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

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

May 24, 2016

[1]Funded 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

Chapter 2

Diffusion

Scilab code Exa 2.1.a diffusivity of nitrogen carbondioxide mixture

```

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

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

//end
```

Scilab code Exa 2.1.b diffusivity of hydrogen chloride mixture

```

clear;
clc;
printf("\t Example 2_1_b\n");
// let A denote Hydrogen chloride and B denote air

//"part(ii)"
rA=.3339;
rB=.3711;
rAB=(rA+rB)/2;                            //molecular
   seperation at collision
ebyk_A=344.7;
ebyk_B=78.6;
ebyk_AB=(ebyk_A/ebyk_B)^.5;               //energy of
   molecular attraction
pt=200*10^3;                              //absolute
   total pressure in pascal
T=298;                                    //absolute
   temperature in kelvin
s=T/ebyk_AB;                             //collision
```

```
function
  //from chart f(T/ebyk_AB) = 0.62   let it be = x
x=0.62;                                    //collision function
MA=36.5;                                   //molecular weight of hydrogen chloride
MB=29;                                    //molecular weight of air
Mnew=((1/MA)+(1/MB))^.5;
Dab=10^-4*(1.084-.249*(Mnew))*T^1.5*((Mnew))/(pt*x*rAB^2);
printf("\n the diffisivity of hydrogen chloride-air is :%f *10^-6 m^2/s",Dab/10^-6)

//end
```

Scilab code Exa 2.2 the diffusivity of isoamyl alcohol

```

clear;
clc;
printf("\t Example 2.2\n");
                //kopp's law is valid

u=1.145*10^-3;                            //viscosity of  water1.145cp
v_a=5*.0148+12*.0037+1*.0074;             //by kopp's law
t=288;                                    //temperature of water in kelvin
MB=18;                                    //molecular weight of water
phi=2.26;                                 //association parameter for solvent-water

```

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

Scilab code Exa 2.3 diffusivity of ccl4 through oxygen

```

clear;
clc;
printf("\t Example 2.3\n");

pa1=(33/760)*1.013*10^5;                    //vapour pressure of ccl4 at 273 in pascal
pa2=0;
d=1.59;                                     //density of liquid ccl4 in g/cm^3
  //considering o2 to be non diffusing and with
T=273;                                      //temperature in kelvin
pt=(755/780)*1.013*10^5;                    //total pressure in pascal
z=.171;                                     //thickness of film
a=.82*10^-4;                                //cross-sectional area of cell in m^2
v=.0208;                                    //volume of ccl4 evaporated
t=10;                                       //time of evaporation
MB=154;                                     //molecular wght of ccl4
rate=v*d/(MB*t);                            //.0208cc of ccl4 is evaporating in 10hrs
Na=rate*10^-3/(3600*a);                     //flux in kmol/m^2*S

D_ab=Na*z*8314*273/(pt*log((pt-pa2)/(pt-pa1)));
```

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

Scilab code Exa 2.4 rate at which crystal dissolves

```

clear;
clc;
printf("\t Example 2.4\n");
z=.0305*10^-3;                          //wall thickness sorrounding the crystal
x1=0.0229;
w1=160;                                 //molecular weight of copper sulphate
w2=18;                                  //molecular weight of water
Dab=7.29*10^-10;                        //diffusivity of copper sulphatein m^2/s
  //av=d/m
Mavg=x1*w1+(1-x1)*w2;                   //average molecular wght of solution
d1=1193;                                //density of copper sulphate solution
av1=d1/Mavg;                            //value of (d/m) of copper solution

  //for pure water
d2=1000;                                //density of water
m2=18;                                  //molecular wght of water
av2=d2/m2;                              //value of (d/m) of water
```

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

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

```

clear;
clc;
printf("\t Example 2.5.a\n");
//position 1      moles          molefraction
//    air          80              0.8
//    water        20              0.2

//position 2         moles         molefraction
//      air         10              0.1
//    water         90              0.9
ya1=0.8;
ya2=0.1;
T=(273+35);                              //temperature in kelvin
pt=1*1.013*10^5;                         //total pressure in pascal
z=0.3*10^-3;                             //gas film thickness in m
Dab=.18*10^-4;                           //diffusion coefficient in m^2/s
R=8314;                                  //universal gas constant
Na=Dab*pt*(ya1-ya2)/(z*R*T)         //diffusion flux
```

```
    in kmol/m^2*s
rate=Na*100*10^-4*3600*46;        //since molecular weight of mixture is 46
printf("\n rate of diffusion of alcohol-water vapour :%f kg/hr ",rate);

//end
```

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

```

clear;
clc;
printf("\t Example 2_5_b\n");
ya1=0.8;
ya2=0.1;
T=(273+35);                              //temperature in kelvin
pt=1*1.013*10^5;                         //total pressure in pascal
z=0.3*10^-3;                             //gas film thickness in m
Dab=.18*10^-4;                           //diffusion coefficient in m^2/s
R=8314;                                  //universal gas constant

//diffusion through stagnant film
Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T);     //diffusion flux in kmol/m^2*s
rate=Na*100*10^-4*3600*46;                  //since molecular weight of mixture is 46
printf("\n rate of diffusion if water layer is stagnant :%f *10^-3 kg/s ",rate/(3600*10^-3));
//end
```

