-5 cm^2/sec
11 //Results
12 printf("The value of Diffusion co efficient is %.f
    x10^-5 cm^2/sec",D)
```

Scilab code Exa 5.1.2 Diffusion coefficient and error

```
1
2 clc
```

```

3 //Initialization of variables
4 sigma1 = 2.92 // angstroms
5 sigma2 = 3.68 // angstroms
6 sigma12 = (sigma1+sigma2)/2 // angstroms
7 T = 294 // Kelvin
8 M1 = 2.02 // Mol wt of hydrogen
9 V1 = 7.07
10 V2 = 17.9
11 M2 = 28 // Mol wt of Nitrogen
12 p = 2 //atm
13 Omega = 0.842
14 Dexp = 0.38 // cm^2/sec
15 //calculations
16 D1 = ((1.86*10^-3)*((T)^1.5)*(((1/M1)+(1/M2))^0.5))
      /((p)*((sigma12)^2)*Omega) //cm^2/sec
17 err1 = ((Dexp-D1)/Dexp)*100
18 D2 = ((10^-3)*((T)^1.75)*(((1/M1)+(1/M2))^0.5))/((p)
      *(((V1)^(1/3))+ ((V2)^(1/3)))^2)) //cm^2/sec
19 err2 = ((Dexp-D2)/Dexp)*100
20 //Results
21 printf("The diffusion co efficient using Chapman-
      enskong theory is %.2f cm^2/sec",D1)
22 printf("\nThe error for the above correlation is %.f
      percent",err1)
23 printf("\nThe diffusion co efficient using Fuller
      correlation is %.2f cm^2/sec",D2)
24 printf("\nThe error for the above correlation is %.f
      percent",err2)

```

Scilab code Exa 5.1.3 Diffusion coefficient

```

1
2
3 clc
4 //initialization of variables

```

```

5 p0 = 1//atm
6 p = 33 //atm
7 D0 = 0.043 // cm^2/sec
8 //Calculations
9 D = (p0*D0/p)*10^5 // x*10^-5 cm^2/sec
10 //Results
11 printf("The diffusion coefficient for the given
    conditions is %.3f x10^-5 cm^2/sec",D)
12 disp("The answer is a bit different due to rounding
    off error in textbook. Also please verify that
    10^-5 factor is utilized outside.")

```

Scilab code Exa 5.2.1 Diffusion coefficient stokes

```

1
2 clc
3 //Initialization of variables
4 R0 = 1.73*10^-8 //cm
5 kb = 1.38*10^-16 // g-cm^2/sec^2-K
6 T = 298 // kelvin
7 Mu = 0.01 // g/cm-sec
8 Mu2 = 1 // Centipoise
9 DE = 1.80//x*10^-5 cm^2/sec
10 phi = 2.6
11 VH20 = 18 // cc/g-mol
12 V02 = 25 // cc/g-mol
13 //calculations
14 D1 = ((kb*T)/(6*%pi*Mu*R0))*10^5// x*10^-5 cm^2/sec
15 err1 = (DE-D1)*100/DE // error percentage
16 D2 = (((8.2*10^-8)*T/(Mu2*((V02)^(1/3))))*(1+ ((3*
    VH20/V02)^(2/3))))*10^5 //x*10^-5 cm^2/sec
17 err2 = (D2-DE)*100/DE // Error percentage
18 D3 = (((7.4*10^-8)*((phi*VH20)^0.5)*T)/(Mu2*((V02)
    ^0.6)))*10^5//x*10^-5 cm^2/sec
19 err3 = (D3-DE)*100/DE// Error percentage

```

```

20 //Results
21 printf("The diffusion coefficient using Stokes
    Einstein correlation is %.1f x10-5 cm2/sec",D1)
22 printf("\nThe error regarding above correlation is %
    .1f percent low",err1)
23 printf("\nThe diffusion coefficient using Wilke-
    Chang correlation is %.1f x10-5 cm2/sec",D3)
24 printf("\nThe error regarding above correlation is %
    .1f percent high",err3)

```

Scilab code Exa 5.2.2 a b calculation

```

1
2 clc
3 //Initialization of variables
4 kb = 1.38*10-16//g-cm2-sec2-K
5 T = 310 // kelvin
6 k = 30 // which is a/b
7 D = 2.0*10-7 // cm2/sec
8 Mu = 0.00695 // g/cm-sec
9 //Calculations
10 a = ((kb*T/(6*pi*Mu*D))*((log(k + ((k2-1)(0.5)))))
    /(((1-(1/k2))0.5))*107 // nm
11 b = a/k // nm
12 //Results
13 printf("The results of a and b are %.f nm and %.1f
    nm",a,b)

```

Scilab code Exa 5.2.3 Diffusion coefficient

```

1
2 clc
3 //initialization of variables

```



```

4 D1 = 1.26*10^-5 // for x1=1 , D0 value in cm^2/sec
5 x1 = 0.5
6 D2 = 4.68*10^-5 // for x2=1 , D0 Value in cm^2/sec
7 x2 = 0.5
8 k = -0.69 // dlngamma1/dx1 value given
9 //Calculations
10 D0 = ((D1)^x1)*((D2)^x2)*10^5 // x*10^-5 cm^2/sec
11 D = D0*(1+k) // Diffusion coefficient in x*10^-5 cm
    ^2/sec
12 //Results
13 printf("The diffusion coefficient is %.2f x10^-5 cm
    ^2/sec",D)

```

Scilab code Exa 5.5.1 Diffusion coefficient

```

1
2 clc
3 //initialization of variables
4 m = 20/(6*10^23)//wt of each molecule
5 kb = 1.38*10^-16 // g-cm^2/sec-K
6 T = 298 // Kelvin
7 dou = 0.04*10^-7 // cm
8 //Calculations
9 v = sqrt(kb*T*2/m) //cm/sec
10 D = (dou*v/6)*10^5 // in x*10^-5 cm^2/sec
11 //Results
12 printf("The value of Diffusion coefficient is %.f
    x10^-5 cm^2/sec",D)

```

Scilab code Exa 5.5.2 Diffusion coefficient

```

1
2 clc

```

```
3 //Intialization of variables
4 sigmasquare = 0.014 // Slope of the graph
5 t = 150 // seconds
6 //Calculations
7 D = (sigmasquare/(2*t))*10^5 // in x*10^-5 cm^2/sec
8 //Results
9 printf("The value of diffusion co efficient is %.1f
    x10^-5 cm^2/sec",D)
```

Chapter 6

Diffusion of Interacting Species

Scilab code Exa 6.1.1 Diffusion coefficient and transference

```
1
2 clc
3 //Initialization of variables
4 DHplus = 9.31*10^-5 // cm^2/sec
5 DClminus = 2.03*10^-5 // cm^2/sec
6 //Calculations
7 DHC1 = (2/((1/DHplus)+(1/DClminus)))*10^5 // x*10^-5
      cm^2/sec
8 tHplus = DHplus/(DHplus+DClminus)
9 percentage = tHplus*100 // percent
10 //Results
11 printf("The diffusion co efficient of the solution
      is %.1f x10^-5 cm^2/sec",DHC1)
12 printf("\\n The transeferce for protons is %.f
      percent",percentage)
```

Scilab code Exa 6.1.2 Diffusion coefficient

```

1
2 clc
3 // Initialization of variables
4 z1 = 3
5 z2 = 1
6 D2 = 2.03*10^-5 // cm^2/sec
7 D1 = 0.62*10^-5 // cm^2/sec
8 // Calculations
9 D = ((z1+z2)/((z1/D2)+(z2/D1)))*10^5 // x*10^-5 cm^2/
    sec
10 // Results
11 printf("The diffusion coefficient is %.2f x10^-5 cm
    ^2/sec",D)

```

Scilab code Exa 6.1.5 Diffusion coefficient

```

1
2 clc
3 // Initialization of variables
4 zCa = 2
5 zCl = 1
6 DC1 = 2.03*10^-5 // cm^2/sec
7 DCa = 0.79*10^-5 // cm^2/sec
8 // Calculations
9 DCaCl2 = ((zCa+zCl)/((zCa/DC1)+(zCl/DCa)))*10^5 // x
    *10^-5 cm^2/sec
10 // Results
11 printf("The diffusion coefficient of CaCl2 is %.2f
    x10^-5 cm^2/sec",DCaCl2)

```

Scilab code Exa 6.2.1 Diffusion coefficient

```

1

```

```

2  clc
3  //initialization of variables
4  pKa = 4.756
5  DH = 9.31*10^-5 // cm^2/sec
6  DCH3COO = 1.09*10^-5 //cm^2/sec
7  D2 = 1.80*10^-5 //cm^2/sec
8  Ct = 10 // moles/lit
9  //Calculations
10 K = 10^pKa // litres/mol
11 D1 = 2/((1/DH)+(1/DCH3COO))
12 D = 2/((1/D1)+(1/D2))*10^5 // Diffusion co efficient
    in x*10^-5 cm^2/sec
13 //Results
14 printf("The diffusion coefficient of acetic acid in
    water is %.1f x10^-5 cm^2/sec",D)

```

Scilab code Exa 6.4.1 Tortuosity calculation

