

Scilab Textbook Companion for
Elements Of Mass Transfer (Part 1)
by N. Anantharaman And K. M. M. S.
Begum¹

Created by
Himanshu Gupta
B. Tech
Chemical Engineering
IT BHU
College Teacher
R.S. Singh
Cross-Checked by

August 10, 2013

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: Elements Of Mass Transfer (Part 1)

Author: N. Anantharaman And K. M. M. S. Begum

Publisher: Prentice - Hall Of India, New Delhi

Edition: 1

Year: 2005

ISBN: 978812032783

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
2 Diffusion	7
3 Mass transfer coefficient and interphase mass transfer	41
5 Humidification	53
6 Drying	74
7 Crystallisation	101

List of Scilab Codes

Exa 2.1.a	diffusivity of nitrogen carbondioxide mixture	7
Exa 2.1.b	diffusivity of hydrogen chloride mixture	8
Exa 2.2	the diffusivity of isoamyl alcohol	9
Exa 2.3	diffusivity of ccl4 through oxygen	10
Exa 2.4	rate at which crystal dissolves	11
Exa 2.5.a	rate of diffusion of alcohol water vapour mixture	12
Exa 2.5.b	rate of diffusion if water layer is stagnant	13
Exa 2.6	rate of loss of hydrogen	14
Exa 2.7	ammonia diffusion through nitrogen	15
Exa 2.8	rate at which crystal dissolves	15
Exa 2.9	diffusion rate of acetic acid	17
Exa 2.10.a	rate of mass transfer	18
Exa 2.10.b	partial pressure of co2	19
Exa 2.11.a	diffusion flux if N2 is non diffusing	20
Exa 2.11.b	diffusion flux of oxygen	21
Exa 2.12	diffusion flux through inert air	22
Exa 2.13	diffusion rate of acetic acid	23
Exa 2.14	diffusion flux of nitrogen	25
Exa 2.15	diffusion rate of loss of benzene	26
Exa 2.16	rate of diffusion of alcohol water vapour mixture	27
Exa 2.17	diffusion rate of ammonia	28
Exa 2.18	rate of evaporation	30
Exa 2.19	diffusivity of the mixture in stefan tube of toluene in air	31
Exa 2.20	diffusion flux of a mixture of benzene and toluene	32
Exa 2.21	diffusivity of the mixture in stefan tube	33
Exa 2.22	diffusivity of ccl4	35
Exa 2.23	rate of transfer of nitrogen and hydrogen	36
Exa 2.24	diffusivity of methanol in carbon tetrachloride	37

Exa 2.25	diffusivity of methanol in water	38
Exa 2.26	rate of passage of hydrogen	39
Exa 3.1	rate of sublimation	41
Exa 3.2	rate of sublimation of solid naphthalene	42
Exa 3.3	rate of absorption and mass transfer coefficient	43
Exa 3.4	mass transfer coefficient and film thickness	44
Exa 3.5	value of liquid and gas film coefficient	45
Exa 3.6	concentration of ammonia and interfacial partial pressure	46
Exa 3.7	diffusivity of gas overall transfer coefficient	47
Exa 3.8	overall gas phase mass transfer flux	48
Exa 3.9	concentration of acid at outlet	50
Exa 5.1	properties through humidity chart	53
Exa 5.2	properties if DBT is 25 and WBT is 22	55
Exa 5.3	properties of nitrogen oxygen vapour mixture	57
Exa 5.4	properties at a temperature of 60 degree celcius	59
Exa 5.5	properties if DBT is 30 and WBT is 25	60
Exa 5.6	properties when dry bulb temperature is 55	61
Exa 5.7	Calculation of final temperature	62
Exa 5.8	Calculation of molal humidity	63
Exa 5.9	Calculation of relative humidity and humid volume	65
Exa 5.10	Calculation of film coefficient and make up water	67
Exa 5.11	Calculation of the make up water needed and the velocity of air	70
Exa 5.12	Calculation of the length of the chamber	72
Exa 6.1	Calculation of the solid temp	74
Exa 6.2	time for drying	75
Exa 6.3.a	plots of drying rate curve	78
Exa 6.3.b	total time for drying	79
Exa 6.4	time for drying the sheets	80
Exa 6.5.a	Calculation and plot of drying rates	82
Exa 6.5.b	amount of air required	84
Exa 6.5.c	actual time of the falling peroid	84
Exa 6.6	time saved in drying	85
Exa 6.7	time for drying the sheets from 30 to 6 percent	86
Exa 6.8.a	time for drying the material	87
Exa 6.8.b	time saved if air velocity is increased	88
Exa 6.9	time for drying the fibreboard	90
Exa 6.10	time for drying the moist material	91

Exa 6.11	time for drying the slab from 65 to 30 percent	92
Exa 6.12	time for drying the slab	93
Exa 6.13	time for drying the filter cake	94
Exa 6.14	time for drying the sheets	95
Exa 6.15	heater load per unit mass of dry air	96
Exa 6.16	surface area of the roller	98
Exa 6.17	time for drying the sheets	99
Exa 7.1	total weight of the solution	101
Exa 7.2	percentage saturation and yield	102
Exa 7.3	temperature to which solution should be cooled	103
Exa 7.4	percentage saturation and yield	104
Exa 7.5	the weight of Na ₂ SO ₄ hydrate crystal	105
Exa 7.6	percentage yield of Na ₂ CO ₃ hydrated crystal	106
Exa 7.7	percentage yield of K ₂ CO ₃ hydrated crystal	107
Exa 7.8	percentage yield FeSO ₄ hydrate crystal	108
Exa 7.9	percentage saturation and weight of Cesium chloride crystal	108
Exa 7.10	weight of Na ₂ CO ₃ hydrate needed to dissolve Na ₂ CO ₃	109
Exa 7.11	weight of Na ₂ CO ₃ hydrated formed	111
Exa 7.12	feed rate of FeSO ₄ hydrated crystal produced per hour	111
Exa 7.13	cooling water requirement	113

Chapter 2

Diffusion

Scilab code Exa 2.1.a diffusivity of nitrogen carbondioxide mixture

```
1
2 clear all;
3 clc;
4 printf("\t Example 2_1_a\n");
5 // let A denote nitrogen and B denote carboondioxide
6 rA=.3798;
7 rB=.3941;
8 rAB=(rA+rB)/2; //molecular
   seperation at collision
9 ebyk_A=71.4;
10 ebyk_B=195.2;
11 ebyk_AB=(ebyk_A/ebyk_B)^.5; //energy of
   molecular attraction
12 pt=1.013*10^5; //absolute
   total pressure in pascal
13 T=298; //absolute
   temperature in kelvin
14 s=T/ebyk_AB; //collision
   function
15 //from chart f(T/ebyk_AB) = 0.5 let it be = x
16 x=.5; //collision
```



```

function
17 MA=28; //molecular
    weight of nitrogen
18 MB=44; //molecular
    weight of carbondioxide
19 Mnew=((1/MA)+(1/MB))^.5;
20 Dab= 10^-4*(1.084-.249*(Mnew))*T^1.5*((Mnew))/(pt*x*
    rAB^2);
21 printf("\n the diffisivity of nitrogen-carbondioxide
    is :%f *10^-5 m^2/s",Dab/10^-5)
22
23 //end

```

Scilab code Exa 2.1.b diffusivity of hydrogen chloride mixture

```

1
2 clear all;
3 clc;
4 printf("\t Example 2_1_b\n");
5 // let A denote Hydrogen chloride and B denote air
6
7 //” part(ii)”
8 rA=.3339;
9 rB=.3711;
10 rAB=(rA+rB)/2; //molecular
    seperation at collision
11 ebyk_A=344.7;
12 ebyk_B=78.6;
13 ebyk_AB=(ebyk_A/ebyk_B)^.5; //energy of
    molecular attraction
14 pt=200*10^3; //absolute
    total pressure in pascal
15 T=298; //absolute
    temperature in kelvin
16 s=T/ebyk_AB; //collision

```

```

function
17 //from chart f(T/ebyk_AB) = 0.62 let it be = x
18 x=0.62; //collision
function
19 MA=36.5; //molecular
weight of hydrogen chloride
20 MB=29; //molecular
weight of air
21 Mnew=((1/MA)+(1/MB))^0.5;
22 Dab=10^-4*(1.084-.249*(Mnew))*T^1.5*((Mnew))/(pt*x*
rAB^2);
23 printf("\n the diffusivity of hydrogen chloride-air
is :%f *10^-6 m^2/s",Dab/10^-6)
24
25 //end

```

Scilab code Exa 2.2 the diffusivity of isoamyl alcohol

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.2\n");
5 //kopp's law is valid
6
7 u=1.145*10^-3; //viscosity of
water1.145cp
8 v_a=5*.0148+12*.0037+1*.0074; //by kopp's
law
9 t=288; //temperature
of water in kelvin
10 MB=18; //molecular
weight of water
11 phi=2.26; //association
parameter for solvent-water
12

```

```

13 D_ab=(117.3*10^-18)*((phi*MB)^.5)*(t)/(u*(v_a)^.6);
14 printf("\n the diffusivity of isoamyl alcohol is :%f
      *10^-9 m^2/s",D_ab/10^-9);
15 //end

```

Scilab code Exa 2.3 diffusivity of ccl4 through oxygen

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.3\n");
5
6 pa1=(33/760)*1.013*10^5;           //vapour
   pressure of ccl4 at 273 in pascal
7 pa2=0;
8 d=1.59;           //density of
   liquid ccl4 in g/cm^3
9 //considering o2 to be non diffusing and with
10 T=273;           //temperature in
   kelvin
11 pt=(755/780)*1.013*10^5;         //total pressure
   in pascal
12 z=.171;           //thickness of film
13 a=.82*10^-4;     //cross-sectional
   area of cell in m^2
14 v=.0208;         //volume of ccl4
   evaporated
15 t=10;           //time of evaporation
16 MB=154;         //molecular wght of
   ccl4
17 rate=v*d/(MB*t); //0.0208cc of ccl4 is
   evaporating in 10hrs
18 Na=rate*10^-3/(3600*a); //flux in kmol/m^2*S
19
20 D_ab=Na*z*8314*273/(pt*log((pt-pa2)/(pt-pa1)));

```

```

    //molecular diffusivity in m^2/s
21 printf("\n the diffusivity of ccl4 through oxygen:
    %f *10^-6 m^2/s", D_ab/10^-6);
22 //end

```

Scilab code Exa 2.4 rate at which crystal dissolves

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.4\n");
5 z=.0305*10^-3; //wall
    thickness surrounding the crystal
6 x1=0.0229;
7 w1=160; //molecular
    weight of copper sulphate
8 w2=18; //molecular weight
    of water
9 Dab=7.29*10^-10; //diffusivity of
    copper sulphate in m^2/s
10 //av=d/m
11 Mavg=x1*w1+(1-x1)*w2; //average molecular
    wght of solution
12 d1=1193; //density of copper
    sulphate solution
13 av1=d1/Mavg; //value of (d/m) of
    copper solution
14
15 //for pure water
16 d2=1000; //density of
    water
17 m2=18; //molecular wght
    of water
18 av2=d2/m2; //value of (d/m)
    of water

```

```

19 allavg=(av1+av2)/2;           //average value of
    d/m
20 xa2=0;
21 Na=Dab*(allavg)*log((1-xa2)/(1-x1))/z; //flux of
    cuso4 from crystal surface to bulk solution
22 printf("\n the rate at which crystal dissolves :%f
    *10^-5 kmol/m^2*s",Na/10^-5);
23 //end

```

Scilab code Exa 2.5.a rate of diffusion of alcohol water vapour mixture

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.5.a\n");
5 //position 1      moles      molefraction
6 //   air          80          0.8
7 //   water        20          0.2
8
9 //position 2      moles      molefraction
10 //   air          10          0.1
11 //   water        90          0.9
12 ya1=0.8;
13 ya2=0.1;
14 T=(273+35);           //temperature
    in kelvin
15 pt=1*1.013*10^5;      //total
    pressure in pascal
16 z=0.3*10^-3;         //gas film
    thickness in m
17 Dab=.18*10^-4;       //diffusion
    coefficient in m^2/s
18 R=8314;              //universal gas
    constant
19 Na=Dab*pt*(ya1-ya2)/(z*R*T) //diffusion flux

```

```

    in kmol/m2*s
20 rate=Na*100*10-4*3600*46;           //since molecular
    weight of mixture is 46
21 printf("\n rate of diffusion of alcohol-water vapour
    :%f kg/hr ",rate);
22
23 //end

```

Scilab code Exa 2.5.b rate of diffusion if water layer is stagnant

```

1
2 clear all;
3 clc;
4 printf("\t Example 2_5_b\n");
5 ya1=0.8;
6 ya2=0.1;
7 T=(273+35);           //temperature
    in kelvin
8 pt=1*1.013*105;           //total
    pressure in pascal
9 z=0.3*10-3;           //gas film
    thickness in m
10 Dab=.18*10-4;           //diffusion
    coefficient in m2/s
11 R=8314;           //universal gas
    constant
12
13 //diffusion through stagnant film
14 Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T); //
    diffusion flux in kmol/m2*s
15 rate=Na*100*10-4*3600*46;           //since
    molecular weight of mixture is 46
16 printf("\n rate of diffusion if water layer is
    stagnant :%f *10-3 kg/s ",rate/(3600*10-3));
17 //end

```

Scilab code Exa 2.6 rate of loss of hydrogen

```
1
2 clear all;
3 clc;
4 printf("\t Example 2.6\n");
5 T=298; //temperature in
kelvin
6 pt=1*1.013*10^5; //total pressure in
pascal
7 ID=25*10^-3; //internal diameter
in m of unvulcanised rubber in m
8 OD=50*10^-3; //internal diameter in
m of unvulcanised rubber in m
9 Ca1=2.37*10^-3; //conc. of hydrogen at
the inner surface of the pipe in kmol/m^3
10 Ca2=0; //conc. of
hydrogen at 2
11 Dab=1.8*10^-10; //diffusion
coefficient in cm^2/s
12 l=2; //length of pipe in
m
13 // Va=Da*Sa*(pa1-pa2)/z;
14 z=(50-25)*10^-3/2; //wall
thickness in m
15 Va=Dab*(Ca1-Ca2)/z; //diffusion
through a flat slab of thickness z
16 Sa=2*3.14*1*(OD-ID)/(2*log(OD/ID)); //average
mass transfer area of
17 rate=Va*Sa; //rate of loss
of hydrogen by diffusion
18 printf("\n rate of loss hydrogen by diffusion
through a pipe of 2m length :%f*10^-12kmol/s",
rate/10^-12);
```

19 //end

Scilab code Exa 2.7 ammonia diffusion through nitrogen

```
1
2 clear all;
3 clc;
4 printf("\t Example 2.7\n");
5
6 pa1=(1.5)*10^4;           //vapour pressure of
   ammonia at pt.1
7 pa2=(0.5)*10^4;         //vapour pressure of
   ammonia at pt.2
8 Dab=2.3*10^-5           //molecular
   diffusivity in m^2/s
9 z=0.15;                 //diffusion path in m
10 R=8314;                 //universal gas
   constant
11 //ammonia diffuses through nitrogen under
   equimolar counter diffusion
12 T=298;                  //temperature in
   kelvin
13 pt=1.013*10^5;         //total pressure in
   pascal
14 Na=Dab*(pa1-pa2)/(z*R*T); //flux in kmol/m^2*S
15 printf("\n the ammonia diffusion through nitrogen
   under equimolar counter diffusion:%f *10^-7 kmol/
   m^2*s",Na/10^-7);
16 //end
```

Scilab code Exa 2.8 rate at which crystal dissolves

1


```

2 clear all;
3 clc;
4 printf("\t Example 2.8\n");
5 //position 1      moles      molefraction      weight
6 //  ethanol      0.1478      0.02775      6.80
7 //  water        5.18        0.9722      93.20
8
9 //position 2      moles      molefraction      weight
10 //  ethanol      0.235      0.0453      10.8
11 //  water        4.96      0.9547      89.20
12 z=0.4*10^-2;      //film thickness
    surrounding the crystal
13 xa1=0.0453;      //mole fraction of
    ethanol at pos.2
14 xa2=0.02775;      //mole fraction of
    ethanol at pos.1
15 w1=46;      //molecular weight of
    ethanol
16 w2=18;      //molecular weight of water
17 Dab=74*10^-5*10^-4; //diffusivity of ethanol
    water sol.in m^2/s
18 //av=d/m
19 Mavg1=xa2*w1+(1-xa2)*w2; //average molecular
    wght of solution at pos 1
20 d1=0.9881*10^3; // density of 6.8 wt%
    solution
21 av1=d1/Mavg1; //value of (d/m) of
    copper solution
22
23 //for pure water
24 d2=972.8; //
    density of 10.8 wt% solution
25 Mavg2=xa1*w1+(1-xa1)*w2; //
    average molecular wght of solution at pos.2
26 av2=d2/Mavg2; //value
    of (d/m) of water
27
28 allavg=(av1+av2)/2; //average