Scilab code Exa 2.6 rate of loss of hydrogen

```

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

```
//end
```

Scilab code Exa 2.7 ammonia diffusion through nitrogen

```

clear;
clc;
printf("\t Example 2.7\n");

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

Scilab code Exa 2.8 rate at which crystal dissolves

```

```

```
clear;
clc;
printf("\t Example 2.8\n");
//position 1     moles        molefraction     weight
//    ethanol    0.1478          0.02775       6.80
//    water      5.18            0.9722        93.20

//position 2     moles       molefraction       weight
//     ethanol   0.235          0.0453          10.8
//    water      4.96         0.9547            89.20
z=0.4*10^-2;                        //film thickness
   sorrounding the crystal
xa1=0.0453;                       //mole fraction of
   ethanol at pos.2
xa2=0.02775;                     //mole fraction of
   ethanol at pos.1
w1=46;                         //molecular weight of
   ethanol
w2=18;                        //molecular weight of water
Dab=74*10^-5*10^-4;     //diffusivity of ethanol
   water sol.in m^2/s
    //av=d/m
Mavg1=xa2*w1+(1-xa2)*w2;          //average molecular
   wght of solution at pos 1
d1=0.9881*10^3;                     // density of 6.8 wt%
     solution
av1=d1/Mavg1;                     //value of (d/m) of
   copper solution

    //for pure water
d2=972.8;                                             //
   density of 10.8 wt%  solution
Mavg2=xa1*w1+(1-xa1)*w2;                            //
   average molecular wght of solution at pos.2
av2=d2/Mavg2;                                     //value
   of (d/m) of water

allavg=(av1+av2)/2;                              //average
```

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

Scilab code Exa 2.9 diffusion rate of acetic acid

```

clear;
clc;
printf("\t Example 2.9\n");
//position 1          kmoles         molefraction      weight
//    acetic acid     0.167           0.0323        10
//    water             5            0.9677         90

//position 2           kmoles        molefraction      weight
//     aceitic acid      0.067            0.0124           4
//    water                5.33          0.9876          96

                //basis : 100kg of mixture
z=2*10^-3;                          //film thickness sorrounding the water
xa1=0.0323;                         //mole fraction of ethanol at pos.2
xa2=0.0124;                        //mole fraction of ethanol at pos.1
w1=60;                            //molecular weight of acetic acid
w2=18;                           //molecular weight of water
```

```
Dab=0.000095;                //diffusivity of acetic
    water sol.in m^2/s
    //av=d/m
Mavg1=xa1*w1+(1-xa1)*w2;     //average molecular wght
     of solution at pos 1
d1=1013;                     // density of 10 % acid
av1=d1/Mavg1;                //value of (d/m) of
    copper solution

    //for pure water
d2=1004;                     //density of 4% acid
Mavg2=xa2*w1+(1-xa2)*w2;     //average molecular wght
     of solution at pos.2
av2=d2/Mavg2;                 //value of (d/m) of
    water

allavg=(av1+av2)/2;          //average value of d/m
        //assuming water to be non diffusing
Na=Dab*(allavg)*log((1-xa2)/(1-xa1))/z;     //
    diffusion rate of acetic acid aacross film of non
     diffusing water sol.
printf("\n diffusion rate of acetic acid aacross
    film of non diffusing water sol. :%f kmol/m^2*s",
    Na);
//end
```

Scilab code Exa 2.10.a rate of mass transfer

```

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

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

//end
```

Scilab code Exa 2.10.b partial pressure of co2

```

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

```
    m^2/s
R=760*22.414/273;          //universal gas constant in
     mmHg*m^3*K*kmol
    //carbondioxide and oxygen experiences equimolar
        counter diffusion
T=298;                         //temperature in
    kelvin
pt=(10/780)*1.013*10^5;        //total pressure in
    pascal

    //part (ii)
//(ya-ya1)/(ya2-ya1)=(z-z1)/(z2-z1);
z2=1;                          //diffusion path in m
    at pos.2
z1=0;                          //diffusion path in m
    at pos.1
z=.75;                        //diffusion at general z
pa=poly([0],'pa');           //calc. of conc. in gas
    phase
x=roots((pa-pa1)/(pa2-pa1)-(z-z1)/(z2-z1));
printf("\n partial pressure of co2 at o.75m from the
     end where partial pressure is 190mmhg is:%f mmHg
    ",x);
//end
```

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

```

clear;
clc;
printf("\t Example 2_11_a\n");
ya1=0.2;                            //initial mole
    fraction
ya2=0.1;                            //final mole
    fraction
```

```
T=(298);                                    //temperature in kelvin
pt=1*1.013*10^5;                            //total pressure in pascal
z=0.2*10^-2;                                //gas film thickness in m
Dab=.215*10^-4;                             //diffusion coefficient in m^2/s
R=8314;                                     //universal gas constant
//part (i)when N2 is non diffusing

Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T);     //diffusion flux in kmol/m^2*s
printf("\n  diffusion flux if N2 is non diffusing : %f *10^-5 kmol/m^2*s ",Na/10^-5);