```

1
2  clc
3  //Initialization of variables
4  sigma1 = 4.23 // angstroms
5  sigma2 = 4.16 //Angstroms
6  sigma12 = (sigma1+sigma2)/2 // angstroms
7  T = 573 // Kelvin
8  M1 = 28
9  M2 = 26
10 p = 1 //atm
11 Omega = 0.99
12 Deff = 0.17 //cm^2/sec
13 //calculations
14 D = ((1.86*10^-3)*((T)^1.5)*(((1/M1)+(1/M2))^0.5))
    /((p)*((sigma12)^2)*Omega) //cm^2/sec
15 Tou = D/Deff
16 //Results

```

```
17 printf("The tortuosity is %.f",Tou)
```

Scilab code Exa 6.4.2 Diffusion coefficient

```
1
2 clc
3 //Initialization of variables
4 kb = 1.38*10^-16 // g-cm^2/sec^2-K
5 T = 310 //Kelvin
6 Mu = 0.01 // g/cm-sec
7 R0 = 2.5*10^-8 //cm
8 d = 30*10^-8 //cm
9 //Calculations
10 D = (kb*T/(6*%pi*Mu*R0))*(1+((9/8)*(2*R0/d)*(log(2*
    R0/d))))+((-1.54)*(2*R0/d)) //cm^2/sec
11 //Results
12 printf("The diffusion coefficient is %.8f cm^2/sec",
    D)
```

Scilab code Exa 6.4.3 Diffusion flux

```
1
2 clc
3 //Initialization of variables
4 kb = 1.38*10^-16 // g-cm^2/sec^2-K
5 T = 373 // K
6 T0 = 273 // K
7 sigma = 2.83*10^-8 // cm
8 p = 1.01*10^6 // g/cm-sec^2
9 l = 0.6 // cm
10 d = 13*10^-7 // cm
11 m = 2/(6.023*10^23) // gm/sec
12 M1 = 2.01
```

```

13 M2 = 28.0
14 sigma1 = 2.92 //cm
15 sigma2 = 3.68 //cm
16 sigma12 = (sigma1+sigma2)/2
17 omega = 0.80
18 deltac1 = (1/(22.4*10^3))*(T0/T)
19 // Calculations
20 DKn = (d/3)*(sqrt((2*kb*T)/m)) //cm^2/sec
21 flux1 = (DKn*deltac1/l)*10^5 //in x*10^-5 mol/cm^2-sec
22 D = (1.86*10^-3)*(T^(1.5))*(((1/M1)+(1/M2))^0.5)/(p
    *(sigma12^2)*omega)
23 flux2 = (D*deltac1/l)*10^11 // in x*10^-11 mol/cm^2-
    sec
24 // Results
25 printf("The steady diffusion flux is %.2f x10^-5 mol
    /cm^2-sec",flux1)
26 printf("\nThe flux through 18.3 micrometre pore is %
    .1f x10^-11 cm^2/sec",flux2)// answer wrong in
    text book

```

Scilab code Exa 6.4.4 Diffusion coefficient in gel

```

1
2 clc
3 //initialization of variables
4 d=0.01 //cm
5 s=2*10^-2 //cm
6 //calculations
7 phi = 4/3 *%pi*(d/2)^3 /(s^3)
8 disp("On solving , D")
9 D=5*10^-7 //cm^2/s
10 //results
11 printf("Diffusion in homogeneous gel = %.1e cm^2/sec
    ",D)

```

Chapter 8

Fundamentals of Mass Transfer

Scilab code Exa 8.1.1 Time taken

```
1
2 clc
3 //initiliazation of variables
4 Vap = (0.05/22.4)*23.8/760 // Vapour concentration
5 V = 18.4*10^3 // Air Volume in cc
6 A = 150 // Liquid Area in Cm^2
7 t1 = 180 // Time in sec
8 N1 = (Vap*V)/(A*t1)
9 k = 3.4*10^-2 // cm/sec
10 C = 0.9
11 //Calculations
12 t = (-V/(k*A))*log(1 - C)
13 thr = t/3600
14 //Results
15 printf("the time taken to reach 90 percent
    saturation is %.3f hr",thr)
```

Scilab code Exa 8.1.2 Mass transfer coefficient


```

1
2
3 clc
4 //initialization of variables
5 Vo = 5 // cm/sec
6 a = 23 //cm2/cm3
7 z = 100 //cm
8 Crat = 0.62 // Ratio of c/Csat
9 //Calculations
10 k = -(Vo/(a*z))*log(1-Crat)
11 //Results
12 printf("the mass transfer co efficient is %.1e cm/
    sec",k)

```

Scilab code Exa 8.1.3 Mass transfer coefficient

```

1
2
3 clc
4 //initialization of variables
5 t = 3*60 // seconds
6 crat = 0.5 // Ratio of c and csat
7 //calculations
8 ka = -(1/t)*log(1-crat)
9 //results
10 printf("the mass transfer co efficient along the
    product with a is %.1e sec-1",ka)

```

Scilab code Exa 8.1.4 Mass transfer coefficient

```

1
2
3 clc

```

```

4 //initialiazation of variables
5 rin = 0.05 // initial radius of oxygen bubble in cm
6 rf = 0.027 //final radius of oxygen bubble in cm
7 tin = 0 // initial time in seconds
8 tf = 420 // final time in seconds
9 c1 = 1/22.4 // oxygen concentration in the bubble in
    mol/litres
10 c1sat = 1.5*10^-3 // oxygen concentration outside
    which is saturated in mol/litres
11 //Calculations
12 k = -((rf-rin)/(tf-tin))*(c1/c1sat)
13 //Results
14 printf("The mass transfer co efficient is %.1e cm/
    sec",k)

```

Scilab code Exa 8.2.1 Mass transfer coefficient

```

1
2
3 clc
4 //initialization of variables
5 kc = 3.3*10^-3 // M.T.C in cm/sec
6 d = 1 // density of oxygen in g/cm^3
7 M = 18 // Mol wt of water in g/mol
8 Hatm = 4.4*10^4 // Henrys constant in atm
9 HmmHg = Hatm*760 // Henrys constant in mm Hg
10 //calculations
11 ratio = d/(M*HmmHg)// Ratio of concentration and
    pressure of oxygen
12 kp = kc*ratio // M.T.O=C in x*10^12mol/cm^2-sec-mm
    Hg
13 //Results
14 printf("the M.T.C in given units is %.1e",kp )

```

Scilab code Exa 8.2.2 Mass transfer coefficient liquid and gas

```
1
2
3 clc
4 //initialization of variables
5 k1 = 1.18 // M.T.C in lb-mol NH3/hr-ft^2
6 k2 = 1.09 // M.T.C in lb-mol NH3/hr-ft^2
7 M2 = 18 // Mol wt of NH3 in lb/mol
8 d = 62.4 // Density of NH3 in lb/ft^3
9 c1 = 30.5 // Conversion factor from ft to cm
10 c2 = 1/3600 // Conversion factor from seconds to
    hour
11 R = 1.314 // Gas constant in atm-ft^3/lb-mol-K
12 T = 298 // Temperature in Kelvin scale
13 //Calculations
14 kf1 = (M2/d)*k1*c1*c2 // M.T.C in cm/sec
15 kf2 = R*T*k2*c1*c2 // M.T.C in cm/sec
16 //Results
17 printf("the M.T.C for liquid is %.1e cm/sec",kf1)
18 printf("\\n the M.T.C for gas is %.1f cm/sec",kf2)
```

Scilab code Exa 8.3.1 Column length

```
1
2 clc
3 //initialization of variables
4 l = 0.07 // flim thickness in cm
5 v = 3 // water flow in cm/sec
6 D = 1.8*10^-5 // diffusion coefficient in cm^2/sec
7 crat = 0.1 // Ratio of c1 and c1(sat)
8 //Calculations
```

```

9 z = (((1^2)*v)/(1.38*D))*((log(1-crat))^2) //Column
    length
10 //Results
11 printf("the column length needed is %.1f cm",z)

```

Scilab code Exa 8.3.2 Mass flux in air and water

```

1
2 clc
3 //Initialization of variables
4 Dw = 1*10^-5 // Diffusion co efficient in cm^2/sec
5 omeg = 20*2*%pi/60 // disc rotation in /sec
6 Nuw = 0.01 // Kinematic viscosity in water in cm^2/
    sec
7 Da = 0.233 // Diffusion co efficient in cm^2/sec
8 Nua = 0.15 // Kinematic viscosity in air in cm^2/
    sec
9 c1satw = 0.003 // Solubility of benzoic acid in
    water in gm/cm^3
10 p1sat = 0.30 // Equilibrium Vapor pressure in mm Hg
11 ratP = 0.3/760 // Ratio of pressures
12 c1 = 1/(22.4*10^3) // Moles per volume
13 c2 = 273/298 // Ratio of temperatures
14 c3 = 122 // Grams per mole
15 //Calculations
16 kw = 0.62*Dw*((omeg/Nuw)^0.5)*((Nuw/Dw)^(1/3)) // cm/
    sec
17 Nw = kw*c1satw*10^6 // mass flux in x*10^-6 in g/cm
    ^2-sec
18 ka = 0.62*Da*((omeg/Nua)^0.5)*((Nua/Da)^(1/3)) //cm/
    sec
19 c1sata = ratP*c1*c2*c3 // Solubility of benzoic acid
    in air in gm/cm^3
20 Na = ka*c1sata*10^6 // mass flux in x*10^-6 in g/cm
    ^2-sec

```