```

```

        value of d/m
29 Na=Dab*(allavg)*log((1-xa2)/(1-xa1))/z; //steady
    state flux in kmol/m^2*s of ethanol water sol.
30 printf("\n the rate at which crystal dissolves :%f
    *10^-5 kmol/m^2*s",Na/10^-5);
31 //end

```

Scilab code Exa 2.9 diffusion rate of acetic acid

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.9\n");
5 //position 1      kmoles      molefraction
    weight
6 //  acetic acid  0.167      0.0323      10
7 //  water        5          0.9677      90
8
9 //position 2      kmoles      molefraction
    weight
10 //  acetic acid  0.067      0.0124      4
11 //  water        5.33      0.9876
    96
12
13 //basis : 100kg of mixture
14 z=2*10^-3; //film thickness
    sorrounding the water
15 xa1=0.0323; //mole fraction of
    ethanol at pos.2
16 xa2=0.0124; //mole fraction of
    ethanol at pos.1
17 w1=60; //molecular weight of
    acetic acid
18 w2=18; //molecular weight of
    water

```

```

19 Dab=0.000095;           //diffusivity of acetic
    water sol.in m^2/s
20 //av=d/m
21 Mavg1=xa1*w1+(1-xa1)*w2; //average molecular wght
    of solution at pos 1
22 d1=1013;               // density of 10 % acid
23 av1=d1/Mavg1;         //value of (d/m) of
    copper solution
24
25 //for pure water
26 d2=1004;               //density of 4% acid
27 Mavg2=xa2*w1+(1-xa2)*w2; //average molecular wght
    of solution at pos.2
28 av2=d2/Mavg2;         //value of (d/m) of
    water
29
30 allavg=(av1+av2)/2;    //average value of d/m
31 //assuming water to be non diffusing
32 Na=Dab*(allavg)*log((1-xa2)/(1-xa1))/z; //
    diffusion rate of acetic acid aacross film of non
    diffusing water sol.
33 printf("\n diffusion rate of acetic acid aacross
    film of non diffusing water sol. :%f kmol/m^2*s",
    Na);
34 //end

```

Scilab code Exa 2.10.a rate of mass transfer

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.10.a\n");
5 //part (i)
6 r=(50/2)*10^-3;        //radius pf circular tube
7 pa1=(190);             //vapour pressure of

```

```

    ammonia at pt.1
8  pa2=(95);           //vapour pressure of
    ammonia at pt.2
9  Dab=2.1*10^-5      //molecular diffusivity in
    m^2/s
10 z=1;
11 R=760*22.414/273;  //universal gas constant in
    mmHg*m^3*K*kmol
12    //carbondioxide and oxygen experiences equimolar
    counter diffusion
13 T=298;             //temperature in
    kelvin
14 pt=(10/780)*1.013*10^5; //total pressure in
    pascal
15 Na=Dab*(pa1-pa2)/(z*R*T); //flux in kmol/m^2*S
16 rate=Na*(3.14*r^2); //rate of mass
    transfer..(3.14*r^2)-is the area
17 printf("\n the rate of mass transfer.:%f *10^-10
    kmol/s",rate/10^-10);
18
19 //end

```

Scilab code Exa 2.10.b partial pressure of co2

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.10.b\n");
5 //part (i)
6 r=(50/2)*10^-3; //radius pf circular tube
7 pa1=(190); //vapour pressure of
    ammonia at pt.1
8 pa2=(95); //vapour pressure of
    ammonia at pt.2
9 Dab=2.1*10^-5 //molecular diffusivity in

```

```

    m^2/s
10 R=760*22.414/273;          //universal gas constant in
    mmHg*m^3*K*kmol
11    //carbondioxide and oxygen experiences equimolar
    counter diffusion
12 T=298;                    //temperature in
    kelvin
13 pt=(10/780)*1.013*10^5;   //total pressure in
    pascal
14
15    //part (ii)
16 //((ya-ya1)/(ya2-ya1)=(z-z1)/(z2-z1));
17 z2=1;                    //diffusion path in m
    at pos.2
18 z1=0;                    //diffusion path in m
    at pos.1
19 z=.75;                   //diffusion at general z
20 pa=poly([0], 'pa');     //calc. of conc. in gas
    phase
21 x=roots((pa-pa1)/(pa2-pa1)-(z-z1)/(z2-z1));
22 printf("\n partial pressure of co2 at 0.75m from the
    end where partial pressure is 190mmhg is:%f mmHg
    ",x);
23 //end

```

Scilab code Exa 2.11.a diffusion flux if N2 is non diffusing

```

1
2 clear all;
3 clc;
4 printf("\t Example 2_11_a\n");
5 ya1=0.2;                    //initial mole
    fraction
6 ya2=0.1;                    //final mole
    fraction

```

```

7 T=(298); //temperature
    in kelvin
8 pt=1*1.013*10^5; //total pressure
    in pascal
9 z=0.2*10^-2; //gas film
    thickness in m
10 Dab=.215*10^-4; //diffusion
    coefficient in m^2/s
11 R=8314; //universal gas
    constant
12 //part (i)when N2 is non diffusing
13
14 Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T); //
    diffusion flux in kmol/m^2*s
15 printf("\n diffusion flux if N2 is non diffusing :
    %f *10^-5 kmol/m^2*s ",Na/10^-5);
16
17 //end

```

Scilab code Exa 2.11.b diffusion flux of oxygen

```

1
2 clear all;
3 clc;
4 printf("\t Example 2_11_a\n");
5 ya1=0.2;
6 ya2=0.1;
7 T=(298); //temperature
    in kelvin
8 pt=1*1.013*10^5; //total pressure
    in pascal
9 z=0.2*10^-2; //gas film
    thickness in m
10 Dab=.215*10^-4; //diffusion
    coefficient in m^2/s

```

```

11 R=8314;                                //universal gas
    constant
12
13 //part (ii) equimolar counter diffusion
14
15 Na=Dab*pt*(ya1-ya2)/(z*R*T)            //
    diffusion flux in kmol/m^2*s
16 printf("\n diffusion flux of oxygen during
    equimolar counter-diffusion :%f *10^-5 kmol/m^2*s
    ",Na/10^-5);
17
18 //end

```

Scilab code Exa 2.12 diffusion flux through inert air

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.12\n");
5 // ammonia diffusing through inert air and air is
    non-diffusing
6 ya1=0.1;
7 ya2=0;
8 T=(293);                                //temperature
    in kelvin
9 pt=1*1.013*10^5;                        //total pressure
    in pascal
10 z=0.2*10^-2;                            //gas film
    thickness in m
11 Dab=.185*10^-4;                        //diffusion
    coefficient in m^2/s
12 R=8314;                                //universal gas
    constant
13 //part (i)when air is assumed to be stagnant and
    non-diffusing

```

```

14
15 Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T); //
    diffusion flux in kmol/m^2*s
16 mw=17; //
    molecular weight of ammonia
17 massflux=Na*mw; //mass flux
    of given NH3
18 printf("\n diffusion flux when total pressure is 1
    atm and air is non-diffusing :%f *10^-4 kg/m^2*s
    ",massflux/10^-4);
19 //part (ii) when pressure is increased to 10atm
20
21 //Dab_1/Dab_2=pt_2/pt_1
22 pt_2=10; //
    final pressure in atm
23 pt_1=1; //
    initially pressure was 1atm
24 Dab_1=.185; //
    initially diffusion coefficient was.185
25 Dab_2=Dab_1*pt_1/pt_2; //for
    gases Dab is proportional to 1/pt
26 Dab=Dab_2*10^-4; //new
    diffusion coefficient
27 pt=pt_2*1.013*10^5; //new
    total pressure
28 Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T); //
    diffusion flux in kmol/m^2*s
29 printf("\n diffusion flux when pressure is
    increased to 10atm :%f *10^-5 kmol/m^2*s ",Na
    /10^-5);
30 printf("\n \n so the rate of diffusion remains same
    on increasing the pressure");
31 //end

```

Scilab code Exa 2.13 diffusion rate of acetic acid


```

1
2 clear all;
3 clc;
4 printf("\t Example 2.13\n");
5 //position 1      moles      molefraction
   weight
6 //  acetic acid  0.15      0.0288      9
7 //  water        5        0.9712      91
8
9 //position 2      moles      molefraction
   weight
10 //  acetic acid  0.05      0.0092
    4
11 //  water        5.389     0.9908
    96
12 T=290;           //temperature in
   kelvin
13 z=2*10^-3;      //film thickness
   sorrounding the water
14 xa2=0.0092;     //mole fraction of
   ethanol at pos.2
15 xa1=0.0288;     //mole fraction of
   ethanol at pos.1
16 w1=60;          //molecular weight of
   acetic acid
17 w2=18;          //molecular weight of
   water
18 Dab=0.95*10^-9; //diffusivity of acetic
   water sol.in m^2/s
19 //av=d/m
20 Mavg1=xa1*w1+(1-xa1)*w2; //average molecular
   wght of solution at pos 1
21 d1=1012;        // density of 10 %
   acid
22 av1=d1/Mavg1;  //value of (d/m) of
   copper solution
23
24 //for position 2

```

```

25 d2=1003; //density of 4% acid
26 Mavg2=xa2*w1+(1-xa2)*w2; //average molecular wght
    of solution at pos.2
27 av2=d2/Mavg2; //value of (d/m) of
    water
28
29 allavg=(av1+av2)/2; //average value of d/m
30
31 //assuming water to be non diffusing
32
33 Na=Dab*(allavg)*log((1-xa2)/(1-xa1))/z; //
    diffusion rate of acetic acid aacross film of non
    diffusing water sol.
34 printf("\n diffusion rate of acetic acid aacross
    film of non diffusing water sol. :%f *10^-7 kmol/
    m^2*s",Na/10^-7);
35 //end

```

Scilab code Exa 2.14 diffusion flux of nitrogen

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.14\n");
5 ya1=0.2; //
    molefraction at pos.1
6 ya2=0.1; //
    molefraction at pos.2
7 T=(293); //temperature
    in kelvin
8 pt=1*1.013*10^5; //total
    pressure in pascal
9 z=0.2*10^-2; //gas film
    thickness in m
10 Dab=.206*10^-4; //diffusion

```

```

        coefficient in m^2/s
11 R=8314;                                //universal gas
        constant
12        //for ideal gases volume fraction =mole
        fraction
13 //part (i)when N2 is non diffusing
14
15 Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T); //
        diffusion flux in kmol/m^2*s
16 printf("\n diffusion flux if N2 is non diffusing :
        %f *10^-5 kmol/m^2*s ",Na/10^-5);
17 //part (ii) equimolar counter diffusion
18
19 Na=Dab*pt*(ya1-ya2)/(z*R*T)            //diffusion
        flux in kmol/m^2*s
20 printf("\n diffusion flux of nitrogen during
        equimolar counter-diffusion :%f *10^-5 kmol/m^2*s
        ",Na/10^-5);
21
22 //end

```

Scilab code Exa 2.15 diffusion rate of loss of benzene

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.15\n");
5 pa1=0.2*10^5;                            //partial pressure at pos
        .1
6 pa2=0;                                    //partial pressure at pos
        .2
7 r=10/2;                                    //radius of tank in which
        benzene is stored
8 T=(298);                                    //temperature in kelvin
9 pt=1*1.013*10^5;                          //total pressure in pascal

```

```

10 z=10*10^-3;           //gas film thickness in m
11 Dab=.02/3600;        //diffusion coefficient in
    m^2/s
12 R=8314;              //universal gas constant
13                       //benzene is stored in atank
    of dia 10m
14 //part (i)when air is assumed to be stagnant
15
16 Na=Dab*pt*log((pt-pa2)/(pt-pa1))/(z*R*T); //
    diffusion flux in kmol/m^2*s
17 rate=Na*(3.14*r^2); //rate of
    loss of benzene if air is stagnant
18 printf("\n diffusion rate of loss of benzene :%f
    *10^-4 kmol/s ",rate/10^-4);
19 //end

```

Scilab code Exa 2.16 rate of diffusion of alcohol water vapour mixture

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.16\n");
5 ya2=0.1; //molefraction
    at pos.2
6 ya1=0.8; //molefraction
    at pos.1
7 T=(370); //temperature
    in kelvin
8 pt=1*1.013*10^5; //total pressure
    in pascal
9 z=0.1*10^-3; //gas film
    thickness in m
10 Dab=.15*10^-2; //diffusion
    coefficient in m^2/s
11 R=8314; //universal gas

```

```

    constant
12 Area=10; //area of the film is
    10m^2
13
14 //alcohol is being absorbed infrom amixture
    of alcohol vapour and water vapour by
    means of non-volatile solvent in which
    alcohol is soluble bt water is not
15 //for gase Dab=T^3/2
16 //Dab1/Dab2=(T1/T2)^3/2
17
18 T2=370; //
    final temperature in kelvin
19 T1=298; //
    initial temperature in kelvin
20 Dab1=.15*10^-2; //
    initial diffusion coefficient
21 Dab2=((T2/T1)^(3/2))*Dab1; //final
    diffusion coefficient
22 Na=Dab2*pt*log((1-ya2)/(1-ya1))/(z*R*T); //
    diffusion flux in kmol/m^2*s
23 rate=Na*3600*46*Area; //rate of
    diffusion of alcohol-water vapour in kg/hour
24 printf("\n rate of diffusion of alcohol-water
    vapour :%f *10^6 kg/hour ",rate/10^6);
25
26 //end

```

Scilab code Exa 2.17 diffusion rate of ammonia

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.17\n");
5 ya2=0; //molefraction at pos

```

```

        .2
6 ya1=0.1; //molefraction at pos.1
7 T=(273); //temperature in kelvin
8 pt=1*1.013*10^5; //total pressure in
    pascal
9 z=2*10^-3; //gas film thickness in m
10 Dab=.198*10^-4; //diffusion coefficient in
    m^2/s
11 R=8314; //universal gas constant
12 //ammonia is diffusing through an inert film
    2mm thick
13
14 //for gase Dab=T^3/2
15 //Dab1/Dab2=(T1/T2)^3/2
16 T2=293; //final temperature
    in kelvin
17 T1=273; //initial temperature
    in kelvin
18 Dab1=0.198*10^-4; //initial diffusion
    coefficient
19 Dab2=((T2/T1)^(3/2))*Dab1; //final diffusion
    coefficient
20 Na=Dab2*pt*log((1-ya2)/(1-ya1))/(z*R*T2); //
    diffusion flux in kmol/m^2*s
21 printf("\n flux of diffusion of ammonia through
    inert film :%f *10^-5 kmol/m^2*s ",Na/10^-5);
22
23 //if pressure is also increased from 1 to 5 atm
24 //for gases Dab=(T^3/2)/pt;
25 //Dab1/Dab2=(T1/T2)^3/2*(p2/p1)
26 T2=293; //final temperature in
    kelvin
27 T1=273; //initial temperature in
    kelvin
28 pa2=5; //final pressure in atm
29 pa1=1; //initial pressure in atm
30 p=pa2*1.013*10^5;
31 Dab1=.198*10^-4; //

```

```

        initial diffusion coefficient
32 Dab2=((T2/T1)^(3/2))*Dab1*(pa1/pa2);           //final
        diffusion coefficient
33 Na=Dab2*p*log((1-ya2)/(1-ya1))/(z*R*T2);     //
        diffusion flux in kmol/m^2*s
34 printf("\n flux of diffusion of ammonia if temp. is
        20 and pressure is 5 atm  :%f*10^-5 kmol/m^2*s "
        ,Na/10^-5);
35 printf("\n \n so there is no change in flux when
        pressure is changed");
36 //end

```

Scilab code Exa 2.18 rate of evaporation

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.18\n");
5 pa1=0.418*10^5;           //partial pressure
        initially
6 pa2=0;                   //partial pressure of
        pure air
7 r=10/2;                  //radius of tank in
        which benzene is stored
8 T=(350);                 //temperature in kelvin
9 pt=1*1.013*10^5;        //total pressure in
        pascal
10 z=2*10^-3;              //gas film thickness in
        m
11 Dab=.2*10^-4;           //diffusion coefficient
        in m^2/s
12 R=8314;                 //universal gas constant
13 r=0.2/2;                //radius of open bowl is
        0.2
14 //when air layer is assumed to be stagnant of

```