//end
```

Scilab code Exa 2.11.b diffusion flux of oxygen

```

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

```
R=8314;                              //universal gas
   constant

//part (ii) equimolar counter diffusion

Na=Dab*pt*(ya1-ya2)/(z*R*T)                    //
   diffusion flux in kmol/m^2*s
printf("\n  diffusion  flux of oxygen during
   equimolar counter-diffusion :%f *10^-5 kmol/m^2*s
    ",Na/10^-5);

//end
```

Scilab code Exa 2.12 diffusion flux through inert air

```

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

```
Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T);     //diffusion flux in kmol/m^2*s
mw=17;                                      //molecular weight of ammonia
massflux=Na*mw;                             //mass flux of given NH3
printf("\n  diffusion flux when total presssure is 1 atm and air is non-diffusing :%f *10^-4 kg/m^2*s ",massflux/10^-4);
   //part (ii) when pressure is increased to 10atm

//Dab_1/Dab_2=pt_2/pt_1
pt_2=10;                                    //final pressure  in atm
pt_1=1;                                     //initially pressure was 1atm
Dab_1=.185;                                 //initially diffusion coefficient was.185
Dab_2=Dab_1*pt_1/pt_2;                      //for gases Dab is proportional to 1/pt
Dab=Dab_2*10^-4;                            //new diffusion coefficient
pt=pt_2*1.013*10^5;                         //new total pressure
Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T);     //diffusion flux in kmol/m^2*s
printf("\n  diffusion  flux when pressure is increased to 10atm :%f *10^-5 kmol/m^2*s ",Na/10^-5);
printf("\n \n so the rate of diffusion remains same on increasing the pressure");
//end
```

Scilab code Exa 2.13 diffusion rate of acetic acid

```

clear;
clc;
printf("\t Example 2.13\n");
//position 1          moles         molefraction         weight
//   acetic acid     0.15            0.0288              9
//   water             5             0.9712              91

//position 2           moles        molefraction         weight
//     aceitic acid      0.05            0.0092                4
//   water                 5.389         0.9908         96
T=290;                             //temperature in kelvin
z=2*10^-3;                         //film thickness sorrounding the water
xa2=0.0092;                        //mole fraction of ethanol at pos.2
xa1=0.0288;                        //mole fraction of ethanol at pos.1
w1=60;                             //molecular weight of acetic acid
w2=18;                             //molecular weight of water
Dab=0.95*10^-9;                    //diffusivity of acetic water sol.in m^2/s
    //av=d/m
Mavg1=xa1*w1+(1-xa1)*w2;           //average molecular wght of solution at pos 1
d1=1012;                           // density of 10 % acid
av1=d1/Mavg1;                      //value of (d/m) of copper solution

    //for position 2
```

```
d2=1003;                      //density of 4% acid
Mavg2=xa2*w1+(1-xa2)*w2;      //average molecular wght of solution at pos.2
av2=d2/Mavg2;                 //value of (d/m) of water

allavg=(av1+av2)/2;          //average value of d/m

        //assuming water to be non diffusing

Na=Dab*(allavg)*log((1-xa2)/(1-xa1))/z;     //diffusion rate of acetic acid aacross film of non diffusing water sol.
printf("\n diffusion rate of acetic acid aacross film of non diffusing water sol. :%f *10^-7 kmol/m^2*s",Na/10^-7);
//end
```

Scilab code Exa 2.14 diffusion flux of nitrogen

```

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

```
coefficient in m^2/s
R=8314;                              //universal gas constant
        //for ideal gases volume fraction =mole fraction
//part (i)when N2 is non diffusing

Na=Dab*pt*log((1-ya2)/(1-ya1))/(z*R*T);     //diffusion flux in kmol/m^2*s
printf("\n  diffusion flux if N2 is non diffusing : %f *10^-5 kmol/m^2*s ",Na/10^-5);
//part (ii) equimolar counter diffusion

Na=Dab*pt*(ya1-ya2)/(z*R*T)                //diffusion flux in kmol/m^2*s
printf("\n  diffusion  flux of nitrogen during equimolar counter-diffusion :%f *10^-5 kmol/m^2*s ",Na/10^-5);

//end
```

Scilab code Exa 2.15 diffusion rate of loss of benzene

```

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

```
z=10*10^-3;                       //gas film thickness in m
Dab=.02/3600;                     //diffusion coefficient in m^2/s
R=8314;                           //universal gas constant
                                  //benzene is stored in atank of dia 10m
//part (i)when air is assumed to be stagnant

Na=Dab*pt*log((pt-pa2)/(pt-pa1))/(z*R*T);     //diffusion flux in kmol/m^2*s
rate=Na*(3.14*r^2);                           //rate of loss of benzene if air is stagnant
printf("\n  diffusion rate of loss of benzene :%f *10^-4 kmol/s ",rate/10^-4);
//end
```

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

```

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

```
    constant
Area=10;                              //area of the film is 10m^2

        //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
        //for gase Dab=T^3/2
        //Dab1/Dab2=(T1/T2)^3/2

T2=370;                                                  //final temperature in kelvin
T1=298;                                                 //initial temperature in kelvin
Dab1=.15*10^-2;                                        //initial diffusion coefficient
Dab2=((T2/T1)^(3/2))*Dab1;                            //final diffusion coefficient
Na=Dab2*pt*log((1-ya2)/(1-ya1))/(z*R*T);     //diffusion flux in kmol/m^2*s
rate=Na*3600*46*Area;                           //rate of diffusion of alcohol-water vapour in kg/hour
printf("\n  rate of diffusion of alcohol-water vapour  :%f *10^6 kg/hour ",rate/10^6);

//end
```