```

21 //Results
22 printf("the mass flux in water is %.1f x10^-6 g/cm
      ^2-sec",Nw)
23 printf("\n the mass flux in air is %.1f x10^-6 g/cm
      ^2-sec",Na)

```

Scilab code Exa 8.5.1 Overall Mass transfer coefficient

```

1
2
3 clc
4 //initialization of variables
5 D1=2.1*10^-5// Diffusion co efficient for Oxygen in
      air in cm^2/sec
6 Dg = 0.23 //Diffusion co efficient for Oxygen in
      water in cm^2/sec
7 R = 82 // Gas constant in cm^3-atm/g-mol-K
8 T = 298 //Temperature in Kelvin
9 l1 = 0.01 // film thickness in liquids in cm
10 l2 = 0.1 // film thickness in gases in cm
11 H1 = 4.3*10^4 // Henrys constant in atm
12 c = 1/18 // concentration of water in g-mol/cm^3
13 //Calculations
14 k1 = (D1/l1)*c // m.t.c in liquid phase in mol/cm^2/
      sec
15 kp = (Dg/l2)/(R*T) // m.t.c in gas phase in gmol/cm
      ^2-sec-atm
16 KL = 1/((1/k1)+(1/(kp*H1))) // Overall m.t.c in mol/
      cm^2-sec liquid phase
17 //Results
18 printf("The overall m.t.c in liquid side is %.1e mol
      /cm^2-sec",KL)

```

Scilab code Exa 8.5.2 Overall Mass transfer coefficient

```
1
2 clc
3 //initialization of variables
4 D1=1.9*10^-5// Diffusion co efficient for liquid
   phase in cm^2/sec
5 Dg = 0.090 //Diffusion co efficient for gas phase in
   cm^2/sec
6 R = 82 // Gas constant in cm^3-atm/g-mol-K
7 T = 363 //Temperature in Kelvin
8 H1 = 0.70 // Henrys constant in atm
9 c = 1/97 // concentration of water in g-mol/cm^3
10 //Calculations
11 k1 = (D1/0.01)*c // m.t.c in liquid phase in mol/cm
   ^2/sec
12 kp = (Dg/0.1)/(R*T)// m.t.c in gas phase in gmol/cm
   ^2-sec-atm
13 KL = 1/((1/k1)+(1/(kp*H1)))*10^5// Overall m.t.c in
   x*10^-5 mol/cm^2-secliquid phase
14 //Results
15 printf("The overall m.t.c in liquid side is %.2f x10
   ^-5 mol/cm^2-sec",KL)// answer wrong in textbook
```

Scilab code Exa 8.5.3 Overall Mass transfer coefficient

```
1
2 clc
3 //Initialization of variables
4 k1 = 3.0*10^-4 // m.t.c in benzene in cm/sec
5 k2 = 2.4*10^-3 // m.t.c in water in cm/sec
6 ratio = 150 // Solubility ratio in benzene to water
7 //Calculations
8 K1 = (1/((1/k1)+(ratio/k2)))*10^5 // Overall m.t.c
   through benzene phase in x*10^-5 cm/sec
```

```

9 //Results
10 printf("The overall M.T.C through benzene phase is %
    .1f x10-5 cm/sec",K1)

```

Scilab code Exa 8.5.4 Overall Mass transfer coefficient

```

1
2 clc
3 //initialization of variables
4 H1 = 75 // henrys constant for ammonia in atm
5 H2 = 41000 // henrys constant for methane in atm
6 p = 2.2 // pressure in atm
7 kya = 18 // product of m.t.c and packing area per
    tower volume in lb-mol/hr-ft3
8 kxa = 530 //product of m.t.c and packing area per
    tower volume in lb-mol/hr-ft3
9 //calcuations
10 Kya1 = 1/((1/kya) + (H1/p)/kxa) //The overall
    coefficient for ammonia in lb-mol/hr-ft3
11 Kya2 = 1/((1/kya) + (H2/p)/kxa) //The overall
    coefficient for methane in lb-mol/hr-ft3
12 //Results
13 printf("The overall coefficient for ammonia is %.1f
    lb-mol/hr-ft3",Kya1)
14 printf("\n The overall coefficient for methane is %
    .2f lb-mol/hr-ft3",Kya2)

```

Chapter 9

Theories of Mass Transfer

Scilab code Exa 9.1.1 Film thickness

```
1
2 clc
3 //initialization of variables
4 p1 = 10 // pressure in atm
5 H = 600 // henrys constant in atm
6 c1 = 0 // gmol/cc
7 N1 = 2.3*10^-6 // mass flux in mol/cm^2-sec
8 c = 1/18 //total Concentration in g-mol/cc
9 D = 1.9*10^-5 // Diffusion co efficient in cm^2/sec
10 //Calculations
11 c1i = (p1/H)*c // Component concentration in gmol/cc
12 k = N1/(c1i-c1)//Mass transfer co efficient in cm/
    sec
13 l = D/k // Film thickness in cm
14 //Results
15 printf("The film thickness is %.5f cm",l)
```

Scilab code Exa 9.2.1 Contact time and residence time


```

1
2 clc
3 //initialization of variables
4 D = 1.9*10^-5 //Diffusion co efficient in cm^2/sec
5 k = 2.5*10^-3 // M.T.C in cm/sec
6 //Calculations
7 Lbyvmax = 4*D/((k^2)*%pi)//sec
8 tou = D/k^2 // sec
9 //Results
10 printf("The contact time is %.1f sec",Lbyvmax)
11 printf("\\nThe surface resident time is %.1f sec",tou
    )

```

Scilab code Exa 9.3.1 Apparent Mass transfer coefficient

```

1
2
3 clc
4 //initialization of variables
5 const = 0.5 // The part of flow in the system which
    bypasses the region where the mass transfer
    occurs
6 v1 = 1 // cm/sec
7 al = 10^3
8 k = 10^-3 // cm/sec
9 v2 = 3 // cm/sec
10 //Calculations
11 C1byC10first = const + (1-const)*(exp(-k*al/v1))//
    c1/c10
12 appk1 = (v1/al)*(log(1/C1byC10first))// Apparent m.t
    .c for first case in cm/sec
13 C1byC10second = const + (1-const)*(exp(-(sqrt(3))*k*
    al/v2))//c1/c10 in second case
14 appk2 = (v2/al)*log(1/C1byC10second)// apparent m.t.
    c for second case in cm/sec

```

```

15 power = log(appk2/appk1)/log(v2/v1)
16 //Results
17 printf("The apparent m.t.c for the first case is %.2
    e cm/sec",appk1)
18 printf("\nThe apparent m.t.c for the second case is
    %.2e cm/sec",appk2)
19 printf("\nThe apparent is proportional to the power
    of %.2f of the velocity",power)

```

Scilab code Exa 9.4.1 Average Mass transfer coefficient

```

1
2 clc
3 //initialization of variables
4 D = 1*10^-5 //cm^2/sec
5 d = 2.3 // cm
6 L = 14 // cm
7 v0 = 6.1 // cm/sec
8 //calculations
9 k = ((3^(1/3))/(gamma(4/3)))*((D/d))*(((d^2)*v0/(D*L
    ))^(1/3))// cm/sec
10 //Results
11 printf("The average mass transfer coefficient is %.6
    f cm/sec",k)

```

Scilab code Exa 9.4.2 Local Mtc

```

1
2 clc
3 //initialization of variables
4 tn = 300000 // turbulence number
5 v0 = 10 // cm/sec
6 p = 1 // g/cc

```

```

7 mu = 0.01 // g/cm-sec
8 delta = 2.5 //cm
9 D = 1*10^-5 // cm^2/sec
10 // Calculations
11 x = tn*mu/(v0*p) // cm
12 delta = ((280/13)^(1/2))*x*((mu/(x*v0*p))^(1/2)) //cm
13 deltac = ((D*p/mu)^(1/3))*delta //cm
14 k = (0.323*(D/x)*((x*v0*p/mu)^0.5)*((mu/(p*D))^(1/3)
    ))*10^5 // x*10^-5 cm/sec
15 // Results
16 printf("The distance at which turbulent flow starts
    is %.f cm",x)
17 printf("\nThe boundary layer for flow at this point
    is %.1f cm",delta)
18 printf("\nThe boundary layer for concentration at
    this point is %.2f cm",deltac)
19 printf("\nThe local m.t.c at the leading edge and at
    the position of transistion is %.1f x10^-5 cm/
    sec",k)

```

Chapter 10

Absorption

Scilab code Exa 10.2.1 Diameter of tower

```
1
2 clc
3 //initialization of variables
4 c = 0.92
5 F = 93 // ft-1
6 nu = 2 // cs
7 d1 = 63 // lb/ft3
8 dg = 2.8 // lb/ft3
9 G = 23 //lb/sex
10 //Calculations
11 G11 = c*((d1-dg)0.5/(((F)0.5*(nu0.05)) // lb/ft
    ^2-sec
12 A = G/G11 // ft2
13 d = sqrt(4*A/%pi) // ft
14 //Results
15 printf("The diameter of the tower is %.1f ft",d)
```

Scilab code Exa 10.3.1 Length of tower