```

    thickness 2mm
15
16 Na=Dab*pt*log((pt-pa2)/(pt-pa1))/(z*R*T);    //
    diffusion flux in kmol/m^2*s
17 rate=Na*(3.14*r^2)*18;                        //rate
    of loss of evaporation
18 printf("\n diffusion rate loss of evaporation :%f
    *10^-4 kg/s ",rate/10^-4);
19 //end

```

Scilab code Exa 2.19 diffusivity of the mixture in stefan tube of toluene in air

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.19\n");
5                                     //stefan tube experiment
6
7 M1=92;                               //molecular weight of
    toluene
8 T=(312.4);                           //temperature in
    kelvin
9 pt=1*1.013*10^5;                     //total pressure in
    pascal
10 R=8314;                              //universal gas
    constant
11 t=275*3600;                          //after 275 hours the
    level dropped to 80mm from the top
12 zo=20*10^-3;                         //intially liquid
    toluene is at 20mm from top
13 zt=80*10^-3;                         //finally liquid
    toluene is at 80mm from top
14 //air is assumed to be satgnant
15 d=850;                               //density in kg/m^3

```



```

16 pa=7.64*10^3;           //vapour pressure of
    toluene in at 39.4degree celcius
17 cal=d/M1;             //conc. at length at
    disxtance l
18 ca=pt/(R*T);         //total conc.
19 xa1=pa/pt;           //mole fraction of toluene
    at pt1 i.e before evaporation
20 xb1=1-xa1;           //mole fraction of air
    before evaporation i.e at pt1
21 xb2=1;               //mole fraction of air
    after evaporation i.e at pt.2
22 xa2=0;               //mole fraction of toluene
    at point 2
23 xbm=(xb2-xb1)/(log(xb2/xb1));
24 //t/(zt-zt0) = (xbm*cal*(zt+zo))/(2*c*(xa1-
    xa2)*t);
25 Dab=(xbm*cal*(zt^2-zo^2))/(2*ca*t*(xa1-xa2));
26 printf("\n the diffusivity of the mixture in stefan
    tube of toluene in air is :%f*10^-5 m^2/s",Dab
    /10^-5);
27 //end

```

Scilab code Exa 2.20 diffusion flux of a mixture of benzene and toluene

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.20\n");
5 //this is the case of
    equimolar counter diffusion
    as the latent heat of
    vaporisation are very close
    to each other
6
7 T=(360);           //temperature

```

```

    in kelvin
8  pt=372.4/760;           //total pressure
    in atm
9  R=82.06;               //universal gas
    constant
10 Dab=0.0506;           //diffusion
    coefficient in cm^2/s
11 z=0.254;              //gas layer
    thickness in cm
12 vp=368/760;           //vapour pressure
    of toluene in atm
13 xtol=.3;              //mole fractoin of
    toluene in atm
14 pb1=xtol*vp;          //partial pressure of
    toluene
15 //since pb1 is .045263 bt in book it is rounded to
    0.145
16 pb2=xtol*pt;          //parial pressure of
    toluene in vapour phase
17 Na=Dab*(pb1-pb2)/(z*R*T); //diffusion flux
18 printf("\n the diffusion flux of a mixture of
    benzene and toluene %f*10^-8 gmol/cm^2*s\n",Na
    /10^-8);
19 printf("\nthe negative sign indicates that the
    toluene is getting transferred from gas phase to
    liquid phase(hence the transfer of benzene is
    from liquid to gas phase)")
20 //end

```

Scilab code Exa 2.21 diffusivity of the mixture in stefan tube

```

1
2  clear all;
3  clc;
4  printf("\t Example 2.21\n");

```

```

5          //stefan tube experiment(pseudo
           steady state diffusion)
6
7 M1=92;          //molecular weight of
   toluene
8 T=(303);       //temperature in kelvin
9 pt=1*1.013*10^5; //total pressure in
   pascal
10 R=8314;       //universal gas constant
11 t=275*3600;   //after 275 hours the
   level dropped to 80mm from the top
12 zo=20*10^-3; //intially liquid toluene
   is at 20mm from top
13 zt=77.5*10^-3; //finally liquid toluene
   is at 80mm from top
14 //air is assumed to be satgnant
15 d=820;        //density in kg/m
   ^3
16 pa=(57/760)*1.0135*10^5; //vapour pressure
   of toluene in at 39.4degree celcius
17 cal=d/M1;     //conc. at length
   at disxtance l
18 ca=pt/(R*T); //total conc.
19 xa1=pa/pt;    //mole fraction of
   toluene at pt1 i.e before evaporation
20 xb1=1-xa1;    //mole fraction of air
   before evaporation i.e at pt1
21 xb2=1;        //mole fraction of air
   after evaporation i.e at pt.2
22 xa2=0;        //mole fraction of
   toluene at point 2
23 xbm=(xb2-xb1)/(log(xb2/xb1));
24 //t/(zt-zt0) = (xbm*cal*(zt+zo))/(2*c*(xa1-
   xa2)*t);
25 Dab=(xbm*cal*(zt^2-zo^2))/(2*ca*t*(xa1-xa2));
26 printf("\n the diffusivity of the mixture in stefan
   tube of toluene in air is :%f*10^-5 m^2/s",Dab
   /10^-5);

```

27 //end

Scilab code Exa 2.22 diffusivity of ccl4

```
1
2 clear all;
3 clc;
4 printf("\t Example 2.22\n");
5           //variation in liquid level with
           respect to time is given
           below
6
7 t=[26 185 456 1336 1958 2810 3829 4822 6385]
8 // let Zt-Zo= x;
9 x=[.25 1.29 2.32 4.39 5.47 6.70 7.38 9.03 10.48]
10 i=1;           //looping starts
11 while(i<10)
12     y(i)=t(i)/x(i);           //for calculating
           the t/Zt-Zo value
13 i=i+1;
14 end
15 plot(x,y,"o-");
16 xtitle(" Fig.2.2 Example 22 ", "X--(zi-zo),cm --->",
           "Y— vs (t/(zi-zo))min/cm ---->");
17 slope=51.4385*60 *10^4;           //slope of the
           curve in 1/sec*m^2
18 //slope = Cal *(xblm)/(2*Dab*C*(xa1-xa2))
19 d=1540;           //density in kg/m^3
20 M1=154;           //molecular weight of
           toluene
21 Cal=d/M1 ;           //conc. at length at
           disxtance l in mol/m^3
22
23 T=(321);           //temperature in
           kelvin
```

```

24 pt=1; //total pressure in atm
25 R=82.06; //universal gas constant
26 C=pt/(R*T) *10^3; //total conc. in kg
    mol/m^3
27
28 pa=(282/760); //vapour pressure of
    toluene
29 xa1=pa/pt; //mole fraction of
    toluene at pt1 i.e before evaporation
30 xb1=1-xa1; //mole fraction of air
    before evaporation i.e at pt1
31 xb2=1; //mole fraction of air
    after evaporation i.e at pt.2
32 xa2=0; //mole fraction of toluene
    at point 2
33 xblm=(xb2-xb1)/(log(xb2/xb1)); //log mean temp.
    difference
34 Dab = Cal *(xblm)/(2*slope*C*(xa1-xa2)); //
    diffusivity coefficient
35 printf("\n the diffusivity of the mixture by
    winklemann method of toluene in air is :%f*10^-6
    m^2/s",Dab/10^-6);
36 //end

```

Scilab code Exa 2.23 rate of transfer of nitrogen and hydrogen

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.23\n");
5 //it is the case of equimolar conter
    diffusion as the tube is
    perfectly sealed to two bulbs at
    the end and the pressure
    throughout is constant

```

```

6 d=0.001;
7 area=3.14*(d/2)^2; //area of the bulb
8 T=298; //temperature in kelvin
9 p=1.013*10^5; //total pressure of both the
bulbs
10 R=8314; //universal gas constant
11 c=p/(R*T); //total concentration
12 Dab=.784*10^-4; //diffusion coefficient in m^2/
s
13 xa1=0.8; //molefraction of nitrogen gas
at the 1 end
14 xa2=0.25; //molefraction of nitrogen gas
at the 2nd end
15 z=.15; //distance between the bulbs
16
17 //rate=area*Na;
18 rate=area*Dab*c*(xa1-xa2)/z; //rate of transfer
of hydrogen and hydrogen
19 printf("\n the rate of transfer from 1 to 2 of
nitrogen and 2 to 1of hydrogen is :%f *10^-11kmol
/s",rate/10^-11);
20 //end

```

Scilab code Exa 2.24 diffusivity of methanol in carbon tetrachloride

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.24\n");
5 //using wilke and chang empirical
correlation
6 //Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6);
7
8 T=288; //temperature in
kelvin

```

```

9 Mb=32; //molecular weight
  of methanol
10 phi=1.9; //association factor
  for solvent
11 va=(14.8+(4*24.6))*10^-3 //solute (CCl4)
  volume at normal BP in m^3/kmol
12 u=.6*10^-3; //viscosity of
  solution in kg/m*s
13 Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6); //
  diffusion coefficient in m^2/s
14 printf("\ndiffusivity of methanol in carbon
  tetrachloride is :%f*10^-9 m^2/s",Dab/10^-9);
15 //end

```

Scilab code Exa 2.25 diffusivity of methanol in water

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.25\n");
5 //using wilke and chang empirical
  correlation
6 //Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6);
7
8 T=288; //temperature in
  kelvin
9 Mb=18; //molecular weight
  of methanol
10 phi=2.26; //association
  factor for solvent
11 va=(2*14.8+(6*3.7)+7.4)*10^-3 //solute (water)
  volume at normal BP in m^3/kmol
12 u=1*10^-3; //viscosity of
  solution in kg/m*s
13 Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6); //

```

```

    diffusion coefficient in m^2/s
14 printf("\ndiffusivity of methanol in water is :%f
    *10^-9 m^2/s", Dab/10^-9);
15 //end

```

Scilab code Exa 2.26 rate of passage of hydrogen

```

1
2 clear all;
3 clc;
4 printf("\t Example 2.26\n");
5 u=20*10^-6; //viscosity in Ns
    /m^2
6 pt=2666; //total pressure
    in N/m^2
7 pa1=pt; //pressure at 1
8 pa2=0; //pressure at 2
9 mw=32; //molecular weight
    of oxygen
10 R=8314; //universal law
    constant
11 T=373; //temp. in kelvin
12 gc=1;
13 l=(3.2*u/pt)*((R*T)/(2*3.14*gc*mw))^0.5; //mean free
    path
14 d=.2*10^-6; //pore diameter
15 s=d/l; //value of dia/l
16 //hence knudsen diffusion occurs
17 Na=0.093*20*273/(760*373*22414*10^-1); //
    diffusion coefficient in kmol/m^2*s
18 Dka=(d/3)*((8*gc*R*T)/(3.14*mw))^0.5;
19 len=Dka*(pa1-pa2)/(R*T*Na); //length
    of the plate
20 printf("\n the length of the plate is :%f m ", len);
21

```



```

22
23 //for diffusion with hydrogen
24 u=8.5*10^-6; //viscosity in
    Ns/m^2
25 pt=1333; //total pressure
    in N/m^2
26 pa1=pt; //pressure at 1
27 pa2=0; //pressure at 2
28 mw=2; //molecular weight
    of oxygen
29 R=8314; //universal law
    constant
30 T=298; //temp. in kelvin
31 gc=1;
32 l=(3.2*u/pt)*((R*T)/(2*3.14*gc*mw))^0.5; //mean free
    path
33 d=.2*10^-6; //pore diameter
34 s=d/l; //value of dia/l
35 //hence knudsen diffusion occurs
36 Dka=(d/3)*((8*gc*R*T)/(3.14*mw))^0.5;
37 Na=Dka*(pa1-pa2)/(R*T*len); //
    diffusion coefficient in kmol/m^2*s
38 printf("\n the diffusion coefficient is :%f *10^-4
    kmol/m^2*s",Na/10^-6);
39 //end

```

Chapter 3

Mass transfer coefficient and interphase mass transfer

Scilab code Exa 3.1 rate of sublimation

```
1
2 clear all;
3 clc;
4 printf("\t Example 3.1\n");
5 v=6;          //velocity in m/s
6 l=6;          //length in m
7 pa1=10;       //pressure at 1 in atm
8 pa2=0;        //pressure at 2 in atm
9 t=373;        // temperature in kelvin
10 p=1;          //pressure of naphthalene in atm at 373
    kelvin
11 D=5.15*10^-6; //diffusivity of naphthalene in CO2
    in m^2/s
12 d=0.946;     //density of air in kg/m^3
13 u=.021*10^-3; //viscosity of air in Newton*s
    /m^2
14 ID=0.075;    //diameter in m
15 nre=(ID*v*d)/(u); //calc. of reynolds no.
16 cf=2*0.023*(nre)^(-0.2); //friction factor
```

```

17 nsc=(u)/(d*D);           //calc of schmidt no.
18 kc=(cf*v)/(2*(nsc)^(2/3));
19 na=(kc*10^5*(pa1/760-0))/(8314*t); //difussion
    flux in kmol/m^2*s
20 sub=na*2*3.14*(ID/2)*1; //rate of
    sublimation
21 printf("\nrate of sublimation :%f *10^-6 kmol/s\n",
    sub/10^-6);
22 //End

```

Scilab code Exa 3.2 rate of sublimation of solid naphthalene

```

1
2 clear all;
3 clc;
4 printf("\t Example 3.2\n");
5 v=0.30;           //velocity of parallelair in m/s
6 t=300;           //temperature of air in kelvin
7 p=10^5/760;      //pressure of air in
    pascal
8 Dab=5.9*10^-4;   //diffusivity of
    naphthalene in in air in m^2/s
9 pa1=0.2*10^5/760; //pressure of air at 1 in
    pascal
10 pa2=0;           //pressure of air at 2 in
    pascal
11 d=1.15;          //density of air in kg/m^3
12 u=0.0185*10^-3; //viscosity of air in Newton*
    s/m^2
13 D=1;             //length in m
14 a=1;             //area of plate in m^2
15 Nsc=u/(d*Dab);  //schmidt no. calculation
16 Nre=(D*v*d)/u;  //reynolds no. calculation
17 //flow is turbulent
18 f=0.072*(Nre)^-.25; //friction factor using "

```

```

    chilton colburn" analogy
19 k_c=(f*v)/(2*(Nsc)^.667);    //mass transfer
    coefficient
20 NA=k_c*(pa1-pa2)/(8314*300);    //mass flux calc.
21 sub=NA*a;    //rate of
    sublimation in kmol/m^2*s
22 printf("\nrate of sublimation :%f *10^-7 kmol/s\n",
    sub/10^-7);
23 //End

```

Scilab code Exa 3.3 rate of absorption and mass transfer coefficient

```

1
2 clear all;
3 clc;
4 printf("\t Example 3.3\n");
5     // a is CO2 and b is water
6 p=2;    //total pressure at 1 in atm
7 pa1=0.2*10^5;    //pressure of CO2 at pt 1 in atm
8 pa2=0;    //pressure of CO2 at pt 2 is 0
    since air is pure
9 ya1=0.1;    //mole fraction of CO2 at 1 is
    0.2/2
10 ya2=0;    //mole fraction of CO2 at 2 is 0
    since air is pure
11 yb1=0.9;    //mole fraction of water at 1 is
    (1-0.1)
12 yb2=1.0;    //mole fraction of water at 2 is
    1.0 since total pressure has to be constant.
13 k_y1=6.78*10^-5;    //mass transfer coefficient
    in kmol/m^2*s*molefraction
14
15 yb_ln=(yb2-yb1)/(log(yb2/yb1));    //log mean is
    represented by yb_ln
16

```

```

17 k_y=k_y1/yb_ln;
18 printf("\nvalue of mass transfer coefficient k_y is:
    %f *10^-5 kmol/m^2*s*(molefractin)",k_y/10^-5);
19 k_g=k_y/p; //mass ttransfer
    coefficient in lmol/m^2*s*atm
20 printf("\nvalue mass transfer coefficient k_g is:%f
    *10^-5 kmol/m^2*s*(atm)",k_g/10^-5);
21
22 NA=k_y*(ya1-ya2); //mass flux in kmol
    /m^2*s
23 printf("\nvalue of rate of mass transfer :%f *10^-6
    kmol/m^2*s",NA/10^-6);
24 //end