Scilab code Exa 2.17 diffusion rate of ammonia

```

clear;
clc;
printf("\t Example 2.17\n");
ya2=0;                              //molefraction at pos
```

```
.2
ya1=0.1;                        //molefraction at pos.1
T=(273);                        //temperature  in kelvin
pt=1*1.013*10^5;                //total pressure in pascal
z=2*10^-3;                      //gas film thickness in m
Dab=.198*10^-4;                 //diffusion coefficient in m^2/s
R=8314;                         //universal gas constant
        //ammonia is diffusing through an inert film 2mm thick

        //for gase Dab=T^3/2
        //Dab1/Dab2=(T1/T2)^3/2
T2=293;                              //final temperature in kelvin
T1=273;                              //initial temperature in kelvin
Dab1=0.198*10^-4;                    //initial diffusion coefficient
Dab2=((T2/T1)^(3/2))*Dab1;          //final diffusion coefficient
Na=Dab2*pt*log((1-ya2)/(1-ya1))/(z*R*T2);     //diffusion flux in kmol/m^2*s
printf("\n  flux of diffusion of ammonia through inert film :%f *10^-5 kmol/m^2*s ",Na/10^-5);

//if pressure is also incresed from 1 to 5 atm
       //for gases Dab=(T^3/2)/pt;
        //Dab1/Dab2=(T1/T2)^3/2*(p2/p1)
T2=293;                         //final temperature in kelvin
T1=273;                         //initial temperature in kelvin
pa2=5;                          //final pressure in atm
pa1=1;                          //initial pressure in atm
p=pa2*1.013*10^5;
Dab1=.198*10^-4;                                     //
```

```
initial diffusion coefficient
Dab2=((T2/T1)^(3/2))*Dab1*(pa1/pa2);          //final diffusion coefficient
Na=Dab2*p*log((1-ya2)/(1-ya1))/(z*R*T2);     //diffusion flux in kmol/m^2*s
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);
printf("\n \n so there is no change in flux when pressure is changed");
//end
```

Scilab code Exa 2.18 rate of evaporation

```

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

```
thickness 2mm

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

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

```

clear;
clc;
printf("\t Example 2.19\n");
                          //stefan tube experiment

Ml=92;                          //molecular weight of toluene
T=(312.4);                      //temperature  in kelvin
pt=1*1.013*10^5;                //total pressure in pascal
R=8314;                         //universal gas constant
t=275*3600;                     //after 275 hours the level dropped to 80mm from the top
zo=20*10^-3;                   //intially liquid toluene is at 20mm from top
zt=80*10^-3;                   //finally liquid toluene is at 80mm from top
  //air is assumed to be satgnant
d=850;                        //density in kg/m^3
```

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

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

```

clear;
clc;
printf("\t Example 2.20\n");
                              //this is the case of
                                 equimolar counter diffusion
                                  as the latent heat of
                                 vaporisation are very close
                                  to each other

T=(360);                                  //temperature
```

```
in kelvin
pt=372.4/760;                              //total pressure in atm
R=82.06;                                   //universal gas constant
Dab=0.0506;                                //diffusion coefficient in cm^2/s
z=0.254;                                   //gas layer thickness in cm
vp=368/760;                                //vapour pressure of toluene in atm
xtol=.3;                                   //mole fractoin of toluene in atm
pb1=xtol*vp;                               //partial pressure of toluene
//since pb1 is .045263 bt in book it is rounded to 0.145
pb2=xtol*pt;                               //parial pressure of toluene in vapour phase
Na=Dab*(pb1-pb2)/(z*R*T);    //diffusion flux
printf("\n the diffusion flux of a mixture of benzene and toluene %f*10^-8 gmol/cm^2*s\n",Na/10^-8);
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)")
//end
```

Scilab code Exa 2.21 diffusivity of the mixture in stefan tube

```

clear;
clc;
printf("\t Example 2.21\n");
```

```
                    //stefan tube experiment(pseudo
                       steady state diffusion)

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

```
//end
```

Scilab code Exa 2.22 diffusivity of ccl4

```

clear;
clc;
printf("\t Example 2.22\n");
                    //variation in liquid level with respect to time is given below

t=[26 185 456 1336 1958 2810 3829 4822 6385]
// let Zt-Zo= x;
x=[.25 1.29 2.32 4.39 5.47 6.70 7.38 9.03 10.48]
i=1;                //looping starts
while(i<10)
    y(i)=t(i)/x(i);                //for calculating the t/Zt-Zo value
i=i+1;
end
plot(x,y,"o-");
xtitle(" Fig.2.2 Example 22 ","X--(zi-zo),cm  --->", "Y-- vs (t/(zi-zo))min/cm ---->");
slope=51.4385*60 *10^4;             //slope of the curve in 1/sec*m^2
//slope = Cal *(xblm)/(2*Dab*C*(xa1-xa2))
d=1540;                         //density in kg/m^3
Ml=154;                         //molecular weight of toluene
Cal=d/Ml ;                      //conc. at length at disxtance l in mol/m^3