```

1
2 clc
3 //Initialization of variables
4 G = 2.3 // Gas flow in gmol/sec
5 L = 4.8 // Liquid flow in gmol/sec
6 y0 = 0.0126 // entering gas Mole fraction of CO2
7 y1 = 0.0004 // Exiting gas mole fraction of CO2
8 x1 = 0 // Exiting liquid mole fraction of CO2
9 d = 40 // Diameter of the tower in cm
10 x0star = 0.0080 // if the amine left in equilibrium
    with the entering gas would contain 0.80 percent
    CO2
11 Kya = 5*10^-5 // Overall M.T.C and the product times
    the area per volume in gmol/cm^3-sec
12 //Calculations
13 A =%pi*(d^2)/4
14 x0 = ((G*(y0-y1))/(L)) + x1 // Entering liquid mole
    fraction of CO2
15 m = y0/x0star // Equilibirum constant
16 c1 = G/(A*Kya)
17 c2 = 1/(1-(m*G/L))
18 c3 = log((y0-m*x0)/(y1-m*x1))
19 l = (G/(A*Kya))*(1/(1-((m*G)/L)))*(log((y0-m*x0)/(y1
    -m*x1)))/100 //length of the tower in metres
20 //Results
21 printf("The length of the tower is %.1f m",l)

```

Scilab code Exa 10.3.2 Percentage of oxygen

```

1
2
3 clc
4 //initialization of variables
5 l = 200 // Length of the tower in cm
6 d = 60 // diameter of the tower

```

```

7 Lf = 300 // Liquid flow in cc/sec
8 Kx = 2.2*10^-3 // dominant transfer co efficient in
  liquid in cm/sec
9 //Calculations
10 A = %pi*60*60/4 // Area of the cross section in sq
  cm
11 L = Lf/A // Liquid flux in cm^2/sec
12 ratio = 1/(exp((1*Kx)/L))
13 percentage = (1-ratio)*100 // Percentage removal of
  Oxygen
14 //Results
15 printf("the percentage of oxygen we can remove is %
  .1f",percentage)
16
17
18
19 // Rounding of error in textbook

```

Scilab code Exa 10.4.1 Length and diameter of tower

```

1
2
3 clc
4 //initialization of variables
5 y1in = 0.37 // mole fraction of Ammonia in gas
  mixture entering
6 y2in =0.16 // mole fraction of nitrogen in gas
  mixture entering
7 y3in = 0.47 // mole fraction of hydrogen in gas
  mixture entering
8 x1out = 0.23 // mole fraction of Ammonia in liquid
  coming out
9 y1out = 0.01 // mole fraction of ammonia in gas
  coming out
10 G0 = 1.20 // Gas glow entering in m^3/sec

```

```

11 Mu = 1.787*0.01*0.3048/2.23 // liquid viscosity in
    american units
12 dl = 62.4 // Density of liquid in lb/ft^3
13 KG = 0.032 // Overall m.t.c in gas phase in gas
    side m/sec
14 a = 105 // surface area in m^2/m^3
15 gc = 32.2 // acceleration due to gravity in ft/sec^2
16 dg = 0.0326 // Density of gas in lb/ft^3
17 //Molecular weights of Ammonia , N2 , H2
18 M1 = 17
19 M2 = 28
20 M3 = 2
21 Nu = 1 // Viscosity
22 //Calculations
23 AG0 = (y2in+y3in)*G0/22.4 // Total flow of non
    absorbed gases in kgmol/sec
24 ANH3 = y1in*G0/22.4- (y1out*AG0)/(1-y1out) //
    Ammonia absorbed kgmol/sec
25 AL0 = ((1-x1out)/x1out)*ANH3 // the desired water
    flow in kgmol/sec
26 avg1 = 11.7 // Average mol wt of gas
27 avg2 = 17.8 // avg mol wt of liquid
28 TFG = avg1*AG0/(y2in+y3in)//Total flow of gas in kg/
    sec
29 TFL = avg2*AL0/(1-x1out)//total flow of liquid in kg
    /sec
30 F = 45 // Packing factor
31 GFF = 1.3*((dl-dg)^0.5)/((F^0.5)*(Nu^0.05))// Flux
    we require in lb/ft^2-sec
32 GFF1 = GFF*0.45/(0.3^2) // in kg/m^2-sec (answer
    wrong in textbook)
33 Area = TFG/GFF1 // Area of the cross section of
    tower
34 dia = (sqrt(4*Area/%pi)) // diameter in metres
35 HTU = (22.4*AG0/%pi*dia^2)/(KG*a*4)
36 NTU = 5555
37 l = HTU*NTU // Length of the tower
38 //Results

```

```
39 printf("The flow of pure water into the top of the
    tower %.4f kgmol/sec",AL0)
40 printf("\n The diameter of the tower is %.1f m",dia)
41 printf("\n The length of the tower is %.f m",l)
```

Chapter 11

Mass Transfer in Biology and Medicine

Scilab code Exa 11.1.1 Mass transfer coefficient

```
1
2 clc
3 // Initialization of variables
4 N1 = 1.6*10^-10 // mol/cm^2-sec
5 c1star = 0 // mol/cc
6 c1 = 2.7*10^-4/1000 // mol/cc
7 // Calculations
8 K = N1/(c1-c1star) // cm/sec
9 // Results
10 printf("The mass transfer coefficient is %.4f cm/
    sec",K)
```

Scilab code Exa 11.2.1 Mass transfer coefficient

```
1
2
```

```

3  clc
4  //Initialization of variables
5  d = 400*10^-4 // cm
6  D = 10^-5 // cm^2/sec
7  v = 1 // cm/sec
8  l = 30 // cm
9  nu = 0.01 // cm^2/sec
10 //Calculations
11 k1 = (D/d)*1.62*(((d^2)*v/D*1)^(1/3))// Mass
    transfer co efficient inside the hollow fibers in
    cm/sec
12 k2 = (D/d)*0.8*((d*v/nu)^0.47)*((nu/D)^(1/3))//Mass
    transfer co efficient outside the hollow fibers
    in cm/sec
13 //Results
14 printf("Mass transfer co efficient inside the hollow
    fibers %.2e cm/sec",k1)
15 printf("\nMass transfer co efficient outside the
    hollow fibers %.1e cm/sec",k2)

```

Scilab code Exa 11.2.2 Percentage of toxins

```

1
2  clc
3  //initialization of variables
4  phi = 0.2
5  d = 200*10^-4 // cm
6  dia = 3.8 // cm
7  Q = 4.1 // blood flow in cc/sec
8  k = 3.6*10^-4 // cm/sec
9  l = 30 // cm
10 //Calculations
11 a = 4*phi/d // cm^2/cm^3
12 B = Q/((%pi*dia^2)/4) // cm/sec
13 ratio1 = 1/(1+(k*a*l/B))// D equals B

```

```

14 percent1 = (1-ratio1)*100 // percentage of toxins
    removed when dialystate flow equals blood flow
15 D = 2*B // in second case
16 ratio2 =1/(((1/(exp(-k*a*1/D))))-0.5)*2) // when D =2
    B
17 percent2 = (1-ratio2)*100 // percentage of toxins
    removed when dialystate flow is twice the blood
    flow
18 ratio3 = exp(-k*a*1/B)// when dialystate flow is
    very large
19 percent3 = (1-ratio3)*100 // percentage of toxins
    removed when dialystate flow is very large
20 //Results
21 printf("The percentage of toxins removed when
    dialystate flow equals blood flow is %.f",
    percent1)
22 printf("\nThe percentage of toxins removed when
    dialystate flow is twice the blood flow is %.f",
    percent2)
23 printf("\nThe percentage of toxins removed when
    dialystate flow is very large is %.f",percent3)

```

Chapter 12

Differential Distillation

Scilab code Exa 12.2.1 Mass transfer

```
1
2 clc
3 //initialization of variables
4 l = 1.22 // length of tower
5 Gflow = 0.026 // mol/sec
6 GbyL = 0.07
7 dia = 0.088 // m
8 p1 = 1.1/100 // p1 = 1-y1
9 p0 = 0.04/100 // p0 = 1-y0
10 //Calculations
11 A = %pi*(dia^2)/4 // cross sectional of tower in m^2
12 G = Gflow/A // Gas flux in mol/m^2-sec
13 Kya = (G/l)*(1/(1-GbyL))*(log(p1/p0)) // Mass
    transfer per volume in mol/m^3-sec
14 //Results
15 printf("The mass transfer per volume is %.1f mol/m
    ^3-sec",Kya)
```

Scilab code Exa 12.2.2 NTU