```

Scilab code Exa 3.4 mass transfer coefficient and film thickness

```

1
2 clear all;
3 clc;
4 printf("\t Example 3.4\n");
5 NA=7.5*10^-7; //mass flux in gmol/cm^2*s
6 Dab=1.7*10^-5; //diffusivity if SO2 in water in
    cm^2/s
7 c=1/18.02; //concentration is density/
    molecular weight in gmol/cm^2*s
8 //SO2 is absorbed from air into water
9
10 xa1=0.0025; //liquid phase mole fraction at 1
11 xa2=0.0003; //liquid phase mole fraction at 2
12 //NA=kc(Ca1-Ca2)=Dab*(Ca1-Ca2)/d
13
14 k_c=NA/(c*(xa1-xa2)); //k_c=Dab/d=NA/c(xa1-xa2)
15 printf("\nmass transfer coefficient k_c is:%f cm/s",
    k_c);
16

```

```

17 d=Dab/k_c;
18 printf("\nfilm thickness d is :%f cm",d);
19 //end

```

Scilab code Exa 3.5 value of liquid and gas film coefficient

```

1 clear all;
2 clc;
3 printf("\t Example 3.5\n");
4 Kg=2.72*10^-4; //overall gas phase mass
   transfer coefficient in kmol/m^2*S*atm
5 r_gas=0.85*(1/Kg); //given that gas phase
   resisitance is 0.85 times overall resistance
6 kg=1/r_gas;
7 m=9.35*10^-3; //henry's law constant in
   atm*m^3/kmol
8 kl=m/(1/Kg-1/kg); //liquid phase mass
   transfer coefficient in m/s
9 printf("\nthe value of liquid film coefficient kl :
   %f*10^-5 m/s",kl/10^-5);
10 printf("\nthe value of gas film coefficient kg : %f
   *10^-5 m/s",kg/10^-5);
11 p=1; //overall pressure in atm
12
13 //NA=Kg(pag-pa*)=kg(pag-pai)=kl(Cai-Cal)
14 Yag=0.1; //molefraction of ammonia
15 Cal=6.42*10^-2; //liquid phase concentration
16 Pag=Yag*p; //pressure of ammonia
17 //Pai and Cai indicates interfacial
   pressure and conc.
18 //Pal and Cal indicates bulk pressure and
   conc.
19
20 //Pai=m*C_ai;
21 //NA=kg(pag-pai)=kl(Cai-Cal)

```

```

22
23 Cai=poly([0], 'Cai');           //calc. of conc. in gas
    phase
24 x=roots((Pag-m*Cai)*(kg/kl)-(Cai-Cal));
25 printf("\nthe value of interphase conc.cai :%f kmol/
    m^3",x);
26 Pai=m*x;
27 printf("\nthe value of interphase pressure pai is:%f
    atm",Pai);
28 //end

```

Scilab code Exa 3.6 concentration of ammonia and interfacial partial pressure

```

1 clear all;
2 clc;
3 printf("\t Example 3.6\n");
4 //kg/kl=0.9=t
5 //Pai=0.3672*Cai    so;
6 m=0.3672;
7 t=.9;
8     //Pai and Cai indicates interfacial pressure
    and conc.
9     //Pal and Cal indicates bulk pressure and
    conc.
10 Yag=0.15;           //molefraction of ammonia
11 Cal=0.147;         //liquid phase concentration in
    kmol/m^3
12 p=1;               //overall pressure
13 Pag=Yag*p;         //pressure of ammonia
14
15     //Pai=m*C_ai;
16     //kg/kl=(Cai-Cal)/(Pag-Pai);
17
18 Cai=poly([0], 'Cai');           //calc. of conc. in gas

```

```

    phase
19 x=roots((Pag-m*cai)*(t)-(cai-cal));
20 printf("\nthe value of conc. of ammonia cai is :%f
    kmol/m^3",x);
21 Pai=m*x;
22 printf("\nthe value of interphase pressure Pai is :
    %f atm",Pai);

```

Scilab code Exa 3.7 diffusivity of gas overall transfer coefficient

```

1
2 clear all;
3 clc;
4 printf("\t Example 3.7\n");
5 D=.1;
6 l=3;           // l is length of bubble in cm
7 a=3.14*D*l;   // area in cm^2
8 Ca_o=0.0001;  //pure conc. of gas in g*mol/cc*atm
9 Ca=0;
10 NA=.482*10^-5; // molar rate of absorption in g*
    moles/s
11           //Pa_o and Ca_o indicates pure pressure and
    conc.
12 kl=NA/(a*(Ca_o-Ca)); //mass transfer coefficient
    acc. to higbie's penetration theory
13 Q=4;         //volumetric flow rate in cc
    /s
14 A=3.14*.1*.1/4; //area of flow
15 v=Q/A;       //velocity of flow in cm/s
16
17 //timt t=bubble length/linear velocity;
18 t=1/v;
19 DAB=(kl^2)*3.14*t/4; //diffusivity in cm^2/s
20 D_new=0.09;      //revised diameter reduced
    to.09

```



```

21 a_new=3.14*1*D_new;           //revised area
22 A_new=3.14*0.09*0.09/4;      //revised flow area
23 v_new=Q/A_new;               //revised velocity
24 printf("\nthe value of diffusivity of gas DAB is :%f
      cm/s",DAB/10^-5);
25
26 t_new=1/v_new;               //revised time
27 kl_new=2*(DAB/(3.14*0.0047))^0.5; //revised mass
      transfer coefficient
28 NA_new=kl_new*a_new*(Ca_o-Ca); //revised molar
      rate absorption in g*moles/s
29 printf("\nthe value of NA_new is :%f*10^-6 kmol/m^3
      ",NA_new/10^-6);
30 //end

```

Scilab code Exa 3.8 overall gas phase mass transfer flux

```

1
2 clear all;
3 clc;
4 printf("\t Example 3.8\n");
5 Kg=7.36*10^-10;
6 p=1.013*10^5;
7 Ky=Kg*p;
8 //resistance in gas phase is 0.45 of total
      resistance & .55 in liquid phase
9 //(resistance in gas phase)r_gas=1/ky and (
      resistance in liq phase)r_liq=m'/kx
10 r_gas=0.45*(1/Ky);
11 ky=1/r_gas;
12 r_liq=0.55*(1/Ky);
13 printf("\n film based liq phase mass transfer coeff.
      ky is :%f ",ky);
14 //from equilibrium relationship indicates linear
      behaviour thus the slope of equilibrium curve is

```

```

86.45
15 m1=86.45;
16 kx=m1/r_liq;
17 yag=.1;
18 xal=(.4/64)/((99.6/18)+(.4/64));
19 printf("\n film based gas phase mass transfer coeff.
      ky is      :%f ",kx);
20 //slope of the line gives -kx/ky=-70.61
21 m2=m1;          // since
      equilibrium line a straight line m'=m'
22 Kx=1/(1/kx+(1/(m2*ky)));          //overall liquid
      phase mass transfer coefficient
23 printf("\n overall liq phase mass transfer
      coefficient Kx is      :%f ",Kx);
24 // equilibrium relation is given under
25 p = [0.2 0.3 0.5 0.7];
26 a = [29 46 83 119];
27 i=1;
28 //looping for calculating mole fraction
29 while (i<5)
30     x(i)= (p(i)/64)/(p(i)/64+100/18);
31     y(i)= a(i)/760;          //mole
      fraction plotted on y-axis
32     i=i+1;          //mole
      fraction plotted on x-axis
33 end
34 plot(x,y,"o-");
35 title(" Fig.3.17,Example 8");
36 xlabel("X— Concentration of SO2 in liquid phase, X
      (10^4)(molefraction)");
37 ylabel("Y— Concentration of SO2 in gas phase, Y(
      molefraction)");
38
39 //from the graph we get these values
40 yao=.083;          //corresponding to the value of xao
      =0.001128
41 xao=.00132;          //corresponding to the value of yag
      =.1

```

```

42 yai=.0925;           //corresponding to the perpendicular
                        dropped from the pt(.001128,0.1)
43 xai=.00123;
44
45     // flux based on overall coefficient
46 NAO_gas=Ky*(yag-yao);
47 NAO_liq=Kx*(xao-xal);
48 printf("\n overall gas phase mass transfer flux -
        NAO_gas is   :%f*10^-6 kmol/m^2*s ",NAO_gas/10^-6)
        ;
49 printf("\n overall liq phase mass transfer flux -
        NAO_liq is   :%f*10^-6 kmol/m^2*s ",NAO_liq/10^-6)
        ;
50
51     // flux based on film coefficient
52 NAf_gas=ky*(yag-yai);
53 NAf_liq=kx*(xai-xal);
54 printf("\n film based gas phase mass transfer flux-
        NAf_gas is   :%f *10^-6 kmol/m^2*s",NAf_gas/10^-6)
        ;
55 printf("\n film based liq phase mass transfer flux-
        NAf_liq is   :%f *10^-6 kmol/m^2*s",NAf_liq/10^-6)
        ;
56 //end

```

Scilab code Exa 3.9 concentration of acid at outlet

```

1
2 clear all;
3 clc;
4 printf("\t Example 3.9\n");
5 Cas=1.521*10^-7;
6 v=1525;           //velocity in m/s
7 D=0.0516;        //diffusivity in cm^2/s
8 d=1.25*10^-3;    //density of air in g/cm^3

```

```

9 u=1.786*10^-4; //viscosity of air in n*s/m^2
10 Dia=2.54; //diameter in cm
11 nre=(Dia*v*d)/(u); //calc. of reynolds no.
12 cf=2*0.036*(nre)^(-0.25); //friction factor
13 nsc=(u)/(d*D); //calc of schmidt no.
14 kc=(cf*v)/(2*(nsc)^(2/3)); //cf/2=kc/uo*(sc)
    ^2/3
15
16 //consider an elemental section of dx at a distance
    of x from the point of entry of air.
17 //let the conc. be c of diffusing component and c+dc
    at the point of leaving. mass balance across
    this elemental gives
18 //rate of mass transfer=(cross
    sectional area)*(air velocity)*
    dc
19 // = (3.14*d
    ^2/4)*v*dc -----1 eqn
20
21 //flux for mass transfer from the surface=kc*(Cas-C)
22 // rate of mass transfer=(flux)*
    mass transfer
23 // =kc*(Cas-C)
    *3.14*dx*D-----2 eqn
24 // solving -----1 & -----2
    we get
25 //
26 // (3.14*d^2/4)*v*dc=kc*(
    Cas-C)*3.14*dx*d;
27 // dc/(Cas-C)=(kc*3.14*d*v
    )/(3.14*d^2/4)*dx
28 // solving this we get
29 // ln [(Cas-C)/(Cas-C_in)
    ]=(kc*4*x)/(d*v)
30
31 x=183; //upper limit of x
32 C_in=0; //C=C_in=0;
33 t=(kc*4*x)/(Dia*v); //variable to take out the

```

```
    exponential
34 z=%e^t;
35 C_final=Cas-(z*(Cas-C_in));           //value of
    c_final in g*mol/cc;
36 printf("\t conc. of acid at outlet   :%f *10^-8 g*
    mol/cc\n",abs(C_final/10^-8));
37 rate=(3.14*Dia^2/4)*v*(C_final-C_in);
38 printf("\trate of mass transfer   :%f *10^-4 g*mol/s\
    n",abs(rate/10^-4));
39 //End
```

Chapter 5

Humidification

Scilab code Exa 5.1 properties through humidity chart

```
1
2 clear all;
3 clc;
4 printf("\t Example 5.1\n");
5         //dry bulb temperature=50 and wet bulb
           temperature=35
6 Tg=50;           //dry bulb temperature=50
7 To=0;           //refrence temperature in
           degree celcius
8 Mb=28.84;       //average molecular weight
           of air
9 Ma=18;          //average molecular weight
           of water
10
11 //part(i)
12 ybar=.0483      //0.003 kg of water vapour
           /kg of dry air
13 printf("\n the humidity(from chart) is \t\t:%f
           percent",ybar);
14
15 //part(ii)
```

```

16 humper=35; //humidity percentage
17 printf("\n the percentage humidity is(from chart) :
    %f percent", humper);
18
19 //part(iii)
20 pt=1.013*10^5; //total pressure in pascal
21 molhum=0.0483; //molal humidity =pa/(pt-
    pa)
22 pa=molhum*pt/(1+molhum);
23 //the vopour pressure of water(steam tables)at 50
    degree = .1234*10^5 N/m^2
24 relhum=(pa/(.1234*10^5))*100; //percentage
    relative humidity =partial pressure/vapour
    pressure
25 printf("\n the percentage relative humidity is \t
    percent:%f ", relhum);
26
27 //part(iv)
28 dewpoint=31.5; //dew point temperature in
    degree celcius
29 printf("\n the dew point temperature \t\t :%f degree
    celcius", dewpoint);
30
31 //part(v)
32 Ca=1.005;
33 Cb=1.884;
34 ybar=.03; //saturation temperature
    inkg water vapour/kg dry air
35 Cs=Ca+Cb*ybar; //humid heat in kj/kg dry
    air degree celcius
36 printf("\n we get humid heat as \t\t\t :%f kj/kg dry
    air degree celcius ", Cs);
37
38 //part(vi)
39 d=2502; //latent heat in kj/kg
40 H=Cs*(Tg-0)+ybar*d; //enthalpy for refrence
    temperature of 0 degree
41 printf("\n we get H as \t\t\t\t :%f kj/kg", H);

```

```

42 Hsat=274; //enthalpy of sturated
    air
43 Hdry=50; //enthalpy of dry air in
    kj/kg
44 Hwet=Hdry+(Hsat-Hdry)*0.35; //enthalpy of wet
    air in kj/kg
45 printf("\n we get enthalpy of wet air as \t:%f kj/
    kg",Hwet);
46
47 //part(vii)
48 VH=8315*[(1/Mb)+(ybar/Ma)]*[(Tg+273)/pt]; //
    humid volume in m^3mixture/kg of dry air
49 printf("\n we get VH as (a)\t\t\t :%f m^3/kg of dry
    air",VH);
50 spvol=1.055; //specific volume of
    saturated air in m^3*kg
51 vdry=0.91; //specific volume of dry
    air in m^3/kg
52 Vh=vdry+(spvol-vdry)*.35 //by interpolation we get
    Vh in m^3/kg of dry air
53 printf("\n by interpolation we get specific volume
    Vh as(b) :%f m^3/kg of dry air",Vh);
54
55 //end

```

Scilab code Exa 5.2 properties if DBT is 25 and WBT is 22

```

1
2 clear all;
3 clc;
4 printf("\t Example 5.2\n");
5 //dry bulb temperature=25 and wet bulb
    temperature=22
6 Tg=25; //dry bulb temperature=50
7 To=0; //refrence temperature in

```



```

    degree celcius
8 Mb=28.84;           //average molecular weight
    of air
9 Ma=18;             //average molecular weight
    of water
10
11 //part(i)
12 hum=.0145         //0.0145 kg of water/kg of
    dry air
13 printf("\n the saturation humidity(from chart) is :
    %f percent",hum);
14
15 //part(ii)
16 humper=57;        //humidity percentage
17 printf("\n the percentage humidity is \t\t:%f
    percent",humper);
18
19 //part(iii)
20 pt=1;             //total pressure in atm
21 sathum=0.0255;    //molal humidity =pa/(pt-
    pa)
22 pa1=sathum*pt*(28.84/18)/(1+(sathum*(28.84/18)));
23 //the vopour pressure of water(steam tables)at 25 =
    .0393*10^5 N/m^2
24 pt=1;             //total pressure in atm
25 molhum=0.0145;    //molal humidity =pa/(pt-
    pa)
26 pa2=molhum*pt*(28.84/18)/(1+(molhum*pt*(28.84/18)));
27 //the vopour pressure of water(steam tables)at 25 =
    .0393*10^5 N/m^2
28 relhum=(pa2/pa1)*100; //percentage relative
    humidity =partial pressure/vapour pressure
29 printf("\n the percentage relative humidity is \t :
    %f ",relhum);
30
31 //part(iv)
32 dewpoint=19.5;    //dew point temperature in
    degree celcius