T=(321);                       //temperature  in kelvin
```

```
pt=1;                              //total pressure in atm
R=82.06;                           //universal gas constant
C=pt/(R*T) *10^3;                      //total conc. in kg mol/m^3

pa=(282/760);                     //vapour pressure of toluene
xa1=pa/pt;                        //mole fraction of toluene at pt1 i.e before evaporation
xb1=1-xa1;                       //mole fraction of air before evaporation i.e at pt1
xb2=1;                           //mole fraction of air after evaporation i.e at pt.2
xa2=0;                           //mole fraction of toluene at point 2
xblm=(xb2-xb1)/(log(xb2/xb1));//log mean temp. difference
Dab = Cal *(xblm)/(2*slope*C*(xa1-xa2));     //diffusivity coefficient
printf("\n the diffusivity of the mixture by winklemann method  of toluene in air is :%f*10^-6 m^2/s",Dab/10^-6);
//end
```

Scilab code Exa 2.23 rate of transfer of nitrogen and hydrogen

```

clear;
clc;
printf("\t Example 2.23\n");
                    //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
```

```
d=0.001;
area=3.14*(d/2)^2;      //area of the bulb
T=298;                   //temperature in kelvin
p=1.013*10^5;           //total pressure of both the
    bulbs
R=8314;                 //universal gas constant
c=p/(R*T);              //total concentration
Dab=.784*10^-4;        //diffusion coefficient in m^2/
    s
xa1=0.8;               //molefraction of nitrogen gas
    at the 1 end
xa2=0.25;              //molefraction of nitrogen gas
    at the 2nd end
z=.15;                 //distance between the bulbs

    //rate=area*Na;
rate=area*Dab*c*(xa1-xa2)/z;    //rate of transfer
    of hydrogen and hydrogen
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);
//end
```

Scilab code Exa 2.24 diffusivity of methanol in carbon tetrachloride

```

clear;
clc;
printf("\t Example 2.24\n");
            //using wilke and chang empirical
                correlation
//Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6);

T=288;                          //temperature in
    kelvin
```

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

Scilab code Exa 2.25 diffusivity of methanol in water

```

clear;
clc;
printf("\t Example 2.25\n");
            //using wilke and chang empirical
                correlation
//Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6);

T=288;                              //temperature in
    kelvin
Mb=18;                              //molecular weight
    of methanol
phi=2.26;                            //association
    factor for solvent
va=(2*14.8+(6*3.7)+7.4)*10^-3        //solute(water)
    volume at normal BP in m^3/kmol
u=1*10^-3;                         //viscosity of
    solution in kg/m*s
Dab=(117.3*10^-18)*(phi*Mb)^0.5*T/(u*va^0.6);      //
```

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

Scilab code Exa 2.26 rate of passage of hydrogen

```

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

```

```

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

Chapter 3

Mass transfer coefficient and interphase mass transfer

Scilab code Exa 3.1 rate of sublimation

```

clear;
clc;
printf("\t Example 3.1\n");
v=6;        //velocity in m/s
l=6;        //length in m
pa1=10;      //pressure at 1 in atm
pa2=0;      //pressure at 2 in atm
t=373;      // temperature in kelvin
p=1;      //pressure of naphthalene in atm at 373
    kelvin
D=5.15*10^-6;     //diffusivity of naphthalene in C02
     in m^2/s
d=0.946;         //density of air in kg/m^3
u=.021*10^-3;           //viscosity of air in Newton*s
    /m^2
ID=0.075;        //diameter in m
nre=(ID*v*d)/(u);     //calc. of reynolds no.
cf=2*0.023*(nre)^(-0.2);         //friction factor
```

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

Scilab code Exa 3.2 rate of sublimation of solid naphthalene

```

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

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

Scilab code Exa 3.3 rate of absorption and mass transfer coefficient

```

clear;
clc;
printf("\t Example 3.3\n");
        // a is CO2 and b is water
p=2;                 //total pressure at 1 in atm
pa1=0.2*10^5;        //pressure of CO2 at pt 1 in atm
pa2=0;               //pressure of CO2 at pt 2 is 0
    since air is pure
ya1=0.1;             //mole fraction of CO2 at 1 is
    0.2/2
ya2=0;               //mole fraction of CO2 at 2 is 0
    since air is pure
yb1=0.9;             //mole fraction of water at 1 is
    (1-0.1)
yb2=1.0;             //mole fraction of water at 2 is
    1.0 since total pressure has to be constant.
k_y1=6.78*10^-5;          //mass transfer coefficient
    in kmol/m^2*s*molefraction

yb_ln=(yb2-yb1)/(log(yb2/yb1));     //log mean is
    represented by yb_ln

```

```
k_y=k_y1/yb_ln;
printf("\nvalue of mass transfer coefficient k_y is:%f *10^-5 kmol/m^2*s*(molefractin)",k_y/10^-5);
k_g=k_y/p;                              //mass ttransfer coefficient in lmol/m^2*s*atm
printf("\nvalue  mass transfer coefficient k_g is:%f *10^-5 kmol/m^2*s*(atm)",k_g/10^-5);

NA=k_y*(ya1-ya2);                       //mass flux in kmol/m^2*s
printf("\nvalue of rate of mass transfer :%f *10^-6 kmol/m^2*s",NA/10^-6);
//end
```

Scilab code Exa 3.4 mass transfer coefficient and film thickness