```

1
2 clc
3 // Initialization of variables
4 x1=0.99
5 x2=0.99
6 y1=0.95
7 y2=0.95
8 alpha=1.5
9 m=0.42
10 l=2
11 HTU=0.34
12 // calculations
13 y1s= (y1-0.58)/m
14 xrd= (x2-y2)/(x1-y1s)
15 Rd=xrd/(1-xrd)
16 Rds=alpha*Rd
17 x1= ((Rds+1)*y1 - x1)/(Rds)
18 function z= ystar(y)
19     z=0.58+0.42*y
20 endfunction
21 NTU = log((ystar(x1) -y1)/(ystar(x1)-x1)) /(1- m*(
    Rds+1)/Rds)
22 NTU2=1/HTU
23 xd2=(ystar(y1)-y1)/%e^(NTU2*(1-m))
24 xd=(0.58-xd2)/(1-m)
25 // results
26 printf(" In case 1, NTU = %.2 f",NTU)
27 printf(" \n In case 2, xd = %.3 f",xd)

```

Scilab code Exa 12.4.1 length of tower

```

1
2 clc
3 // initialization of variables
4 F=3500 //mol/hr

```

```
5  xf=0.4
6  x1=0.98
7  y1=0.97
8  y2=0.625
9  x1=0.97
10 x2=0.4
11 ratio=1.5
12 HTU=0.2
13 // calculations
14 A=[1 1;x1 1-x1]
15 B=[F;xf*F]
16 C=A\B
17 DA=C(1)
18 BA=C(2)
19 Rds=(y1-y2)/(x1-x2)
20 Rd=Rds/(1-Rds)
21 Rdreq=ratio*Rd
22 NTU=13.9
23 l=HTU*NTU
24 // results
25 printf("length of the tower = %.1f m",l)
```

Chapter 13

Staged Distillation

Scilab code Exa 13.1.1 Column diameter

```
1
2
3 clc
4 //Intialization of variables
5 xD = 0.90 // Distillate mole fraction
6 xB = 0.15// Reboiler mole fraction
7 xF = 0.50 //Feed mole fraction
8 F = 10 // mol/sec
9 dg = 3.1*10^-3 // g/cc
10 dl = 0.65 // g/cc
11 C = 0.11 // m/sec
12 //Calculations
13 D = ((xF*F)-(xB*F))/(xD-xB)
14 B = ((xF*F)-(xD*F))/(xB-xD)
15 L = 3.5*D
16 G = L+D
17 L1 = L+F
18 G1 = G
19 f = (L1/G1)*(sqrt(dg/dl)) // flow parameter
20 vG = C*(sqrt((dl-dg)/dg))//vapor velocity in m/sec
21 c = (22.4*10^-3)*340/373
```

```

22 d = sqrt(4*G1*c/(vG*%pi))//m
23 //Results
24 printf("The column diameter is %.1f m",d)
25
26 //Calculation mistake in textbook

```

Scilab code Exa 13.2.1 Number of stages

```

1
2
3 clc
4 //Initialization of variables
5 y1 = 0.9999
6 x0 = y1 // For a total condenser
7 y0 =0.58 + 0.42*x0 // The equilbirum line
8 LbyG = 0.75
9 yNplus1 = 0.99
10 A = LbyG/0.42
11 n= 1
12 //Calculations
13 xN = (yNplus1 - ((1-LbyG)*y1))/LbyG
14 yN = 0.58 + 0.42*xN
15 N = (log((yNplus1-yN)/(y1-y0))/log(A))+n// , number
    of stages
16 //Results
17 printf("the number of stages approximately is %.f",N
    )

```

Scilab code Exa 13.2.2 Number of stages

```

1
2
3 clc

```



```

4 //initialization of variables
5 x0 = 0.0082
6 xB = 10^-4
7 L = 1
8 //Calculations
9 y0 = 36*x0
10 //There are two balancing equations , mole fraction
    balance , mole balance , from them G is
11 G0 = (xB-x0)*L/(xB-y0)
12 G = 3*G0
13 B = L-G
14 y1 = ((L*x0)-(B*xB))/G
15 yNplus1 = 36*xB
16 xN = (L*x0 - (G*(y1-yNplus1)))/L
17 yN = 36*xN
18 N = (log((yNplus1-yN)/(y1-y0)))/log((yNplus1-y1)/(yN
    -y0))
19 //Results
20 printf("The number of stages are %.1f",N)
21 //Answer wrong in textbook

```

Scilab code Exa 13.4.1 Murphree efficiency

```

1
2 clc
3 //initialization of variables
4 yn = 0.84
5 ynplus1 = 0.76
6 ystarn = 0.874
7 GA = 0.14 // kg-mol/sec
8 A1 = 0.04 // m^3
9 //Calculations
10 Murphree = (yn-ynplus1)/(ystarn-ynplus1)
11 Kya = GA/(A1*((1/Murphree)-1))
12 //results

```

```
13 printf("the murphree efficiency is %.2f",Murphree)
14 printf("\n the m.t.c along with the product with a
    is %.2f kg-mol/m^3-sec",Kya)
```

Scilab code Exa 13.4.2 Murphree efficiency

```
1
2 clc
3 //Initialization of variables
4 R = 82 // atm-cm^3/gmol-K
5 T = 273 + 60 // Kelvin
6 pk = 1 // atm
7 a1 = 440 // sec^-1 (of gas)
8 a2 = 1.7 //sec^-1 (of liquid)
9 ck = 1.5/((0.47*(76.1))+(0.53*(158.7)))
10 x = 0.2
11 Vs = 10 // litres
12 GA = 59 // gmol/sec
13 m = 1.41
14 //Calculations
15 k = (R*T)/(pk*a1) + (m/(ck*a2))
16 Kya = (1/k)*1000 // gmol/l-sec
17 Murphree = 1 - exp(-Kya*Vs/(GA))
18 //Results
19 printf("The murphree efficiency is %.2f",Murphree)
```

Chapter 14

Extraction

Scilab code Exa 14.3.1 Length required

```
1
2 clc
3 //initialization of variables
4 Rat1 = (6.5/3)*(1-0.47) // as Rat = x0/y0
5 m = 0.14
6 H = (6.5*10^3)/3600 // Extract flow in g/sec
7 L = (3*10^3)/3600 // Solvent flow in g/sec
8 d= 10 // cm
9 A = 0.25*%pi*d^2 // cm^2
10 l = 65 // cm
11 //Calculations and Results
12 Kya = ((H/(1*A))*(1/(1-((m*H)/L)))*(log((1-0.14*Rat1
    )/(0.47))))*10^3 // kg/m^3-sec
13 printf("The value of Kya is %.2f kg/m^3-sec",Kya)
14 Rat2 = (6.5/3)*(1-0.1) //For case B
15 l2 = l*(log(1/((1-0.14*Rat2)/(0.1))))/(log
    (1/((1-0.14*Rat1)/(0.47))))/100 // m
16 printf("\\nThe length for 90 percent recovery is %.1f
    m",l2)
```

Scilab code Exa 14.4.1 No of stages required

```
1
2
3 clc
4 //Initialization of variables
5 m = 0.018
6 H = 450 // litres/hr
7 L = 37 // litres/hr
8 Ynplus1byY1 = 100
9 //Calculations
10 E =m*H/L
11 nplus1 = log((Ynplus1byY1*((1/E)-1))+1)/log(1/E)
12 n = nplus1 -1
13 printf("The number of ideal stages are %.f",n)
14 N = 0.60//Murphree efficienct
15 E1 = (m*H/L) + (1/N) - 1
16 nplus1 = log((Ynplus1byY1*((1/E1)-1))+1)/log(1/E1)
17 n=nplus1-1
18 printf("\\nThe number of stages required if Murphree
    efficiency is 60 percent is %.f",n)
```

Scilab code Exa 14.5.1 No of stages including feed

```
1
2 clc
3 //initialization of variables
4 F = 5 //kg feed
5 S = 2 // kg solvent
6 E = F-S // kg extract
7 W = 1 // kg waste
8 EG = 80 // ppm
```

```
9 y0 = (100-99)/100 // mole fraction of gold left
10 y1 = y0*EG*W/S // concentration in raffinate
11 // Calculations
12 xN = (EG*W - y1*S)/E // solvent concentration
13 xNminus1 = ((xN*(E+S)) - EG*W)/F // feed stage balance
14 N = 1 + ((log((xN-xNminus1)/(y1))/log(F/S))) // number
    of stages including feed stage
15 // Results
16 printf("The number of stages including feed stage is
    %.f",N)
```

Chapter 15

Adsorption

Scilab code Exa 15.3.2 Length of bed

```
1
2 clc
3 //initialization of variables
4 tE = 33 // Time taken for ferric ion to exhaust the
   bed in min
5 tB = 23 // Time taken for nickel to break through
   ferric in min
6 l = 120 //bed length in cm
7 //Calculations
8 Theta = 2*tB/(tB+tE)
9 lunused = (1-Theta)*120 // cm
10 //Results
11 printf("the length of the bed unused is %.1f cm",
   lunused)
```

Scilab code Exa 15.3.3 volume of adsorbent

```
1
```

```

2  clc
3  //initialization of variables
4  tB = 10 // min
5  tE = 14 // min
6  l = 0.12 //m
7  l2 = 10 // m
8  c = 10000
9  A = 1/10000 // m^2
10 //Calculations
11 theta = 2*tB/(tB+tE)
12 l1 = l*(1-theta)// m , length of bed unused in first
    case
13 V1 = c*A*l // m^3
14 l3 = l2-l1 // length of bed unused in second case
15 d = sqrt(V1*4/(l3*pi))// m
16 V2 = c*(l-l1)*A*l2/l3 // volume needed for second
    case
17 //Results
18 printf("The volume of adsorbent needed if the bed is
    kept 12 cm deep is %.2f m^3",V1)
19 printf("\nThe volume of adsorbent needed if the bed
    length is 10 m long is %.4f m^3",V2)

```

Scilab code Exa 15.4.1 Breakthrough time