```

```

33 printf("\n the dew point temperature \t :%f degree
    celcius",dewpoint);
34
35 //part(v)
36 Ca=1005;
37 Cb=1884;
38 ybar=.0145;           // humidity inkg water /
    kg dry air
39 Cs=Ca+Cb*ybar;       //humid heat in j/kg dry
    air degree celcius
40 d=2502300;           //latent heat in j/kg
41 H=Cs*(Tg-0)+ybar*d;  //enthalpy for refrence
    temperature of 0 degree
42 printf("\n we get Humid heat H as \t :%f j/kg",H);
43 //the actual answer is 62091.3 bt in book it is
    given 65188.25(calculation mistake in book)
44 //end

```

Scilab code Exa 5.3 properties of nitrogen oxygen vapour mixture

```

1
2 clear all;
3 clc;
4 printf("\t Example 5.3\n");
5 //part(i)
6 pt=800;               //total pressure in mmHg
7 pa=190;               //vapour pressure of
    acetone at 25 degree
8 ys_bar=pa*(58/28)/(pt-pa) //
9 //percentage saturation = y_bar/ys_bar *100
10 s=80;                //percent saturation
11 y_bar=ys_bar*s/100;   //absolute humidity
12 printf("\n the absolute humidity is \t :%f kg
    acetone/kmol N2 ",y_bar);
13

```

```

14 //part(ii)
15 //y_bar=pa*(58/28)/(pt-pa)
16 pa1=pt*y_bar*(28/58)/(1+(y_bar*(28/58)));
17 printf("\n the partial pressure of acetone is:%f
    mmHg",pa1);
18
19 //part(iii)
20 y=pa1/(pt-pa1); //absolute molal
    humidity
21 printf("\n absolute molal humidity \t:%f kmol
    acetone/kmol N2",y);
22
23 //part(iv)
24 //volume of .249kmol acetone vapour at NTP
    =.249*22.14
25 //p1v1/T1 =p2v2/T2
26 p2=800; //final pressure of
    acetone and nitrogen at 25 degree
27 p1=760; //initial pressure of
    acetone and nitrogen at 25 degree
28 T2=298; //final temperature of
    acetone and nitrogen at 25 degree
29 T1=273; //initial temperature of
    acetone and nitrogen at 25 degree
30 vA1=5.581; //initial volume of
    acetone at 25 degree
31 vN1=22.414; //initial volume of
    nitrogen at 25 degree
32 vA2=T2*vA1*p1/(T1*p2); //final volume of
    acetone at 25 degree
33 vN2=T2*vN1*p1/(T1*p2); //final volume of
    nitrogen at 25 degree
34 vttotal=vA2+vN2; //total volume of the
    mixture
35 vper=vA2*100/vttotal; //percentage volume
    of acetone
36 printf("\n the percentage volume of acetone is :%f m
    ^3",vper);

```

37 //end

Scilab code Exa 5.4 properties at a temperature of 60 degree celcius

```
1
2 clear all;
3 clc;
4 printf("\t Example 5.4\n");
5
6 //part(i)
7 pa=13.3; //pressure in kpa
8 pa2=20.6; //vapour pressure at 60
   degree
9 pt=106.6 //total pressure in kpa
10 y=pa/(pt-pa); //absolute molal humidity
11 y_bar=y*(18/28.84); //relative humidity
12 printf("\n absolute humidity of mixture :%f kg
   water-vapour/kg dry air",y_bar);
13
14
15 //part(ii)
16 mf=pa/pt; //mole fraction
17 printf("\n the mole fraction is :%f",mf);
18
19 //part(iii)
20 vf=mf; //volume fraction
21 printf("\n the volume fraction is :%f",vf);
22
23 //part(iv)
24 Ma=18; //molecular weight
25 Mb=28.84; //molecular weight
26 Tg=60; //temperature of
   mixture
27 rh=(pa/pa2)*100; //relative humidity in
   pecentage
```

```

28 printf("\n we get relative humidity as as :%f
    percent",rh);
29
30 //part(v)
31 VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt]*10^-3;
    //humid volume in m^3mixture/kg of dry air
32 x=y_bar/VH; //g water/m^3 mixture
33 printf("\n we get x i.e. gwater/m^3 mixture as :%f "
    ,x*1000);
34 //end

```

Scilab code Exa 5.5 properties if DBT is 30 and WBT is 25

```

1
2 clear all;
3 clc;
4 printf("\t Example 5.5\n");
5
6 //part(i)
7 y_bar=.0183; //kg water vapour/kg dry
    air
8 printf("\n we get humidity as(from chart) :%f kg of
    water/kg dry air",y_bar);
9 printf("\n we get saturation humidity as(from chart)
    :%d percent",67);
10 Ma=18; //molecular weight
11 Mb=28.84; //molecular weight
12 Tg=30; //temperature of
    mixture
13 rh=(pa/pa2)*100; //relative humidity in
    pecentage
14 pt=1.013*10^5; //total pressure in pascal
15 VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt]; //
    humid volume in m^3mixture/kg of dry air
16 printf("\n we get humid volume as \t:%f m^3/kg dry

```

```

    air",VH);
17
18 //part(ii)
19 Ca=1005;
20 Cb=1884;
21 Cs=Ca+Cb*y_bar;          //humid heat in j/kg dry
    air degree celcius
22 printf("\n we get humid heat as \t\t :%f j/kg dry
    air degree celcius ",Cs);
23
24 //part(iii)
25 d=2502300;              //latent heat in j/kg
26 H=Cs*(Tg-0)+y_bar*d;    //enthalpy for refrence
    temperature of 0 degree
27 printf("\n we get Enthalpy H as \t\t:%f j/kg dry air
    ",H);
28
29 //part(iv)
30 dewpoint=23.5;          //dew point temperature in
    degree celcius
31 printf("\n the dew point temperature \t :%f degree
    celcius",dewpoint);
32
33 //end

```

Scilab code Exa 5.6 properties when dry bulb temperature is 55

```

1
2 clear all;
3 clc;
4 printf("\t Example 5.6\n");
5
6 //part(i)
7 y=.048;                  //humidity kmol water
    vapour/kmol dry air

```

```

8  y_bar=y*(18/28.84);          //(from chart) absolute
   humidity
9  printf("\n we get absolute humidity as  :%f kg of
   water/kg dry air",y_bar);
10 printf("\n we get percentage humidity as(from chart)
   :%f percent",25.5);
11 y_bar=y*(18/28.84);          //relative humidity
12 Ma=18;                       //molecular weight
13 Mb=28.84;                    //molecular weight
14 Tg=55;                       //temperature of
   mixture
15 pt=1.013*10^5;              //total pressure in pascal
16 VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt];          //
   humid volume in m^3mixture/kg of dry air
17 printf("\n we get VH  as \t :%f m^3/kg dry air",VH);
18
19 //part(ii)
20 Ca=1005;
21 Cb=1884;
22 Cs=Ca+Cb*y_bar;             //humid heat  in j/kg dry
   air degree celcius
23 printf("\n we get humid heat as \t :%f j/kg dry air
   degree celcius ",Cs);
24
25 //part(iii)
26 d=2502300;                  //latent heat in j/kg
27 H=Cs*(Tg-0)+y_bar*d;        //enthalpy for refrence
   temperature of 0 degree
28 printf("\n we get H as \t :%f j/kg dry air",H);
29
30 //end

```

Scilab code Exa 5.7 Calculation of final temperature

1

```

2 clear all;
3 clc;
4 printf("\t Example 5.7\n");
5     //given 0.03 kg of water vapour/kg of dry
        air is contacted with water at an
        adiabatic temperature and humidified and
        cooled to 70 percent saturation
6
7 //from psychometric chart
8 ft=46;           //final temperature in degree
        celcius
9 printf("\n final temperature is (from chart):%f
        degree celcius",ft);
10 y_bar=.0475;    // humidity of air
11 printf("\n the humidity of air(from chart) :%f kg
        water vapour /kg dry air",y_bar);
12
13 //end

```

Scilab code Exa 5.8 Calculation of molal humidity

```

1
2 clear all;
3 clc;
4 printf("\t Example 5.8\n");
5 pa1=4.24         //data:   vapour pressure of
        water at 30degree = 4.24 kpa
6 pa2=1.70         //         vapour pressure of
        water at 30degree = 1.70 kpa
7
8 //part(i)
9 pt=100;          //total pressure
10 ys_bar=pa1/(pt-pa1); //kg water vapour/
        kg dry air
11 rh=.8;          //relative humidity

```



```

12 pa3=rh*pa1; //partial pressure
13 y_bar=pa3*(18/28.84)/(pt-pa3); //molal
    humidity
14 printf("\n the molal humidity:%f kg/kg dry air",
    y_bar);
15
16 //part(ii)
17 //under these conditions the air will be saturated
    at 15 degree as some water is condensed
18 pa=1.7;
19 pt=200;
20 ys=pa/(pt-pa);
21 ys_bar=ys*(18/28.84);
22 printf("\n the molal humidity if pressure doubled
    and temp. is 15 :%f kg/kg dry air",ys_bar);
23
24 //part(iii)
25 Ma=18; //molecular weight
26 Mb=28.84; //molecular weight
27 Tg=30; //temperature of
    mixture
28 rh=(pa/pa2)*100; //relative humidity in
    pecentage
29 pt=10^5; //total pressure in pascal
30 VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt]; //
    humid volume in m^3mixture/kg of dry air
31 printf("\n we get humid volume VH as \t :%f m^3/kg
    of dry air",VH);
32 w=100/VH; //100 m^3 of original
    air
33 wo= w*y_bar; //water present in
    original air
34 wf= w*ys_bar; //water present finally
35 wc=wo-wf; //water condensed from
    100m^3 of original sample
36 printf("\n the weight water condensed from 100m^3 of
    original sample:%f kg",wc);
37

```

```

38 //part(iv)
39 Tg=15; //temperature of
    mixture
40 pt=2*10^5; //total pressure in pascal
41 VH=8315*[(1/Mb)+(ys_bar/Ma)]*[(Tg+273)/pt];
    //humid volume in m^3mixture/kg of dry air
42 vf=VH*110.6; //final volume of
    mixture
43 printf("\n we get VH final volume of mixture as \t :
    %f m^3",vf);
44
45 //end

```

Scilab code Exa 5.9 Calculation of relative humidity and humid volume

```

1 // Calculation of relative humidity ,humid volume
    enthalpy and heat required if 100m^3 of this air
    is heated to 110 degree
2 clear all;
3 clc;
4 printf("\t Example 5.9\n");
5
6 //part(i)
7 y_bar=.03; // humidity inkg water
    /kg dry air
8 pt=760; //total pressure in
    pascal
9 pa2=118; //final pressure
10 y=y_bar/(18/28.84); //humidity kmol water
    vapour/kmol dry air
11 pa=(y*pt)/(y+1); //partial pressure
12 rh=pa/pa2; //relative humidity
13 sh=pa2/(pt-pa2); //saturated humidity
14 ph=(y/sh)*100; //percentage humidity
15 printf("\n percentage humidity is :%f",ph);

```

```

16
17 ///part(ii)
18 Ma=18; //molecular weight
19 Mb=28.84; //molecular weight
20 Tg=55; //temperature of
    mixture
21 pt=1.013*10^5; //total pressure in pascal
22 VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt]; //
    humid volume in m^3mixture/kg of dry air
23 printf("\n we get VH humid volume as :%f m^3/kg dry
    air",VH);
24
25
26 //part(iii)
27 Ca=1005;
28 Cb=1884;
29 Cs=Ca+Cb*y_bar; //humid heat in j/kg
    dry air degree celcius
30 printf("\n we get humid heat as \t :%f j/kg dry air
    degree celcius ",Cs);
31 d=2502300; //latent heat in j/kg
32 H=Cs*(Tg-0)+y_bar*d; //enthalpy for refrence
    temperature of 0 degree
33 printf("\n we get H enthalpy as \t :%f j/kg",H);
34
35 //part(iv)
36 v=100; //volume of air
37 mass=v/VH; //mass of dry air
38 Tg=110; //temperature of
    mixture
39 d=2502300; //latent heat in j
40 H_final=Cs*(Tg-0)+y_bar*d; //enthalpy for
    refrence temperature of 0 degree
41 H_added=(H_final-H)*102.25; //HEAT added in kj
42 printf("\n we get heat added as \t :%f kj",H_added
    /1000);
43 //end

```

Scilab code Exa 5.10 Calculation of film coefficient and make up water

```
1
2 clear all;
3 clc;
4 printf("\t Example 5.10\n");
5 L=2000; //flow rate of water to
   be cooled in kg/min
6 T1=50; //temperature of inlet
   water
7 T2=30; //temp. of outlet water
8 H1=.016; //humidity of incoming air
9 cp=4.18; //specific heat of water
10 cpair=1.005; //specific heat capacity of
   air
11 cpwater=1.884; //specific heat capacity of
   water
12 tg=20; //temperature in degree
13 to=0;
14 ybar=0.016; //saturated humidity at 20
   degree
15 d=2502; //latent heat
16 Ky_a=2500; //value of masstransfer
   coefficient in kg/hr*m^3*dybar
17 E=cpair*(tg-to)+(cpwater*(tg-to)+d)*ybar; //
   enthalpy
18 //similarly for other temperatures
19 T=[20 30 40 50 55] //differnt temperature
   for different enthalpy calculation
20 i=1;
21 while(i<6) //looping for different
   enthalpy calculation of operating line
22 E(i)=cpair*(T(i)-to)+(cpwater*(T(i)-to)+d)*ybar;
23 printf("\n the enhalpy at :%f is :%f",T(i),E(i));
```

```

24 i=i+1;
25 end //end of lop
26 ES=[60.735 101.79 166.49 278.72 354.92] //
    enthalpy of eqll condition
27
28 plot(T,E,"o—");
29 plot(T,ES,"+—");
30 title("Fig.5.10(b),Temperature-Enthalpy plot");
31 xlabel("X— Temperature, degree celcius");
32 ylabel("Y— Enthalpy ,kj/kg");
33 legend("operating line","Enthalpy at saturated cond"
    )
34
35 //locate (30,71.09) the operating conditions at the
    bottom of the tower and draw the tangent to the
    curve
36 Hg1=71.09; //point on the oper.
    line(incoming air)
37 Hg2=253; //point after drawing
    the tangent
38 slope=(Hg2-Hg1)/(T1-T2); //we gt slope of the
    tangent
39 //slope = (L*Cl/G)_min
40 Cl=4.18;
41 G_min=L*60*Cl/slope; //tangent
    gives minimum value of the gas flow rate
42 G_actual=G_min*1.3; //since actual
    flow rate is 1.3 times the minimum
43 slope2=L*Cl*60/G_actual; //slope of
    operating line
44 Hg2_actual=slope2*(T1-T2)+Hg1; //actual
    humidityat pt 2
45 Ggas=10000; //minimum gas rate
    in kg/hr*m^2
46 Area1=G_actual/Ggas; //maximum area of
    the tower(based on gas)
47 Gliq=12000; //minimum liquid
    rate in kg/hr*m^2

```

```

48 Area2=60*L/Gliq; //maximum area of the
    tower(based on liquid)
49 printf("\n \n the maximum area of the tower(based on
    gas) is :%f m^2",Area1);
50 printf("\n the maximum area of the tower(based on
    liquid) is :%f m^2",Area2);
51 dia=(Area1*4/3.14)^0.5; //diameter of the
    tower in m
52
53 //let us assume the resistance to mass transfer lies
    basically in gas phase. hence the,interfacial
    conditions and the eqlb cond. are same.vertical
    line drawn between oper. and equil. line we get
    conditions of gas and equil. values are tabulated
    below as follows
54
55
56 //table
57 T=[20 30 40 50 55] //differnt temperature
    for different enthalpy calculation
58 //enthalpy
59 H_bar=[101.79 133.0 166.49 210.0 278.72] //
    H_bar i.e. at equil.
60 Hg=[71.09 103.00 140.00 173.00 211.09] //Hg i
    .e. of operating line
61 i=1;
62 while(i<6) //looping for different enthalpy
    calculation of operating line
63 y(i)=1/(H_bar(i)-Hg(i));
64 printf("\n the enhalpy at :%f is :%f",T(i),y(i));
65 i=i+1;
66 end //end of lop
67 xset('window',1);
68 plot(Hg,y,"o-");
69 xtitle(" Fig.5.10(c) Example 10 (1/(Hf-Hg)) vs Hg",
    "X— Hg —>", "Y— 1/(Hf-Hg) —>");
70
71 //area under this curve gives Ntog =4.26