```

clear;
clc;
printf("\t Example 3.4\n");
NA=7.5*10^-7;          //mass flux in gmol/cm^2*s
Dab=1.7*10^-5;        //diffusivity if SO2 in water in cm^2/s
c=1/18.02;            //concentration is density/molecular weight in gmol/cm^2*s
    //SO2 is absorbed from air into water

xa1=0.0025;           //liquid phase mole fraction at 1
xa2=0.0003;           //liquid phase mole fraction at 2
    //NA=kc(Ca1-Ca2)=Dab*(Ca1-Ca2)/d

k_c=NA/(c*(xa1-xa2));     //k_c=Dab/d=NA/c(xa1-xa2)
printf("\nmass transfer coefficient k_c is:%f cm/s",k_c);

```

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

Scilab code Exa 3.5 value of liquid and gas film coefficient

```
clear;
clc;
printf("\t Example 3.5\n");
Kg=2.72*10^-4;              //overall gas phase mass transfer coefficient in kmol/m^2*S*atm
r_gas=0.85*(1/Kg);          //given that gas phase resisitance is 0.85 times overall resistance
kg=1/r_gas;
m=9.35*10^-3;               //henry's law constant in atm*m^3/kmol
kl=m/(1/Kg-1/kg);           //liquid phase mass transfer coefficient in m/s
printf("\nthe value of liquid film coefficient kl : %f*10^-5 m/s",kl/10^-5);
printf("\nthe value of gas film coefficient kg : %f*10^-5 m/s",kg/10^-5);
p=1;                        //overall pressure in atm

        //NA=Kg(pag-pa*)=kg(pag-pai)=kl(Cai-Cal)
Yag=0.1;                    //molefraction of ammonia
Cal=6.42*10^-2;             //liquid phase concentration
Pag=Yag*p;                  //pressure of ammonia
         //Pai and Cai indicates interfacial pressure and conc.
        //Pal and Cal indicates bulk pressure and conc.

        //Pai=m*C_ai;
        //NA=kg(pag-pai)=kl(Cai-Cal)
```

```

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

Scilab code Exa 3.6 concentration of ammonia and interfacial partial pressure

```
clear;
clc;
printf("\t Example 3.6\n");
//kg/kl=0.9=t
//Pai=0.3672*Cai      so;
m=0.3672;
t=.9;
        //Pai and Cai indicates interfacial pressure and conc.
        //Pal and Cal indicates bulk pressure and conc.
Yag=0.15;                    //molefraction of ammonia
Cal=0.147;             //liquid phase concentration in kmol/m^3
p=1;                         //overall pressure
Pag=Yag*p;                   //pressure of ammonia

        //Pai=m*C_ai;
        //kg/kl=(Cai-Cal)/(Pag-Pai);

Cai=poly([0],'Cai');          //calc. of conc. in gas
```

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

Scilab code Exa 3.7 diffusivity of gas overall transfer cofficient

```

clear;
clc;
printf("\t Example 3.7\n");
D=.1;
l=3;                // l is length of bubble in cm
a=3.14*D*l;         // area in cm^2
Ca_o=0.0001;       //pure conc. of gas in g*mol/cc*atm
Ca=0;
NA=.482*10^-5;      // molar rate of absorption in g*moles/s
        //Pa_o and Ca_o indicates pure pressure and conc.
kl=NA/(a*(Ca_o-Ca));     //mass transfer coefficient acc. to higbie's penetration theory
Q=4;                        //volumetric flow rate in cc/s
A=3.14*.1*.1/4;            //area of flow
v=Q/A;                     //velocity of flow in cm/s

//timt t=bubble length/linear velocity;
t=l/v;
DAB=(kl^2)*3.14*t/4;      //diffusivity in cm^2/s
D_new=0.09;                //revised diameter reduced to.09
```

```
a_new=3.14*l*D_new;          //revised area
A_new=3.14*0.09*0.09/4;      //revised flow area
v_new=Q/A_new;              //revised velocity
printf("\nthe value of diffusivity of gas DAB is :%f
    cm/s",DAB/10^-5);

t_new=l/v_new;              //revised time
kl_new=2*(DAB/(3.14*0.0047))^0.5;     //revised mass
    transfer coefficient
NA_new=kl_new*a_new*(Ca_o-Ca);        //revised molar
    rate absorption in g*moles/s
printf("\nthe value of NA_new is  :%f*10^-6 kmol/m^3
    ",NA_new/10^-6);
//end
```

Scilab code Exa 3.8 overall gas phase mass transfer flux

```

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

```
   86.45
m1=86.45;
kx=m1/r_liq;
yag=.1;
xal=(.4/64)/((99.6/18)+(.4/64));
printf("\n film based gas phase mass transfer coeff.
   ky is  :%f ",kx);
//slope of the line gives -kx/ky=-70.61
m2=m1;                                    // since
   equilibrium line a straigth line m'=m''
Kx=1/(1/kx+(1/(m2*ky)));          //overall liquid
   phase mass transfer coefficient
printf("\n overall  liq phase mass transfer
   coefficient Kx is  :%f ",Kx);
// equillibrium relation is given under
p = [0.2 0.3 0.5 0.7];
a = [29 46 83 119];
i=1;
    //looping for calcullating mole fraction
while (i<5)
    x(i)= (p(i)/64)/(p(i)/64+100/18);
    y(i)= a(i)/760;                          //mole
       fraction plotted on y-axis
    i=i+1;                                   //mole
       fraction plotted on x-axis
end
plot(x,y,"o-");
title("Fig.3.17,Example 8");
xlabel("X-- Concentration of SO2 in liquid phase, X
   (10^4)(molefraction)");
ylabel("Y-- Concentration of SO2 in gas phase, Y(
   molefraction)");