```

1
2  clc
3  //intialization of variables
4  tB1 = 38 // days , breakthrough time
5  tE1 = 46 // days , exhaustion time
6  c = 2 // number of times flow doubled
7  //Calculations
8  theta1 = 2*tB1/(tB1+tE1)// in the first case
9  ratio1 = 1-theta1 // ratio of unused bed length to
    total bed length

```

```

10 ratio2 = ratio1*c
11 tB2 = ((1/c)*(tB1 + 0.5*(tE1-tB1)))*ratio2//
    breakthrough time for second case
12 tE2 = (c-ratio2)*tB2/ratio2//exhaustion time for
    second case
13 //Results
14 //answer slightly wrong in textbook
15 printf("The breakthrough time for this case is %.1f
    days",tB2)

```

Scilab code Exa 15.4.2 rate constant

```

1
2 clc
3 //initialization of variables
4 slope = 0.93/3600 // sec-1
5 q0 = 300 // 300 times y0
6 E = 0.4 // void fraction
7 d = 310*10-4 //cm
8 v = 1/60 //cm/sec
9 Nu = 0.01 //cm2/sec
10 D = 5*10-6 //cm2/sec
11 //Calculations
12 ka1 = slope*q0*(1-E) //sec-1
13 k = (D/d)*1.17*((d*v/Nu)0.58)*((Nu/D)0.33) // cm/
    sec
14 a = (6/d)*(1-E) //cm2/cm3
15 ka2 = k*a //sec-1
16 //Results
17 printf("The rate constant is %.3f sec-1",ka1)
18 printf("\n\nThe rate constant of literature is %.3f
    sec-1",ka2)

```

Chapter 16

General Questions and Heterogenous Chemical Reactions

Scilab code Exa 16.3.2 rate constant

```
1
2 clc
3 //initialization of variables
4 K = 0.0087 // overall m.t.c in cm/sec
5 D = 0.98*10^-5 // cm^2/sec
6 L = 0.3 // cm
7 v = 70 // cm/sec
8 nu = 0.01 //cm^2/sec
9 //Calculations
10 k1 = 0.646*(D/L)*((L*v/nu)^(0.5))*((nu/D)^(1/3)) //
    cm/sec
11 k2 = (1/((1/K)-(1/k1))) /// cm/sec
12 //Results
13 printf("The rate constant for the reaction is %.2f
    cm/sec",k2)
```

Scilab code Exa 16.3.3 rate of reaction

```
1
2
3 clc
4 //initialization of variables
5 D =2*10^-6 // cm^2/sec
6 nu = 0.036 // cm^2/sec
7 d1 = 1.59 // cm
8 d2 = 1 // cm
9 deltap = 1*10^-5 // g/cc ( change in density)
10 p = 1 // g/cc
11 Re = 11200 // Reynolds number
12 g = 980 // cm/sec^2
13 dis = 5.37*10^-9 // g/cm^2-sec // Dissolution rate
14 sol = 1.48*10^-3 // g/cc
15 //Calculations
16 k11 = 0.62*(D/d1)*(Re^(0.5))*((nu/D)^(1/3)) // cm/sec
17 K1 = dis/sol // the overall mass transfer co
    efficient in cm/sec
18 k2 = (1/((1/K1)-(1/k11)))/ // cm/sec // the rate
    constant in cm/sec
19 k12 = (D/d2)*(2+(((0.6*((d2^3)*(deltap)*g/(p*nu^2)))
    ^0.25)*((nu/D)^(1/3)))) // cm/sec
20 K2 = 1/((1/k12)+(1/k2)) // cm/sec (the overall mtc)
21 //Results
22 printf("the rate of surface reaction is %.7f cm/sec"
    ,k2)
23 printf("\\nThe dissolution rate for 1 cm gallstone is
    %.7f cm/sec",K2)
```

Chapter 17

Homogenous Chemical Reactions

Scilab code Exa 17.1.1 Diffusion coefficient

```
1
2 clc
3 //initialization of variables
4 K = 1.46*10^-4 // lit/mol-sec (rate constant)
5 cpyridine = 0.1 // mol/lit
6 K1 = 2.0*10^-5 // cm^2/sec
7 //Calculations
8 D = K*cpyridine // sec^-1
9 k0 = (sqrt(D*K1))*10^5//in x*10^-5 cm/sec
10 //Results
11 printf("The diffusion co efficient of methyl iodide
    in benzene is %.1f x10^-5 cm/sec",k0)
```

Scilab code Exa 17.1.2 Reduction in reaction rate

```
1
```

```

2  clc
3  //initialization of variables
4  R = 0.3 // cm
5  K1 = 18.6 // sec-1
6  D = 0.027 // cm2/sec
7  //Calculations
8  l = R/3 // cm
9  n = (sqrt(D/(K1*(l2))))*coth(sqrt(K1*(l2)/D))
10 //Results
11 printf("The value of reduction in reaction rate due
    to diffusion is %.2f",n)

```

Scilab code Exa 17.1.3 rate constant

```

1
2  clc
3  //initialization of variables
4  k = 16*10-3 // m.t.c in cm/sec
5  D = 1.25*10-5 // Diffusion co efficient in cm2/
    sec
6  //Calculations
7  K1 = (k2)/D
8  //Results
9  printf("The rate constant is %.f sec-1",K1)

```

Scilab code Exa 17.2.1 Increase in rate

```

1
2  clc
3  //initialization of variables
4  D2 = 5*10-6 // The diffusion co efficient of the
    new compound in cm2/sec
5  Nu = 3 // The factor

```

```

6 D1 = 0.7*10^-5 // The diffusion co efficient of the
   original compound in cm^2/sec
7 c21 = 1.5*10^-5 // the new solubility in mol/cc
8 c11 = 3*10^-7 // The old solubility in mol/cc
9 //Calculations
10 k = 1 + ((D2*c21)/(Nu*D1*c11))// The number of times
   the rate has increased to the previous rate
11 //Results
12 printf("There is about a %.f fold increase in rate",
   k)

```

Scilab code Exa 17.4.1 Rate constant

```

1
2
3 clc
4 //initialization of variables
5 //For first reaction
6 D1 = 9.3*10^-5 // cm^2/sec
7 D2 = 5.3*10^-5 // cm^2/sec
8 K1exp = 1.4*10^11 // litre/mol-sec
9 sigma12 = 2.8*10^-8 // cm
10 N = (6.02*10^23)/10^3// liter/cc-mol
11 K1 = 4*pi*(D1+D2)*sigma12*N // Rate constant for
   first reaction in litre/mol-sec
12 printf("The rate constant for this reaction is %.1e
   litre/",K1)
13 if K1>K1exp
14     then disp("This reaction is controlled more by
   chemical factors")
15     else
16         disp("This reaction is diffusion controlled
   ")
17 end
18 //Second reaction

```

```

19 D1 = 5.3*10^-5 // cm^2/sec
20 D2 = 0.8*10^-5 // cm^2/sec
21 sigma12 = 5*10^-8 // cm
22 K1exp = 3.8*10^7 // litre/mol-sec
23 K1 = 4*pi*(D1+D2)*sigma12*N // Rate constant for
    second reaction in litre/mol-sec
24 printf("The rate constant for this reaction is %.1e
    litre/mol-sec",K1)
25 if K1>K1exp then
26     disp("This reaction is controlled more by
    chemical factors")
27 else
28     disp("The reaction is diffusion controlled")
29 end

```

Scilab code Exa 17.5.1 relaxation time

```

1
2 clc
3 //intitailization of variables
4 d = 5 // cm
5 v = 200 // cm/sec
6 nu = 0.01 // cm^2/sec
7 D = 3.2*10^-5 // cm^2/sec
8 l = 30*10^-4 // cm
9 //Calculations
10 Re = d*v/nu // Flow is turbulent
11 E = d*v/2 // cm^2/sec
12 tou1 = (d^2)/(4*E) // sec
13 tou2 = (l^2)/(4*D)
14 tou = tou1 + tou2 // sec
15 //Results
16 printf("The relaxation time is %.2f sec",tou)

```

Chapter 18

Membranes

Scilab code Exa 18.1.1 Rapidness calculation

```
1
2 clc
3 //initialization of variables
4 d = 240*10^-4 // cm
5 D = 2.1*10^-5 // cm^2/sec
6 v = 10 // cm/sec
7 Nu = 0.01 // cm^2/sec
8 E = 0.5
9 ka1 = 0.09 // sec^-1
10 //Calculations
11 k = 0.8*(D/d)*((d*v/Nu)^0.47)*((Nu/D)^0.33)
12 a = 4*(1-E)/d // cm^2/cm^3
13 ka2 = k*a
14 ratio = ka2/ka1
15 //results
16 printf("The rapidness is roughly %.f times that of
    sparger",ratio)
```

Scilab code Exa 18.2.1 Permeability calculation