```

```

72 Ntog=4.26;           //no. of transfer unit
73 Gs=10000;          //gas flow rate
74 Htog=Gs/Ky_a;      // height of transfer
    unit
75 height=Ntog*Htog;   //height of the tower
76 printf("\n \nthe tower height is :%f m",height);
77
78
79 //make up water is based onevaporation loss (E),blow
    down loss (B),windage loss (W)
                                M = E + B + W
80 W=.2/100 *L*60;    //windage loss (W)
81 B=0;               //blow down loss
    neglected
82 E=G_actual*(.064-.016); //assuming air leaves
    fully saturated
83 M = E + B + W;    //make up water is
    based onevaporation loss (E),blow down loss (B),
    windage loss (W)
84 printf("\n make up water is based onevaporation loss
    (E),blow down loss (B),windage loss (W) is :%f kg /
    hr",M);
85 //end

```

Scilab code Exa 5.11 Calculation of the make up water needed and the velocity of air

```

1
2 clear all;
3 clc;
4 printf("\t Example 5.11\n");
5 //air leaves at 19 degree at fully saturated
    condition
6
7 T1=30;              //temperature at the inlet in

```

```

    degree celcius
8  T2=17;           //temperature at the exit in
    degree celcius
9  f=100000;       //flow rate of water in kg/hr
10 hi=.004;        //humidity of incoming air in kg
    /kg of dry air
11 hl=.015;        //humidity of leaving air in kg/
    kg of dry air
12 Hi=18.11;       //enthalpy of incoming air in kg
    /kg of dry air
13 Hl=57.16;       //enthalpy of leaving air in kg
    /kg of dry air
14 //w=mdry*(hl-hi) = mdry*0.011;
                                -----equn 1st
15 //mass of water evaporated
16
17 //making energy balance:  total heat in = total heat
    out
18 //heat in entering water + heat in entering air =
    heat in leaving water + heat in leaving air
19 //100000*1*(30-0) + mdry*Hi = (100000-w)*1*(17-0) +
    mdry*Hl  -----eqn 2nd
20
21 //substituting eqn 1st in 2nd we get;
22 a=14.4;          //cross
    sectional area of the tower in m^2
23 mdry=(T1*f-T2*f)/(Hl-Hi-T2*.011); //mass of
    dry air
24 velocity=mdry/a; //air
    velocity in kg/m^2* hr
25 x=mdry*.011;    //make up
    water needed in kg/hr
26 printf("\n the make up water needed is :%f kg /hr",x
    );
27 printf("\n the velocity of air is as :%f kg/hr",
    velocity);
28 //end

```

Scilab code Exa 5.12 Calculation of the length of the chamber

```
1
2 clear all;
3 clc;
4 printf("\t Example 5.12\n");
5 //horizontal spray with recirculated water . air is
   cooled and humidified to 34 and leaves at 90
   percent saturation
6
7 T1=65;           //dry bulb temperature at the
   inlet in degree celcius
8 f=3.5;          //flow rate of air in m^3/s
9 hi=1.017;       //humidity of incoming air in
   kg/kg of dry air
10 hl=.03;        //humidity of leaving air in kg/
   kg of dry air
11 k=1.12;        //mass transfer coefficient in kg
   /m^3*s
12 y1=.017;       //molefraction at recieving end
13 y2=.03;        //molefraction at leaving end
14
15 //substituting eqn 1st in 2nd we get;
16 a=2;           //cross
   sectional area of the tower in m^2
17 d=1.113;       //density
   o fair in kg/m^3
18 m=(f*d)        //mass
   flow rate of air
19 gs=m/hi;       //air
   velocity in kg/m^2* hr
20 ys_bar=.032;
21 //for recirculation humidifier
22 z=log((ys_bar-y1)/(ys_bar-y2))*gs/k; //
```

```
    length of the chamber required
23 printf("\n the length of the chamber required is :%f
    m",z);
24
25 //end
```

Chapter 6

Drying

Scilab code Exa 6.1 Calculation of the solid temp

```
1
2 clear all;
3 clc;
4 printf("\t Example 6.1\n");
5
6 //air leaves the pre-heater of the dryer at 325K
7
8 H1=.005; //humidity of
incoming air per kg of dry air
9 T1=25; //wet bulb
temperature
10 //moisture is removed along constant wet bulb
temp. till 60per R.H is reached
11 // from the chart ,humidity of ai rleaving first
shelf =.016 kg water /kg dry air.
12
13 //dry bulb temp. of exit air is at 27 degree aand is
at humidity of .016 kg water/kg dry air.the air
is again heated to 52 degree dry bulb temp. in 2
nd heater .
14
```

```

15 //so air leaves heater at 52 degree and humidity
    of .016 kg water/kg dry air. when it leaves
    the 2nd shelf the correspondin dry bulb temp.
    is 34 degree and humidity is .023 kg water/
    kg dry air. the air enters the 3rd shelf
    after preheating to 52 degree .
16
17 //similarly fro 3rd shelf , exit air has a humidity
    of .028 kg water/kg dry air and adry bulb temp.
    is 39 degree. the air is leaving the 4rth shelf
    has a humidity of .016 kg water/kg dry air and
    adry bulb temp. of 42 degree(the figure is only
    indicative and does not correspond toactual one)
18
19 printf("\n the solid temp. correspond to wbt and
    they are 23, 27,32 and 34 degree respectively");
20
21 //part(ii)
22 Ybar=.032; //kg water/kg dry air//
    final moist air condotions
23 T2=42; //dry bulb temperature
24 Mair=28.84; //molecular weight of air
25 Mwater=18; //molecular weight of
    water
26 pt=1.013*10^5; //total pressure in pascal
27 Vh=8315*((1/Mair)+(Ybar/Mwater))*((T2+273)/pt);
28 r=300; //flow rate of moist air
    leaving the dryer
29 a=r*60/Vh; //amount of dry air leaving
    /hr
30 w=a*(Ybar-0.005); // water removed /hr
31 printf("\n the water removed /hr is :%fkg /hr",w);
32
33 //end

```

Scilab code Exa 6.2 time for drying

```
1
2 clear all;
3 clc;
4 printf("\t Example 6.2\n");
5 //table      X*100,(kgmoisture/kg dry solid)      N*100 (
      kg moisture evaporated /hr*m^2)
6
7 //              35
8 //              25
9 //              20
10 //             18
11 //             16
12 //             14
13 //             12
14 //             10
15 //              9
16 //              8
17 //              7
18 //             6.4
19
20
21 Ls=262.5; //mass of bone dry solid
      ais the drying surface
22 A=262.5/8; //both upper surafce and
```

```

    lower surface are exposed
23 Nc=0.3; //in kg/m^2*hr
24 x2=.06; //moisture content on wet
    basis finally after drying
25 x1=.25; //moisture content on wet
    basis finally after drying
26 Xcr=0.20; //critical moisture content
27 X1=x1/(1-x1); //moisture content
    on dry basis intially
28 X2=x2/(1-x2); //moisture content
    on dry basis finally after drying
29 Xbar=0.025; //equillibrium
    moisture
30
31 t1=Ls/(A*Nc) *(X1-Xcr); //so for constant rate
    period
32
33 //for falling rate period we find time graphically
34 p = [.20 .18 .16 .14 .12 .10 .09 .08 .07 .064];
35 a = [3.3 5.56 6.25 7.14 8.32 10.00 11.11 12.5 14.29
    15.625];
36
37 plot(p,a,"o-");
38 title("Fig.6.18 Example2 1/N vs X for fallling rate
    period");
39 xlabel("X— Moisture content , X(kg/kg)");
40 ylabel("Y— 1/N, hr,m^2/kg");
41
42 Area=1.116; //area under the curve
43 t2=Area *Ls/A; //falling rate period we
    find time graphically
44 tttotal=t1+t2; //total time for drying
45 printf("\n the total time for drying the wet slab on
    wet basis is :%f min",tttotal);
46 //end

```

Scilab code Exa 6.3.a plots of drying rate curve

```

1
2 clear all;
3 clc;
4 printf("\t Example 6_3_a\n");
5 //part(i)
6 //table wt of wet slab ,kg — 5.0 4.0 3.6 3.5
7 // 3.4 3.06 2.85
8 // drying rate ,kg/m^2s— 5.0 5.0 4.5 4.0
9 // 3.5 2.00 1.00
10 // X,Dry basis — 1.0 0.6 .44 0.4
11 // .36 .224 0.14
12 // equilibrium relation is given under
13 p = [1.0 0.6 .44 0.4 .36 .224 0.14];
14 a = [5.0 5.0 4.5 4.0 3.5 2.00 1.00];
15
16 i=1; //looping for calc.
17 of 1/N
18 while(i<8) //looping begins
19 t(i)=1/(a(i));
20 i=i+1;
21 end //as 1/N plot is needed
22
23 plot(p,a,"o-");
24 title("Fig.6.19(a) Example3 Drying Rate curve");
25 xlabel("X— Moisture content , X(kg/kg) ——>");
26 ylabel("Y— Drying Rate, N(kg/hr.m^2 ——>");
27 xset('window',1);
28 plot(p,t,"o-");
29 title("Fig.6.19(b) Example3 1/N vs X");
30 xlabel("X— Moisture content , X(kg/kg) ——>");
31 ylabel("Y— 1/N, hr,m^2/kg ——>");
32 //from X=0.6 to 0.44 ,falling rate is non linear and

```

```

    from X=.44 to .14 falling rate is linear
29
30 printf("\n from the graph we get critical moisture
    content as 0.6 kg moisture/kg dry solid");
31
32 //end

```

Scilab code Exa 6.3.b total time for drying

```

1
2
3 clear all;
4 clc;
5 printf("\t Example 6_3_b\n");
6
7
8 //part(ii)
9 w1=5; //wet of wet solid
10 c1=.5/(1-.5); //moisture content
    per kg wet solid
11 w2=5*0.5; //moisture for 5kg
    wet solid
12 w3=w1-w2; //weight of dry
    solid
13 xbar=0.05; //equillibrium
    moisture content
14 Xbar=xbar/(1-xbar); //equillibrium
    moisture content
15 Ls=2.5; //mass of bone dry
    solid ais the drying surface
16 A=5; //both upper surafce
    and lower surface are exposed
17 Nc=0.6; //in kg/m^2*hr
18 //from X=0.6 to 0.44 ,falling rate is non linear and
    from X=.44 to .14 falling rate is linear

```



```

19 X2=.15/(1-.15);
20 Xcr=.6; //kg moisture per kg
    dry solid
21 //so we can find time fro drying from 0.6 to .44
    graphically and then for X=.44 to .1765
22 X1=1; //moisture content on dry
    basis intially
23 t1=Ls/(A*Nc) *(X1-Xcr); //time taken for
    constant drying rate(fromX=1 to .6)
24 X1=.44; //moisture content on
    dry basis
25 t2=(Ls/(A*Nc))*((Xcr-Xbar)*log((X1-Xbar)/(X2-Xbar)))
    ;
26 t3=0.0336*Ls/Nc; //fro graph we get
    from X=.6 to .44
27 tttotal=t1+t2+t3; //total time for
    drying the wet slab
28 printf("\n the total time for drying the wet slab to
    15 percent moisture on wet basis is :%f min",
    tttotal*60);
29
30 //end

```

Scilab code Exa 6.4 time for drying the sheets

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.4\n");
5
6 //table X— .30    .20    .18    .15    .14    .11    .07
    .05
7 //      N— 1.22    1.22    1.14    .90    .80    .56    .22
    .05
8 //let Ls/A=p

```

```

9 p=48; //mass of bone dry solid ais
    the drying surface
10 v=1.5*1.5*.5; //volume of material
11 Nc=1.22; //in kg/m^2*hr
12 Xcr=0.2; //critical moisture content
13 X1=0.25; //moisture content on dry
    basis intially
14 X2=0.08; //moisture content on dry
    basis finally after drying
15 Xbar=0.025; //equillibrium moisture
16
17 //tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-
    Xbar)));
18
19 t1=p/(Nc) * (X1-Xcr); //time taken for constant
    drying rate period
20 //table X— .18 .15 .14 .11 .07
    .05
21 // 1/N— .8772 1.11 1.25 1.7857 4.545 20
22
23 // equillibrium relation is given under
24 p = [.18 .15 .14 .11 .07 .05];
25 a = [.8772 1.11 1.25 1.7857 4.545 20];
26
27 plot(p,a,"o-");
28 title("Fig.6.20 Example4 1/N vs X for fallling rate
    period");
29 xlabel("X— Moisture content , X(kg/kg) ——>");
30 ylabel("Y— 1/N, hr,m^2/kg ——>");
31
32 a=14*.025*1; //area under the curve
33 t2=a*48; //time taken for varying drying
    period
34 tttotal=t1+t2; //total time taken
35 printf("\n total time for drying the material from
    25 to 8 percent moisture under same drying
    conditions is :%f hr",tttotal);
36

```

37 //end

Scilab code Exa 6.5.a Calculation and plot of drying rates

```
1
2 clear all;
3 clc;
4 printf("\t Example 6_5_a\n");
5
6 //table 6.5.1
7 //S.NO.      Time (Hr)                weight of
   wet material(kg)
8 //          0.0                        5.314
9 //          0.4                        5.238
10 //         0.8                        5.162
11 //         1.0                        5.124
12 //         1.4                        5.048
13 //         1.8                        4.972
14 //         2.2                        4.895
15 //         2.6                        4.819
16 //         3.0                        4.743
17 //         3.4                        4.667
18 //         4.2                        4.524
19 //         4.6
```

```

20 //                                     4.468
                                     5.0
                                     4.426
21 //                                     6.0
                                     4.340
22 //                                     infinite
                                     4.120
23
24 w=[5.314 5.238 5.162 5.124 5.048 4.972 4.895 4.819
      4.743 4.667 4.524 4.468 4.426 4.340 4.120]
25 t=[0.0 0.4 0.8 1.0 1.4 1.8 2.2 2.6 3.0 3.4 4.2 4.6
      5.0 6.0]
26 //part(i)
27 x=4.120; //weight of the
          dried material
28 printf("\n moisture content (dry basis) ");
29 i=1; //looping starts
30 while(i<16) //calculation of
          moisture content
31 p(i)=(w(i)-x)/x;
32 printf("\n      :%f",p(i));
33 i=i+1;
34 end
35 printf("\n      \n Drying rate kg/hr*m^2");
36 i=2;
37 while(i<15)
38     a(i)=(p(i-1)-p(i))*4.12/(t(i)-t(i-1));
39     printf("\n      :%f ",a(i));
40     i=i+1;
41 end
42 a(1)=.19;
43 a(15)=0;
44 printf("\n\n from the above data it is clear that
          critical moisture content Xcr=0.11");
45 plot(p,a,"o-");
46 title("Fig.6.19(a) Example3 Drying Rate curve");
47 xlabel("X— Moisture content , X(kg/kg) ——>");
48 ylabel("Y— Drying Rate, N(kg/hr.m^2 ——>");

```

```
49
50 //end
```

Scilab code Exa 6.5.b amount of air required

```
1
2 clear all;
3 clc;
4 printf("\t Example 6_5_b\n");
5
6 //part(ii)
7 w1=4.934;           //weight after two
   hours
8 w0=5.314;           //initial weight
9 w2=w0-w1;           // water evaporated in 2
   hrs
10 H1=.01;            //humidty of incoming air
11 H2=.03;            //humidity of leaving air
12 yout=.03;
13 yin=.01;
14 Gs=w2/(yout- yin); //water carried away
15 printf("\n the amount of air required in 2hours is :
   %f kg",Gs);
16 //end
```

Scilab code Exa 6.5.c actual time of the falling peroid

```
1
2 clear all;
3 clc;
4 printf("\t Example 6_5_c\n");
5
6 //part(iii)
```

```

7      //let us choose the consistency of 11 and 13
      readings
8  Xbar=0;          //equillibrium moisture
      content
9  Ls=4.12;        //mass of bone dry solid
      ais the drying surface
10 A=1;           //both upper surafce and
      lower surface are exposed
11 Nc=0.19;       //in kg/m^2*hr
12 X1=.098;       //moisture content on dry
      basis intially
13 Xcr=.11;       //kg moisture per kg dry
      solid
14 X2=0.074;      //moisture content on
      dry basis finally
15 tfall=(Ls/(A*Nc))*((Xcr-Xbar)*log((X1-Xbar)/(X2-Xbar
      )));
16 printf("\n from this data we get time as :%f hour",
      tfall);
17 printf("\n the actual time is 0.8 hours");
18 //end

```

Scilab code Exa 6.6 time saved in drying