    //from the graph we get these values
yao=.083;         //corresponding to the value of xao
   =0.001128
xao=.00132;       //corresponding to the value of yag
   =.1
```

```
yai=.0925;       //corresponding to the perpendicular dropped from the pt(.001128,0.1)
xai=.00123;

    // flux based on overall coefficient
NAo_gas=Ky*(yag-yao);
NAo_liq=Kx*(xao-xal);
printf("\n overall gas phase mass transfer flux -NAo_gas is  :%f*10^-6 kmol/m^2*s ",NAo_gas/10^-6);
printf("\n overall liq phase mass transfer flux -NAo_liq is  :%f*10^-6 kmol/m^2*s ",NAo_liq/10^-6);

    // flux based on film coefficient
NAf_gas=ky*(yag-yai);
NAf_liq=kx*(xai-xal);
printf("\n film based gas phase mass transfer flux-NAf_gas is  :%f *10^-6 kmol/m^2*s",NAf_gas/10^-6);
printf("\n film based liq phase mass transfer flux-NAf_liq is  :%f *10^-6 kmol/m^2*s",NAf_liq/10^-6);
//end
```

Scilab code Exa 3.9 concentration of acid at outlet

```

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

```
u=1.786*10^-4;          //viscosity of air in n*s/m^2
Dia=2.54;          //diameter in cm
nre=(Dia*v*d)/(u);      //calc. of reynolds no.
cf=2*0.036*(nre)^(-0.25);          //friction factor
nsc=(u)/(d*D);          //calc of schmidt no.
kc=(cf*v)/(2*(nsc)^(2/3));          //cf/2=kc/uo*(sc)^2/3

//consider an elelmental section of dx at a distance of x from the point of entry of air.
//let the conc. be c of diffusing component and c+dc at the point of leaving. mass balance across this elelmental gives
                    //rate of mass transfer=(cross sectional area)*(air velocity)*dc
                    //                              =(3.14*d^2/4)*v*dc   -----1 eqn

//flux for mass transfer from the surface=kc*(Cas-C)
//                    rate of mass transfer=(flux)*mass transfer
//                                                  =kc*(Cas-C)*3.14*dx*D------2 eqn
//                              solving ----1 & -----2 we get
//
//                              (3.14*d^2/4)*v*dc=kc*(Cas-C)*3.14*dx*d;
//                              dc/(Cas-C)=(kc*3.14*d*v)/(3.14*d^2/4)*dx
//                              solving this we get
//                              ln[(Cas-C)/(Cas-C_in)]=(kc*4*x)/(d*v)

x=183;              //upper limit of x
C_in=0;              //C=C_in=0;
t=(kc*4*x)/(Dia*v);     //variable to take out the
```

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

Chapter 5

Humidification

Scilab code Exa 5.1 properties through humidity chart

```

clear;
clc;
printf("\t Example 5.1\n");
            //dry bulb temperature=50  and  wet bulb temperature=35
Tg=50;                  //dry bulb temperature=50
To=0;                   //refrence temperature in degree celcius
Mb=28.84;               //average molecular weight of air
Ma=18;                  //average molecular weight of water

//part(i)
ybar=.0483               //0.003 kg of water vapour /kg of dry air
printf("\n the humidity(from chart) is  \t\t:%f percent",ybar);

//part(ii)
```

```
humper=35;                    //humidity percentage
printf("\n the percentage humidity is(from chart) : %f  percent",humper);

//part(iii)
pt=1.013*10^5;                //total pressure in pascal
molhum=0.0483;                //molal humidity =pa/(pt-pa)
pa=molhum*pt/(1+molhum);
//the vopour pressure of water(steam tables)at 50 degree = .1234*10^5 N/m^2
relhum=(pa/(.1234*10^5))*100;        //percentage relative humidity =partial pressure/vapour pressure
printf("\n the percentage relative humidity is \t percent:%f  ",relhum);

//part(iv)
dewpoint=31.5;                //dew point temperature in  degree celcius
printf("\n the dew point temperature \t\t :%f degree  celcius",dewpoint);

//part(v)
Ca=1.005;
Cb=1.884;
ybar=.03;                    //saturation temperature inkg water vapour/kg dry air
Cs=Ca+Cb*ybar;                //humid heat  in kj/kg dry  air degree celcius
printf("\n we get humid heat as \t\t\t :%f kj/kg dry  air degree celcius ",Cs);

//part(vi)
d=2502;                        //latent heat in kj/kg
H=Cs*(Tg-0)+ybar*d;            //enthalpy for refrence temperature of 0 degree
printf("\n we get H as \t\t\t\t :%f kj/kg",H);
```

```
Hsat=274;                    //enthalpy of sturated air
Hdry=50;                     //enthalpy of dry air in kj/kg
Hwet=Hdry+(Hsat-Hdry)*0.35;       //enthalpy of wet air in kj/kg
printf("\n we get enthalpy of wet air  as \t:%f kj/kg",Hwet);

//part(vii)
VH=8315*[(1/Mb)+(ybar/Ma)]*[(Tg+273)/pt];         //humid volume in m^3mixture/kg of dry air
printf("\n we get VH  as (a)\t\t\t :%f m^3/kg of dry air",VH);
spvol=1.055;                 //specific volume of saturated air in m^3*kg
vdry=0.91;                   //specific volume of dry air in m^3/kg
Vh=vdry+(spvol-vdry)*.35  //by interpolation we get Vh in m^3/kg of dry air
printf("\n by interpolation we get specific volume Vh as(b) :%f m^3/kg of dry air",Vh);

//end
```