```

1
2 clc
3 //initialization of variables
4 p1 = 10^-10 // cm^3(stp)cm/cm^2-sec-cm-Hg
5 c = 1/(22.4*10^3) // mol at stp /cc
6 P = p1*c // for proper units
7 R = 6240 // cmHg cm^3 //mol-K (gas constant)
8 T = 298 // Kelvin
9 //Calculations
10 DH = P*R*T*10^9 // Permeability in x10^-9 cm^2/sec
11 //Results
12 printf("The permeability is %.1f x10^-9 cm^2/sec",DH
    )

```

Scilab code Exa 18.2.2 Time spent

```

1
2 clc
3 //initialization of variables
4 P = 1*10^-4 // membrane permeability in cm^2/sec
5 l = 2.3*10^-4 // membrane thickness in cm
6 d = 320*10^-4 // fiber dia in cm
7 E = 0.5 // void fraction
8 c0 = 1 // initial concentration
9 c = 0.1 // final concentration
10 //Calculations
11 a = 4*(1-E)/d // surface area per module volume in
    cm^2/cm^3
12 t = (log(c0/c))*(1/P)/a // t = z/v in seconds , time
    gas spends in the module in sec
13 //Results
14 printf("The gas spends %.2f sec in the module",t)

```

Scilab code Exa 18.3.1 Osmotic pressure difference

```
1
2 clc
3 //initialization of variables
4 R = 0.082 // litre-atm/mol-K
5 T = 283 // Kelvin
6 V2 = 0.018 // litre/mol
7 //For first solution contents are sucrose and water
8 w1 = 0.01 // gm of sucrose
9 MW1 = 342 // MW of sucrose
10 w2 = 0.09 // gm of water
11 MW2 = 18 // MW of water
12 n1 = 1 // no of particles sucrose divides into in
    water
13 //Calculations
14 x1juice = (n1*w1/MW1)/((n1*w1/MW1)+(w2/MW2)) // Mole
    fracion of sucrose
15 //For second solution , contents are NaCl and water
16 w1 = 35 // gm of NaCl
17 MW1 = 58.5 // MW of Nacl
18 w2 = 100 // gm of water
19 MW2 = 18 // MW of water
20 n1 = 2 // no of particles sucrose divides into in
    water
21 //Calculations
22 x1brine = (n1*w1/MW1)/((n1*w1/MW1)+(w2/MW2)) // Mole
    fracion of sucrose
23 //Calculation of difference in Osmotic pressure
24 DeltaPi = (R*T/V2)*log((1-x1juice)/(1-x1brine)) //
    atm
25 //Results
26 printf("The osmotic pressure difference is %.f atm",
    DeltaPi)
27 //answer wrong in textbook
```

Scilab code Exa 18.3.2 Transport coefficient

```
1
2 clc
3 //initialization of variables
4 D1=0.0035
5 l=2.59 //cm
6 t=1.62 //hr
7 C1=0.03 //mol/l
8 T1=298 //K
9 R=0.0821 //arm/mol K
10 D2=0.005
11 t2=0.49 //hr
12 Ps=733 //mm of Hg
13 P=760 //mm of Hg
14 //calculations
15 Lps=D1*l/(t*3600) /(C1*R*T1)
16 Lp=(D2*l/(t2*3600) + Lps*(C1*R*T1))/(Ps/P)
17 Lp=2.4*10^-6
18 sig=Lps/Lp
19 sig2=0.95
20 //results
21 printf("Transport coefficient for phase 1 = %.2f",
        sig)
22 printf("\\n Transport coefficient for phase 2 = %.2f"
        ,sig2)
```

Scilab code Exa 18.4.1 Membrane selectivity

```
1
2 clc
3 //initialization of variables
```

```

4 D1 = 3.0*10^-7 // cm^2/sec
5 H1 = 0.18 // mol/cc-atm
6 D2 = 1.4*10^-6 // cm^2/sec
7 H2 = 2.2*10^-3 // mol/cc-atm
8 H11 = 13 // atm-cc/mol
9 H21 = 0.6 // atm-cc/mol
10 // Calculations
11 Beta = (D1*H1/(D2*H2))*(H11/H21) // Membrane
    selectivity
12 // Results
13 printf("The membrane selectivity is %.f",Beta)

```

Scilab code Exa 18.5.2 total flux

```

1
2 clc
3 // Initialization of variables
4 D = 2*10^-5 // cm^2/sec
5 l = 32*10^-4 // cm
6 c = 6.8*10^-6 // mol/cc
7 C10 = 10^-4 // mol/cc
8 function [j] = Totalflux(H,K)
9     j = (D*H*C10/l)+((D*H*K*c*C10)/(l*(1+(H*K*C10))))
10 endfunction
11 //For Lithium Chloride
12 H1 = 4.5*10^-4 // Partition coefficient
13 K1 = 2.6*10^5 // cc/mol association constant
14 j1 = (Totalflux(H1,K1))*10^10 // TOfal flux in x
    *10^-10 mol/cm^2-sec
15 printf("The total flux for Lithium Chloride is %.1f
    x10^-10 mol/cm^2-sec",j1)
16 //For Sodium Chloride
17 H2 = 3.4*10^-4 // Partition coefficient
18 K2 = 1.3*10^7 // cc/mol association constant

```

```
19 j2 = (Totalflux(H2,K2))*10^10 // TOfal flux in x
    *10^-10 mol/cm^2-sec
20 printf("\nThe total flux for Sodium Chloride is %.1f
    x10^-10 mol/cm^2-sec",j2)
21 //For potassium Chloride
22 H3 = 3.8*10^-4 //Partition coefficient
23 K3 = 4.7*10^9 // cc/mol association constant
24 j3 = (Totalflux(H3,K3))*10^10 // TOfal flux in x
    *10^-10 mol/cm^2-sec
25 printf("\nThe total flux for Potassium Chloride is %
    .1f x10^-10 mol/cm^2-sec",j3)
```

Chapter 19

Controlled Release and Phenomena

Scilab code Exa 19.1.1 Permeability

```
1
2
3 clc
4 //initialization of variables
5 VP = 0.045*10^-3 // Vapor pressure of permethrin in
   kg/m-sec^2
6 R = 8.31 // Gas constant in kg-m^2/sec^2-gmol-K
7 l = 63*10^-6 // membrane thickness in m
8 A = 12*10^-4 // area surrounded by the membrane in m
   ^2
9 M1 = 19*10^-3 // Permethrin release in gmol
10 t = 24*3600 // time taken to release
11 T = 298 // Kelvin
12 MW = 391 // Mol wt
13 //Calculations
14 c1 = VP/(R*T) // C1sat
15 P = (M1/(t*MW))*(1/c1)*(1/A)*10^-3 //Permeability in
   cm^2/sec
16 //Results
```

```
17 printf("The permeability is %.1e m^2/sec",P)
```

Scilab code Exa 19.2.1 Membrane area calculation

```
1
2 clc
3 //initialization of variables
4 M= 25*10^-6 //gm/hr
5 d = 0.006 //g/cc
6 P = 1.4*10^-4 // permeance in cm/sec
7 Deltac1 = 0.006 //Equivalent//cc
8 //Calculations
9 c1 = 1/3600 // unit conversion factor hr/sec
10 c2 = 1/18 //unit conversion factor mole/cc
11 m = M*c1*c2/d // moles/sec
12 A = m/(P*Deltac1)//cm^2
13 //Results
14 printf("you will need a membrane area of %.3f cm^2"
    ,A)
```

Chapter 20

Heat Transfer

Scilab code Exa 20.1.1 Thermal diffusivity

```
1
2 clc
3 //initialization of variables
4 T = 26.2 // centigrade
5 T0 = 4 // centigrade
6 Tinf = 40 //centigrade
7 z = 1.3//cm
8 t = 180 //seconds
9 //calculations
10 k = erfinv((T-T0)/(Tinf-T0))
11 alpha = (1/(4*t))*((z/k)^2)//cm^2/sec
12 //Results
13 printf("The thermal diffusivity is %.3f",alpha)//
    answer wrong in textbook
```

Scilab code Exa 20.3.1 Overall heat transfer coefficient

```
1
```

```

2  clc
3  //initialization of variables
4  Q = 18 // m^3/hr
5  z = 2.80 //m
6  T = 140 //C
7  T1 = 240 //C
8  T2 = 20 //C
9  p= 900 //kg/m^3
10 Cp = 2 // W/kg-K
11 d = 0.05 //m
12 // Calculations
13 A = %pi*(d^2)/4
14 v = Q*(1/(3600*40))/(A)
15 U = (v*p*Cp*d/(4*z))*(log((T1-T2)/(T1-T))) //W/m^2-K
16 DeltaT = ((T1-T2)+(T1-T))/2 //C
17 q = (Q*(1/(3600*40))*p*Cp/(%pi*d*z))*(T-T2) //W/m^2-K
18 U1 = q/DeltaT //W/m^2-K
19 // Results
20 printf("The overall heat transfer co efficient based
    on local temp difference is %.2f W/m^2-K",U)
21 printf("\nThe overall heat transfer co efficient
    based on average temp difference is %.2f W/m^2-K"
    ,U1)

```

Scilab code Exa 20.3.2 Time taken