```

1
2  clear all;
3  clc;
4  printf("\t Example 6.6\n");
5      //wooden cloth is dried from 100 to 10
      and then th efinal moisture content
      is changed to 16 percent from 10;
6
7  Xcr=0.55;       //critical moisture content
8  X1=1;          //moisture content on dry basis
      intially

```

```

9 X2=.1; //moisture content on dry basis
  finally after drying
10 Xbar=.06; //equilibrium moisture
11 //since eqn 1 is to be divided by eqn 2 so let
  the value of  $L_s/A*N_c$  be = 1 as it will be
  cancelled
12 p=1; //let  $L_s/A*N_c$  be =p
13 p=poly([0], 'p'); //calc. of time 1
14 tbar=1; //since the eqns are
  independent of tbar
15 t1=roots(tbar- p*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
  /(X2-Xbar))))); //-----eqn1
16 X2bar=.16;
17 p=poly([0], 'p'); //calc. of time 2
18 t2=roots(tbar- p*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
  /(X2bar-Xbar))))); //-----eqn2
19
20 //let  $t_1/t_2$  be = k
21 k=t1/t2;
22 ans=1/k-1; //reduction in time for drying
23 printf("\n the reduction in time for drying is :%f
  percent", ans*100);
24 //end

```

Scilab code Exa 6.7 time for drying the sheets from 30 to 6 percent

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.7\n");
5 //assume rate of drying in the falling
  rate period is directly proportional to
  the free moisture content
6
7 //Ls/ A*Nc is unknown;

```

```

8
9 Xcr=0.14;           //critical moisture content
10 X1=.3/(1-.3);     //moisture content on dry
    basis intially
11 X2=0.1/(1-0.1);   //moisture content on dry
    basis finally after drying
12 Xbar=.04;         //equillibrium moisture
13 tbar=5;           //time needed to dry from 30 to
    6 percent on bone dry basis
14
15 //let Ls / A*Nc be = p
16 p=poly([0], 'p'); //calc. of Ls / A*Nc be = p
    value
17 x=roots(tbar-p * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
    /(X2-Xbar))));
18 printf("\n the value of Ls/ A*Nc is :%f",x);
19
20 //new X1 AND X2 are now given as follows
21 X1=0.3/(1-.3);     //new moisture content
    on dry basis intially
22 X2=0.064;         //new moisture content on
    dry basis finally after drying
23 tbar=x * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-
    Xbar)));
24 printf("\n the time for drying the sheets from 30 to
    6 percent moisture under same drying conditions
    is :%f hr",tbar);
25
26 //end

```

Scilab code Exa 6.8.a time for drying the material

```

1
2 clear all;
3 clc;

```



```

4 printf("\t Example 6_8_a\n");
5
6 Ls=1000;           //mass of bone dry solid ais
   the drying surface
7 A=55;             //both upper surafce and lower
   surface are exposed
8 v=.75;           //velocity of air
9 Nc=.3*10^-3;     //in kg/m^2*s
10 x2=.2;           //moisture content on wet
   basis finally after drying
11 Xcr=0.125;       //crtical moisture content
12 X1=0.15;         //moisture content on dry
   basis intially
13 X2=0.025;        //moisture content on dry
   basis finally after drying
14 Xbar=0.0;        //equillibrium moisture
15
16 tbar=(Ls/(A*Nc))*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar)));
17
18 printf("\n the time for drying the sheets from .15
   to .025 kg water /kg of dyr solid moisture under
   same drying conditions is :%f hour",tbar/3600);
19
20 //end

```

Scilab code Exa 6.8.b time saved if air velocity is increased

```

1
2 clear all;
3 clc;
4 printf("\t Example 6_8_b\n");
5
6 //tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-
   Xbar)));

```

```

7
8 //part(i)
9 // assuming only surface evaporation and
    assuming air moves parellel to surface
10
11 //Nc=G^0.71;           G=V*d
12 //so NC = k* V^.71
13 Ls=1000;              //mass of bone dry solid ais
    the drying surface
14 A=55;                 //both upper surafce and lower
    surface are exposed
15 v=.75;                //velocity of air
16 Nc=.3*10^-3;         //in kg/m^2*s
17 x2=.2;                //moisture content on wet
    basis finally after drying
18 Xcr=0.125;           //critcal moisture content
19 X1=0.15;              //moisture content on dry
    basis intially
20 X2=0.025;            //moisture content on dry
    basis finally after drying
21 Xbar=0.0;             //equillibrium moisture
22 tbar=3.8077;         //time to dry material ,
    calculated from previous part
23 V1=.75;              //old velocity
24 V2=4;                //new velocity
25 Nc2=Nc*(V2/V1)^.71; //in kg/m^2*s
26 t2=(Ls/(A*Nc2))*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
    /(X2-Xbar)));      //if air velocity is increased
    to 4
27 t=tbar-t2/3600;     //time saved
28
29 printf("\n the time saved , if air velocity is
    increased to 4 m/s: %f",t);
30
31 //end

```

Scilab code Exa 6.9 time for drying the fibreboard

```
1
2 clear all;
3 clc;
4 printf("\t Example 6.9\n");
5
6         //determine the drying condition of
           sample 0.3*0.3 size.sheet lost weight
           at rate of 10-4 kg/s until the
           moisture fell to 60 percent
7
8 x1=.75;           //moisture content on wet
   basis
9 xbar=0.1;        //equilllibrium moisture on
   dry basis
10 xcr=0.6;        //critical moisture content
11 Ls=0.90;        //mass of bone dry solid ais
   the drying surface
12 A=0.3*0.3*2;    //both upper surafce and
   lower surface are exposed
13 //A*Nc=10-4;
14 x2=.2;          //moisture content on wet
   basis finally after drying
15 Xcr=0.6/0.4;    //crtical moisture content
16 X1=3;           //moisture content on dry
   basis intially
17 X2=0.25;        //moisture content on dry
   basis finally after drying
18 Xbar=0.1/0.9;   //equillibrium moisture
19 tbar=Ls/(10-4) * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar
   )/(X2-Xbar)));
20 printf("\n the time for drying the sheets from 75 to
   25 percent moisture under same drying conditions
```

```

        is :%f hr",tbar/3600);
21
22 //end

```

Scilab code Exa 6.10 time for drying the moist material

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.10\n");
5         //determine the drying condition of
           sample 0.3*0.3 size.sheet lost weight
           at rate of 10-4 kg/s until the
           moisture fell to 60 percent
6
7 //Ls/ A*Nc is unknown;
8
9 Xcr=0.16;           //critical moisture content
10 X1=.33;           //moisture content on dry
    basis intially
11 X2=0.09;           //moisture content on dry
    basis finally after drying
12 Xbar=.05;         //equilibrium moisture
13 tbar=7;           //time needed to dry from 33 to
    9 percent on bone dry basis
14
15         //let Ls / A*Nc be = p
16 p=poly([0], 'p'); //calc. of Ls / A*Nc be = p
    value
17 x=roots(tbar-p * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
    /(X2-Xbar))));
18
19 //new X1 AND X2 are now given as follows
20 X1=0.37;           //new moisture content on dry
    basis intially

```

```

21 X2=0.07; //new moisture content on dry
    basis finally after drying
22 tbar=x * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-
    Xbar)));
23 printf("\n the time for drying the sheets from 33 to
    9 percent moisture under same drying conditions
    is :%f hr",tbar);
24
25 //end

```

Scilab code Exa 6.11 time for drying the slab from 65 to 30 percent

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.11\n");
5
6 d=.22; //density of dry pulp in g
    /cc;
7 x1=.65; //moisture content on wet
    basis
8 x2=.3; //moisture content on wet
    basis
9 Ls=2.5; //mass of bone dry solid ais
    the drying surface in kg
10 A=1.5*1.5*2; //both upper surafce and
    lower surface are exposed
11 v=1.5*1.5*.5; //volume of material
12 Nc=1.4; //in kg/m^2*hr
13 Xcr=1.67; //critical moisture content
14 X1=x1/(1-x1); //moisture content on
    dry basis intially
15 X2=x2/(1-x2); //moisture content on dry
    basis finally after drying
16 Xbar=0.0; //equillibrium moisture

```

```

17
18         //initial moisture is more than Xcr, so
           there is constant rate drying period and
           only falling rate peroid is observed
19 tbar=(Ls/(A*Nc))*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
           /(X2-Xbar)));
20 printf("\n the time for drying the sheets from 65 to
           30 percent moisture under same drying conditions
           is :%f hour",tbar);
21
22 //end

```

Scilab code Exa 6.12 time for drying the slab

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.12\n");
5
6 d=.22; //density of dry pulp in g
        /cc;
7
8 Ls=1.125*10^-2*.22*10^3; //mass of
        bone dry solid ais the drying surface
9 A=1.5*1.5*2; //both upper surafce and
        lower surface are exposed
10 v=1.5*1.5*.5; //volume of material
11 Nc=1.4; //in kg/m^2*hr
12 x2=.2; //moisture content on wet
        basis finally after drying
13 Xcr=0.46; //crtical moisture content
14 X1=0.15; //moisture content on dry
        basis intially
15 X2=0.085; //moisture content on dry
        basis finally after drying

```

```

16 Xbar=0.025;           //equillibrium moisture
17
18 //tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-
    Xbar)));
19 // but initial moisture is more than Xcr, so
    there is constant rate drying period and
    only falling rate peroid is observed
20 tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((X1-Xbar)/(X2-Xbar)
    ));
21 printf("\n the time for drying the sheets from 15 to
    8.5 percent moisture under same drying
    conditions is :%f min",tbar*60);
22
23 //end

```

Scilab code Exa 6.13 time for drying the filter cake

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.13\n");
5
6 //Ls/ A*Nc is unknown;
7
8 Xcr=0.14;           //critical moisture content
9 x1=0.3;           //moisture content on wet
    basis
10 x2=0.1;           //moisture content on wet
    basis
11 X1=x1/(1-x1);           //moisture content on
    dry basis intially
12 X2=x2/(1-x2);           //moisture content on dry
    basis finally after drying
13 Xbar=0.04;           //equillibrium moisture
14 tbar=5;           //time needed to dry from 30 to

```

```

15         10 percent on bone dry basis
16         //let Ls / A*Nc be = p
17 p=poly([0], 'p');           //calc. of Ls / A*Nc be = p
    value
18 x=roots(tbar-p * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
    /(X2-Xbar))));
19
20 //new X1 AND X2 are now given as follows
21 x1=.3;                       //new moisture content on wet
    basis
22 x2=0.06;                     //new moisture content on
    wet basis
23 X1=x1/(1-x1);               //new moisture content on
    dry basis intially
24 X2=x2/(1-x2);               //new moisture content on
    dry basis finally after drying
25 tbar=x * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-
    Xbar)));
26 printf("\n the time for drying the sheets from 30 to
    10 percent moisture under same drying
    conditions is :%f hr",tbar);
27
28 //end

```

Scilab code Exa 6.14 time for drying the sheets

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.14\n");
5
6 d=450;                       //density of dry pulp in
    kg/m^3;
7 thickness=0.05;              //thickness in m^2

```



```

8 Ls=d*thickness;           //mass of bone dry solid
   ais the drying surface
9 A=1;                     //area in m^2
10 v=1*5*10^-2;           //volume of material
11 Nc=4.8;                 //in kg/m^2*hr
12 xcr=.2;
13 xbar=0.02;
14 x1=.45;                 //new moisture content on wet
   basis
15 x2=0.05;                 //new moisture content on
   wet basis
16 X1=x1/(1-x1);           //new moisture content on
   dry basis intially
17 X2=x2/(1-x2);           //new moisture content on
   dry basis finally after drying
18 Xbar=xbar/(1-xbar);     //critical moisture
   content
19 Xcr=xcr/(1-xcr);        //equillibrium moisture
20
21 //tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-
   Xbar)));
22 // but initial moisture is more than Xcr, so
   there is constant rate drying period and
   only falling rate peroid is observed
23 tbar=Ls/(A*Nc) * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar)));
24 printf("\n the time for drying the sheets from 45 to
   5 percent moisture under same drying conditions
   is :%f min",tbar);
25
26 //end

```

Scilab code Exa 6.15 heater load per unit mass of dry air

1

```

2 clear all;
3 clc;
4 printf("\t Example 6.15\n");
5
6 t1=20;           //ambient air temperature
7 t2=70;           //exhaust air temperature
8 r1=150;          //evaporation of water
9 r2=.25;          //outlet solid moisture
   content
10 t3=15;           //inlet solid temperature
11 t4=65;           //outlet solid temperature
12 p=5;            //power demand in KW
13 h=18;           //heat loss in kj
14
15 h1=1;           //mean specific heat of dry
   air in kj/kg*K
16 h2=1.25;        //mean specific heat of
   dry material in kj/kg*K
17 h3=4.18;        //mean specific heat of
   moisture in kj/kg*K
18 e=2626;         //enthalpy of saturated
   water vapour in kj/kg
19
20           //basis is 1hr
21 a1=r1*h3*(t4-t3); //heat required for
   heating 150 kg of water from 15 to 65
22 a2=r1*e;         //heat required for 150
   kg water evaporation
23 a3=2000*h1*(t2-t1); //heat required for
   heating air from 20 to 70
24 a4=r2*h3*(t4-t3); //heat required for
   heating moisture from 15 to 65
25 a5=120*h2*(t4-t3); //heat required for
   heating dry solid from 15 to 65
26 hlost=h*3600;   //heat lost in kj
27 total=(a2+a3+a4+a5+hlost)/3600; //total heat lost
28 printf("\n :%f kW of heat required for 2000kg/hr of
   dry air",total);

```

```

29 ans1=a2+a1;           //heat needed for evaporation
30 printf("\n heat needed fro evaporation is :%f",ans1
    /3600);
31 ans2=(ans1/3600)/total; //fraction of this heat
    needed for evaporation
32 printf("\n fraction of this heat needed for
    evaporation:%f",ans2);
33
34
35 //end

```

Scilab code Exa 6.16 surface area of the roller

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.16\n");
5 m1=.12;           //initial moisture
    content
6 dT=85;           //product of 85 degree is
    used in design purpose
7 U=1700;           //overall heat transfer
    coefficient
8 m2=.4;           //final moisture content
9 r=20;           //production rate
10 //4 kg of moisture is present in 100 kg product
11 t=4*20/100;     // moisture content in 20 kg
    moisture
12 w=20-t;         //dry solid weight
13 i=w*m1/(1-m1); //initial moisture content
14 j=i-t           //water evaporated
15 ds=2296.1;     //latent heat for vaporisation
    at 85 degree in kj/kg
16 h=j*ds;        //heat required (assuming th
    esolid mix. enters at 85)

```

```

17 //U*A*dT = j*ds
18 A=h/(U*dT); //surface area of the roller
    required to produce aproduction rate of 20 kg
    product per hour
19 printf("\n surface area of the roller required to
    produce aproduction rate of 20 kg product per
    hour:%f m^2",A/3.600);
20
21 //end

```

Scilab code Exa 6.17 time for drying the sheets

```

1
2 clear all;
3 clc;
4 printf("\t Example 6.17\n");
5 //moisture content reduces from 25 to 2
6 r=7.5*10^-5; //constant drying
    rate in kg/s
7 A1=.3*.3**2; // area of the
    sppecimen
8 Nc=r/A1; //drying rate
9 Xcr=.15/0.85; //0.15 is the
    critical moisture content
10 Xo=.25/.75; //0.25 is the initial
    moisture content
11 Xfinal=.02/0.98; //0.02 is the final
    moisture content
12 Xbar=0; //equillibrium moisture
    content
13 A=1.2*.6*2; //area of the new solid
14 Ls=28.8; //bone dry weight of
    new solid
15 v1=.3*.3*.006; //volume of the old
    solid;

```

```

16 v2=.6*1.2*.012;           //volume of the new
    solid
17 w2=1.8;                   //weight of the old
    solid
18 w3=864*10-5*1.8*10-5/54; //weight of the bone
    dry solid
19
20 //Nc is prporional to =(t-ts) = (G)0.71-----
    whrere G is the mass flow rate
21 v1=3;                     //old velocity
22 Tg=52;                    //old dry bulb
    temperature
23 Tw=21;                    //wet bulb temperature
24 H=.002;                   //humidity
25 SH=0.015;                 //saturated humidity
26 vnew=5                    //new velocity
27 Tgnew=66;                 //new DBT
28 Twnew=24;                 //new WBT
29 Hnew=.004;                //new humidity
30 SH=.020;                  //new satuurated humidity
31
32 //hence drying rate of air under new
    condition
33 Nc=4.167*10-4*((vnew/v1)*(273+Tg)/(273+Tgnew))0.71
    * ((.019-H)/(.015-H)); //drying rate of air under
    new condition in kg/m2*s
34 DT=Ls/(A*Nc) * ((Xo-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(
    Xfinal-Xbar)));
35 printf("\n the time for drying the sheets from 25 to
    2 percent moisture under same drying conditions
    is :%f hours",DT/3600);
36
37
38 //end