```

1
2  clc
3  //initialization of variables
4  T = 32 //F
5  T0 = 10 //F
6  Tinf= 80 //F
7  U = 3.6 //Btu/hr-ft^2-F
8  A = 27 //ft^2
9  d = 8.31 //lb/gal

```



```

10 V = 100 //gal
11 Cv = 1//Btu/lb-F
12 //Calculations
13 t = (-log((T-T0)/(Tinf-T0)))*d*V*Cv/(U*A) //hr
14 //Results
15 printf("The time we can wait before the water in the
        tank starts to freeze is %.f hr",t)

```

Scilab code Exa 20.3.3 Savings calculation

```

1
2 clc
3 //initialization of variables
4 //Given q = h*DeltaT and 0.6q = (1/(1/h)+10/12*0.03)
    *delta T , divide both to get
5 l = 10/12 //ft
6 k = 0.03 //Btu/hr-ft-F
7 //Calculations
8 l2 = 2//feet
9 k2 = 0.03 //Btu/hr-ft-F
10 h = ((1/0.6)-1)*k/l //Btu/hr-ft^2-F
11 U = 1/((1/h)+(l2/k2)) //Btu/hr-ft^2-F
12 Savings = U*100/h
13 //Results
14 printf("The savings due to insulation is about %.f
        percent",Savings)

```

Scilab code Exa 20.4.1 Overall heat transfer coefficient

```

1
2 clc
3 //initialization of variables
4 T = 673 // Kelvin

```

```

5 M = 28
6 sigma = 3.80 // angstroms
7 omega = 0.87
8 d1 = 0.05 //m
9 v1 = 17 //m/sec
10 Mu1 = 3.3*10^-5 // kg/m-sec
11 p1 = 5.1*10^-1 // kg/m^3
12 Cp1 = 1100 // J/kg-K
13 k2 = 42 // W/m-K
14 l2 = 3*10^-3 //m
15 d3 = 0.044 //m
16 v3 = 270 //m/sec
17 p3 = 870 //kg/m^3
18 Mu3 = 5.3*10^-4 // kg/m-sec
19 Cp3 = 1700 // J/kg-K
20 k3 = 0.15 //W/m-K
21 // Calculations
22 kincal = (1.99*10^-4)*(sqrt(T/M))/((sigma^2)*omega)
    //W/m^2-K
23 k = kincal*4.2*10^2 // k in W/m-K
24 h1 = 0.33*(k/d1)*((d1*v1*p1/Mu1)^0.6)*((Mu1*Cp1/k)
    ^0.3) //W/m^2-K
25 h2 = k2/l2 //W/m^2-K
26 h3 = 0.027*(k3/d3)*((d3*v3*p3/Mu3)^0.8)*((Mu3*Cp3/k3)
    ^0.33) //W/m^2-K
27 U = 1/((1/h1)+(1/h2)+(1/h3)) //W/m^2-K
28 // Results
29 printf("The overall heat transfer co efficient is %.
    f W/m^2-K",U)

```

Scilab code Exa 20.4.2 heat transfer coefficient

```

1
2 clc
3 //initialization of variables

```

```

4 //For window with two panes 3 cm apart
5 k = 0.57*10^-4 //cal/cm-sec-K
6 l = 3 //cm
7 g = 980 // cm/sec^2
8 Nu = 0.14 // cm^2/sec
9 DeltaT = 30 // Kelvin
10 T = 278 // Kelvin
11 L = 100 // cm
12 //calculations
13 h = (0.065*(k/l)*(((l^3)*g*DeltaT/((Nu^2)*T))^(1/3))
      *((1/L)^(1/9)))*10^4//for two pane in x*10^-4 cal
      /cm^2-sec-K
14 printf("The heat transfer co efficient for two panes
      is %.2f x10^-4 cal/cm^2-sec-K",h)
15
16 //For window with three panes 1.5 cm each apart
17 k = 0.57*10^-4 //cal/cm-sec-K
18 l = 1.5//cm
19 DeltaT = 15 // Kelvin
20 g = 980 // cm/sec^2
21 Nu = 0.14 // cm^2/sec
22 //calculations
23 h = (0.065*(k/l)*(((l^3)*g*DeltaT/((Nu^2)*T))^(1/3))
      *((1/L)^(1/9)))*10^4//for two pane in x*10^-4 cal
      /cm^2-sec-K
24 printf("\nThe heat transfer co efficient for three
      panes is %.2f x10^-4 cal/cm^2-sec-K",h/2)//
      Because there are two gaps

```

Chapter 21

Simultaneous Heat and Mass Transfer

Scilab code Exa 21.1.2 Thermal diffusivity

```
1
2
3 clc
4 //initialization of variables
5 Tdisc = 30 // Centigrade
6 T = 21 // Centigrade
7 T0 = 18 // Centigrade
8 R0 = 1.5 // cm
9 V = 1000 // cc
10 t = 3600 //seconds
11 Nu = 0.082 //cm2/sec
12 omeg = 2*%pi*10/60 //sec-1
13 //Calculations
14 k = -V*(log((Tdisc-T)/(Tdisc-T0)))/(%pi*(R02)*t) //
    k = h/d*cp cm/sec
15 alpha = ((1/0.62)*(k)*(Nu(1/6))*(omeg(-0.5)))1.5
    // cm2/sec
16 //Results
17 printf("the value of thermal diffusivity is %.1e cm
```

^2/sec ", alpha)

Scilab code Exa 21.3.1 Time taken

```
1
2 clc
3 //initialization of variables
4 d =1000 // kg/m^3
5 h = 30 // W/m^2-C-sec
6 Hvap = 2300*10^3 // J/kg
7 T = 75 // C
8 Ti = 31 // C
9 l = 0.04 // m
10 epsilon = 0.36
11 c = 3600 // sec/hr
12 t1 = (Hvap/h)*(1/(T-Ti))*(l*epsilon*d) // sec
13 t = t1/c // in hr
14 //Results
15 printf("The time taken for drying is %.f hr",t)//
    answer wrong in textbook
```

Scilab code Exa 21.3.2 Mass transfer and time taken

```
1
2 clc
3 //initialization of variables
4 d = 100*10^-4 // cm
5 v = 10^-3 // cm/sec
6 nu = 0.2 // cm^2/sec
7 DS = 0.3 // cm^2/sec
8 DG = 3*10^-7 // cm^2/sec
9 H = 4.3*10^-4 // at 60 degree centigrade
10 //Calculations
```

```

11 kG = (2+(0.6*((d*v/nu)^0.5)*((nu/DS)^(1/3))))*DS/d//
    cm/sec
12 k = kG*H
13 t = 30*DG/k^2
14 //Results
15 printf("The mass transfer coefficient is %.5f cm/sec
    ",k)
16 printf("\nThe time needed to dry the particle is %.6
    f sec",t)
17 //Answer wrong in textbook starting from kG

```

Scilab code Exa 21.4.1 Flow rate area and length

```

1
2 clc
3 //initialization of variables
4 slope = 230 //J/g-mol-C
5 nair = 60 // gmol/cm^2-sec
6 CpH2O = 75 // J/gmol-C
7 f = 0.4 // Correction factor
8 F = 2150/(60*0.018) //gmol/m^2-sec
9 kc= 20/3
10 a = 3 // m^2/m^3
11 k = 2.7 // integral of dH/Hi-H with limits Hout and
    Hin
12 //Calculations
13 nH2Omax = slope*nair/CpH2O //gmol/m^2-sec
14 nH2O = nH2Omax*(1-f) //gmol/m^2-sec
15 A = F/nH2O // m^2
16 l = (nair/(kc*a))*k // m
17 //Results
18 printf("The flow rate of the water per tower cross
    section is %.f gmol H2O/m^2-sec",nH2O)
19 printf("\nThe area of tower cross section is %.f m^2
    ",A)

```

```
20 printf("\nThe length of the tower is %.1f m",l)
```

Scilab code Exa 21.5.1 Time taken

```
1
2 clc
3 //initialization of variables
4 A = 0.01 // cm^2
5 l = 1 // cm
6 VA = 3 // cc
7 VB = 3 // cc
8 alphagas = 0.29
9 alphaliquid = -1.3
10 x1 = 0.5
11 x2 = 0.5
12 deltaT = 50 // Kelvin Thot-Tcold = 50
13 Tavg = 298 // kelvin
14 Dgas = 0.3 // cm^2/sec
15 Dliquid = 10^-5 // cm^2/sec
16 //calculations
17 deltaY = alphagas*x1*x2*deltaT/Tavg // Y1hot-Y1cold
    = DeltaY
18 deltaX = alphaliquid*x1*x2*deltaT/Tavg// X1hot-
    X1cold = DeltaX
19 Beta = (A/l)*((1/VA)+(1/VB))//cm^-2
20 BetaDgasinverse = 1/(Beta*Dgas)// sec
21 BetaDliquidinverse = (1/(Beta*Dliquid))
    /(365*24*60*60)
22 //Results
23 printf("The seperation achieved for gas is %.3f",
    deltaY)
24 printf("\nThe seperation achieved for liquid is %.2
    f",deltaY)
25 printf("\nThe time taken for seperation for gas will
    be %.f seconds",BetaDgasinverse)
```

```
26 printf("\nThe time taken for seperation for liquid  
    will be %.1f year",BetaDliquidinverse)
```