```

Chapter 7

Crystallisation

Scilab code Exa 7.1 total weight of the solution

```
1
2 clear all;
3 clc;
4 printf("\t Example 7.1\n");
5
6 //let x be the weight of water in the quantity of
   solution needed
7
8 c=.498; //solute content afetr
   crystallisation
9 W1=111; //molecular weight of CaCl2
10 W2=219; //molecular weight of CaCl2.6
   H2O
11 M1=(108/W2)*100; //water present in 100kg of
   CaCl2.6H2O
12 M2=(W1/W2)*100; //CaCl2 present in 100kg of
   CaCl2.6 H2O
13 //t=M2+c*x; //total weight entering the
   solubility
14 //x+49.3; total water solubility used
15 //s*(x+49.3)/100 //total Cacl2 after solubility
```

```

16 x=poly([0], 'x');           //calc. x the weight of
    crystal
17 t=roots((M2+c*x)-(x+49.3)*.819);
18 printf("\nthe weight of water in the quantity of
    solution needed :%f kg",t);
19
20 h=(c)*t;                    //weight of CaCl2 corresponding
    to weight water
21 tw=t+h;                    // total weight of the solution
22 printf("\nthe total weight of the solution is :%f
    kg",tw);
23 //end

```

Scilab code Exa 7.2 percentage saturation and yield

```

1
2 clear all;
3 clc;
4 printf("\t Exercise 7.1\n");
5
6 //part(i)
7 w1=1000;                    //weight of solution to be
    cooled
8 s1=104.1;                  //solubility at 50 degree per
    100 kg of water
9 s2=78.0;                   //solubility at 10 degree per
    100 kg of water
10 a2=45;                    //percentage of sodium nitrate
    in the solution per 100kg of solution
11
12 x1=s1/(100+s1)*100;        //percentage of
    saturated solution at 50 degree
13 tw=(a2/(100-a2))/(x1/(100-x1)); //the
    percentage saturation
14 printf("\nthe percentage saturation is :%f percent"

```

```

    ,tw*100);
15
16 //part(ii)
17 //let x be the weight of NaNO3 crystal formed after
    crystallisation
18 x=poly([0], 'x'); //calc. x the weight of
    crystal
19 t=roots((w1*a2/100)-(x+(w1-x)*s2/(100+s2)));
20 printf("\n the weight of NaNO3 crystal formed after
    crystallisation :%f kg",t);
21
22 //part(iii)
23 yield=t/(a2*w1/100); //yield = weight of
    NaNO3 crystal formed/weight of NaNO3
24 printf("\n the percentage yield is:%f percent",yield
    *100);
25 //end

```

Scilab code Exa 7.3 temperature to which solution should be cooled

```

1
2 clear all;
3 clc;
4 printf("\t Example 7.3\n");
5
6 s1=19.75; //solubility at 70 degree per
    100 gm of water
7 s2=16.5; //solubility at 50 degree per
    100 gm of water
8 s3=12.97; //solubility at 30 degree per
    100 gm of water
9 s4=9.22; //solubility at 10 degree per
    100 gm of water
10 s5=7.34; //solubility at 0 degree per
    100 gm of water

```



```

11          //basis is 1000kg of saturated solution
12 w1=1000*(s1/(s1+100));          //weight of K2SO4
    in the original solution
13 w2=1000-w1;          //weight of water
    in kg
14 w3=w1*.5;          //weight of K2SO4 in
    the solution
15 wp=w3/(w3+w2);          //weight percent of
    K2SO4 in the solution after crystallisation
16 printf("\n for the corresponding temperature to :%f
    percent of K2SO4 is 15 degree (by linear
    interpolation between 10 to 30 degree) ",wp*100);
17
18 //end

```

Scilab code Exa 7.4 percentage saturation and yield

```

1
2 clear all;
3 clc;
4 printf("\t Example 7.4\n");
5 //part(i)
6 a1=146;          //solubility at 70 degree
7 a2=121;          //solubility at 10 degree
8 t1=58;          // percentage of solute
    content
9 t2=40.66;
10 x1=a1/(100+a1) *100;          //percentage of
    saturated solution at 50 degree
11 tw=(t1/42)/(x1/t2);          //the percentage saturation
12 printf("\nthe percentage saturation is :%f percent"
    ,tw*100);
13
14 //part(ii)
15 p1=2000*.58;          //weight of solute in 200

```

```

kg of solution 2000*.58
16 //let x be the weight of crystal formed after
    crystallisation
17 x=poly([0], 'x'); //calc. x the weight of
    crystal
18 t=roots((1160)-(x+(1055.02-.547*x)));
19 printf("\n the weight of NaNO3 crystal formed after
    crystallisation :%f kg",t);
20
21 //part(iii)
22 yield=t/p1; //yield = weight of NaNO3
    crystal formed/weight of NaNO3
23 printf("\n the percentage yield is:%f percent",yield
    *100);
24 //end

```

Scilab code Exa 7.5 the weight of Na₂SO₄ hydrate crystal

```

1
2 clear all;
3 clc;
4 printf("\t Example 7.5\n");
5
6 p1=.3; //percentage of the solute
    in the solution
7 w1=1000; //weight of the solution
    taken
8 w2=142; //molecular weight of Na2SO4
    .
9 M1=(w2/(180+w2)); //solute (Na2SO4) present
    in the Na2CO3.10H2O solution
10 s1=40.8; //solubility of Na2SO4 at 30
    degree per 100 gm of water
11 s2=9.0; //solubility of Na2SO4 at 10
    degree per 100 gm of water

```

```

12 //percent weight of solute in Na2SO4.10H2O= 144/322
13 //let x be the weight of crystal formed
14 x=poly([0], 'x'); //calc. x the weight of
    crystal
15 t=roots((w1*40.8/140.8) - (.442*x+(w1-x)*(s2/(100+s2))
    ));
16 printf("\n the weight of crystal formed after
    crystallisation :%f kg",t);
17
18 //end

```

Scilab code Exa 7.6 percentage yield of Na₂CO₃ hydrated crista

```

1
2 clear all;
3 clc;
4 printf("\t Example 7.6\n");
5
6 s1=12.5; //solubility of Na2CO3 at
    10 degree per 100 gm of water
7 p1=.3; //percentage of the solute
    in the solution
8 w1=2000; //weight of the solution
    taken
9 w2=106; //molecular weight of Na2CO3
    .
10 M1=(w2/(180+w2)); //solute (Na2CO3) present
    in the Na2CO3.10H2O solution
11 //let x be the quantity of Na2CO3.10H2O crystal
    formed
12 x=poly([0], 'x'); //calc. x the weight of
    crystal
13 t=roots(w1*p1-M1*x-(w1-x)*(s1/(100+s1)));
14 printf("\n the weight of quantity of Na2CO3.10H2O :
    %f kg",t);

```

```

15 //in the book the ans is wrong, they have calculated
    2000*0.3-2000*12.5/112.5 as =x(miscalculation)
16
17 p=(286/106)*w1*p1;           //weight of Na2CO3.10
    H2O crystal present in the original solution
18 yield=t/p;                   //percentage yield
19 printf("\n percentage yield :%f percent",yield*100);
20 //end

```

Scilab code Exa 7.7 percentage yield of K₂CO₃ hydrated crystal

```

1
2 clear all;
3 clc;
4 printf("\t Example 7.7\n");
5
6 s1=139.8;                       //solubility at 80 degree per
    100 gm of water
7 s2=110.5;                       //solubility at 20 degree per
    100 gm of water
8 w2=174.2;                       //molecular weight of
    K2CO3.10H2O
9 M1=(138/w2)*100;                //water present in 100kg of
    K2CO3.10H2O
10 //let x be the quantity of Na2CO3.10H2O
11 x=poly([0], 'x');              //calc. x the weight of
    crystal
12 t=roots(500*(139.8/239.8) - .7921*x - (500-x)
    *110.5/210.5);
13 printf("\n the weight of quantity of K2CO3.10H2O
    formed :%f kg",t);
14
15 p=(174/138)*500*(139.8/239.8); //weight of
    crystal present in the original solution
16 yield=t/p;                       //percentage yield

```

```
17 printf("\n percentage yield :%f percent",yield*100);
18 //end
```

Scilab code Exa 7.8 percentage yield FeSO₄ hydrate crystal

```
1
2 clear all;
3 clc;
4 printf("\t Example 7.8\n");
5
6 s1=20.51; //solubility at 10 degree per
   100 gm of water
7 w2=277.85; //molecular weight of
   FeSO4.7H2O
8
9 //let x be the quantity of Na2CO3.10H2O
10 x=poly([0], 'x'); //calc. x the weight of
   crystal
11 t=roots(900*.4-.5465*x-(900-x)*20.5/120.5);
12 printf("\n the weight of quantity of FeSO4.7H2O
   formed :%f kg",t);
13
14 p=(277.85/151.85)*900*(0.4); //weight of
   crystal present in the original solution
15 yield=t/p; //percentage yield
16 printf("\n percentage yield :%f percent",yield*100);
17 //end
```

Scilab code Exa 7.9 percentage saturation and weight of Cesium chloride crystal

```
1
2 clear all;
```

```

3  clc;
4  printf("\t Example 7.2\n");
5
6  //part(i)
7  a1=229.7;           //solubility at 60 degree
8  a2=174.7;          //solubility at 60 degree
9  t1=68;             // percentage of sodium
    nitrate
10 t2=30.34;
11 x1=a1/329.7 *100;   //percentage of saturated
    solution at 50 degree
12 tw=(t1/32)/(x1/t2); //the percentage saturation
13 printf("\nthe percentage saturation is :%f percent"
    ,tw*100);
14
15 //part(ii)
16 //let x be the weight of Cesium chloride crystal
    formed after crystallisation
17 x=poly([0], 'x');   //calc. x the weight of
    crystal
18 t=roots(1000*.68-(x+(1000-x)*174.7/274.7));
19 printf("\n the weight of CaCl2 crystal formed after
    crystallisation :%f kg",t);
20
21 //part(iii)
22 yield=t/680;        //yield = weight of CaCl2
    crystal formed/weight of CaCl2
23 printf("\n the percentage yield of Cesium chloride
    is:%f percent",yield*100);
24 //end

```

Scilab code Exa 7.10 weight of Na₂CO₃hydrate needed to dissolve Na₂CO₃

```

1
2  clear all;

```

```

3  clc;
4  printf(" \t Example 7.10\n");
5
6  s1=38.8;           //solubility at 30 degree per
   100 gm of water
7  s2=12.5;          //solubility at 10 degree per
   100 gm of water
8  w2=296;           //molecular weight of Na2CO3
   .10H2O
9  per=116/w2 *100;  //percentage solute in
   Na2CO3.10H2O
10
11 //let x be the quantity of Na2CO3.10H2O
12 w=200;            //original solotion weight
13 m1=w*(s2/(s2+100)); //weight of Na2CO3.10H2O
   needed to dissolve Na2CO3 present in the original
   solotion
14 w3=w-m1;         //weight of water
15 //w4=m1+per/100; //weight of Na2CO3 after
   dissolution
16 x1=s1/(s1+100); //weight fraction of
   solute after dissolution
17 printf(" \n the weight of quantity of Na2CO3.10H2O
   formed :%f kg",w3);
18
19 //for the total solution after dissolution
20 x=poly([0], 'x'); //calc. x the weight of
   crystal
21 t=roots((m1+per*x/100)-((m1+per*x/100)+(w3+.609*x))*
   x1);
22 printf(" \nweight of Na2CO3.10H2O needed to dissolve
   Na2CO3 present in the original solution %f kg",t
   );
23
24 //end

```

Scilab code Exa 7.11 weight of Na₂CO₃ hydrated formed

```
1
2 clear all;
3 clc;
4 printf("\t Example 7.11\n");
5
6 s1=35; //percentage of solution
7 x1=6000; //weight of Na2CO3 solution
8 s2=21.5; //solubility at 20 degree per
   100 gm of water
9 w2=296; //molecular weight of Na2CO3
   .10H2O
10 per=116/w2 *100; //percentage solute in
   Na2CO3.10H2O
11 w1=s1*x1; //weight of solute
12 w3=x1*0.04; //weight of solution lost by
   vaporisation
13 //let x be the quantity of Na2CO3.10H2O formed
14 //making material balance
15 x=poly([0], 'x'); //calc. x the weight of
   crystal
16 t=roots(2100-(.391*x)-(6000-240-x)*(21.5/121.5));
17 printf("\n the weight of Na2CO3.10H2O crystal formed
   after crystallisation :%f kg",t);
18
19
20 //end
```

Scilab code Exa 7.12 feed rate of FeSO₄ hydrated crystal produced per hour


```

1
2 clear all;
3 clc;
4 printf("\t Example 7.12\n");
5
6             //FeSO4.7H2O
7 C=1000;           //crystal formed in kg
8 hf=26.002;       //enthalpy of the feed
9             at 80 degree in cal/g
10 hl=-1.33;       //enthalpy of the
11             saturated sol at 30 degree in cal/g
12 hc=-50.56;      //enthalpy of crystal
13 xf=40/(100+40);
14 xm=30/(100+30);
15 xc=151.84/277.85; //151.84 is the weight of
16             FeSO4
17             //component balance
18 //      F*xf = M*xm + C*xc      -----eqn 1st
19 //      F = M + 10000 + V      -----eqn 2nd
20 //      F*Hf = V*Hv + M*Hm +C*Hc-----eqn 3rd
21 Hf=26.002;      //enthalpy of the feed at
22             80 degree in cal/g
23 Hv=612;         //
24 Hm=-1.33;       //enthalpy of the saturated
25             sol at 30 degree in cal/g
26 Hc=-50.56;     //enthalpy of crystal leaving
27             the crystalliser
28
29 //solving these we get
30 a=[1 -1 -1;.286 -.231 0;26.002 1.33 -612]
31 b=[10000;5470;-505600]
32 x=inv(a)*b;     //solving out the
33             values using matrices
34 t1=x(1);        //3 solution of the
35             eqn
36 t2=x(2);
37 t3=x(3);
38 printf("\n the feed rate F= : %f kg/hr \n value of M

```

```

    = : %f kg/hr\n value of V=: %f kg/hr",t1,t2,t3);
31 //end

```

Scilab code Exa 7.13 cooling water requirement

```

1
2 clear all;
3 clc;
4 printf("\t Example 7.13\n");
5
6 C=800; //crystal formed in kg/
   hr
7 t2=49; //temp. of the
   entering fed
8 t1=27; //temp. of the product
9 t3=21; //temp. of the leaving
   cooling water
10 t4=15; //temp. of the enetring
   cooling water
11 U=175; //overall heat transfer
   coefficient
12 F=140*151.85/277.85; //feed concentration
13 xf=F/240; //concentration in feed
   solution
14 P=74*151.85/277.85; //product concentration
15 xm=P/174; //concentration of FeSO4
   in product solution
16 xc=151.85/277.85; //
17 //mass balance F = M+C -----eqn 1st
18 //sloute balance F*xf = M*xm + C*xc -----eqn
   2nd
19 //solving these we get
20 F=800*.3141/0.0866; //feed conc.
21 M=F-C; //product concentration
22 //making energy balance

```

```

23          //heat to be removed by cooling water =heat
           to be removed from solution + heat of
           crystallization
24 cp=.7;           //specific heat capacity
25 dt=(t2-t1);     //change in temp.
26 dh=15.8;        //heat of crystallization
27 Q=F*cp*dt+dh*C; //heat to be removed by
           cooling water
28 cp=1;           //specific heat capacity of
           water
29 dt=(t3-t4);     //change in temp.
30 mw=Q/(cp*dt);   //cooling water needed
31 printf("\n cooling water requiement is :%f kg/hr",mw
           );
32 //Q=U*A*(dtlm)
33 dtlm=((t2-t3)-(t1-t4))/(log((t2-t3)/(t1-t4))); //log
           mean temp. difference
34 A=Q/(U*dtlm);   //area of the
           crystallizer section
35 l=A/1.3;
36 printf("\n length of crystallliser sections needed
           is :%f m",l);
37
38 //end

```