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

```

clear;
clc;
printf("\t Example 5.2\n");
            //dry bulb temperature=25  and  wet bulb temperature=22
Tg=25;                       //dry bulb temperature=50
To=0;                        //refrence temperature in
```

```
degree celcius
Mb=28.84;                //average molecular weight of air
Ma=18;                   //average molecular weight of water

//part(i)
hum=.0145                //0.0145 kg of water/kg of dry air
printf("\n the saturation humidity(from chart) is : %f  percent",hum);

//part(ii)
humper=57;                //humidity percentage
printf("\n the percentage humidity is  \t\t:%f percent",humper);

//part(iii)
pt=1;             //total pressure in atm
sathum=0.0255;            //molal humidity =pa/(pt−pa)
pa1=sathum*pt*(28.84/18)/(1+(sathum*(28.84/18)));
//the vopour pressure of water(steam tables)at 25 = .0393*10^5 N/m^2
pt=1;                     //total pressure in atm
molhum=0.0145;            //molal humidity =pa/(pt−pa)
pa2=molhum*pt*(28.84/18)/(1+(molhum*pt*(28.84/18)));
//the vopour pressure of water(steam tables)at 25 = .0393*10^5 N/m^2
relhum=(pa2/pa1)*100;      //percentage relative humidity =partial pressure/vapour pressure
printf("\n the percentage relative humidity is \t : %f  ",relhum);

//part(iv)
dewpoint=19.5;            //dew point temperature in degree celcius
```

```
printf("\n the dew point temperature \t :%f degree celcius",dewpoint);

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

Scilab code Exa 5.3 properties of nitrogen oxygen vapour mixture

```

clear;
clc;
printf("\t Example 5.3\n");
//part(i)
pt=800;                        //total pressure in mmHg
pa=190;                        //vapour pressure of acetone at 25 degree
ys_bar=pa*(58/28)/(pt-pa)    //
//percentage saturation = y_bar/ys_bar *100
s=80;                          //percent saturation
y_bar=ys_bar*s/100;            //absolute humidity
printf("\n the absolute humidity is \t :%f kg acetone/kmol N2 ",y_bar);

```

```
//part(ii)
//y_bar=pa*(58/28)/(pt-pa)
pa1=pt*y_bar*(28/58)/(1+(y_bar*(28/58)));
printf("\n the partial pressure of acetone is:%f mmHg",pa1);

//part(iii)
y=pa1/(pt-pa1);                    //absolute molal humidity
printf("\n absolute molal humidity \t:%f kmol acetone/kmol N2",y);

//part(iv)
//volume of .249kmol acetone vapour at NTP =.249*22.14
//p1v1/T1 =p2v2/T2
p2=800;                           //final pressure of acetone and nitrogen at 25 degree
p1=760;                           //initial pressure of acetone and nitrogen at 25 degree
T2=298;                           //final temperature of acetone and nitrogenat 25 degree
T1=273;                          //initial temperature of acetone and nitrogen at 25 degree
vA1=5.581;                           //initial volume of acetone at 25 degree
vN1=22.414;                          //initial volume of nitrogen at 25 degree
vA2=T2*vA1*p1/(T1*p2);               //final volume of acetone at 25 degree
vN2=T2*vN1*p1/(T1*p2);               //final volume of nitrogen at 25 degree
vtotal=vA2+vN2;                   //total volume of the mixture
vper=vA2*100/vtotal;                 //percentage volume of acetone
printf("\n the percentage volume of acetone is :%f m^3",vper);
```

```
//end
```

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

```

clear;
clc;
printf("\t Example 5.4\n");

//part(i)
pa=13.3;                              //pressure in kpa
pa2=20.6;                            //vapour pressure at 60 degree
pt=106.6                             //total pressure in kpa
y=pa/(pt-pa);                      //absolute molal humidity
y_bar=y*(18/28.84);                //relative humidity
printf("\n absolute  humidity of mixture :%f kg water-vapour/kg dry air",y_bar);


//part(ii)
mf=pa/pt;                            //mole fraction
printf("\n the mole fraction is :%f",mf);

//part(iii)
vf=mf;                                        //volume fraction
printf("\n the volume fraction is :%f",vf);

//part(iv)
Ma=18;                              //molecular weight
Mb=28.84;                            //molecular weight
Tg=60;                               //temperature of mixture
rh=(pa/pa2)*100;                   //relative humidity in pecentage
```

```
printf("\n we get relative humidity as  as :%f percent",rh);

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

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

```

clear;
clc;
printf("\t Example 5.5\n");

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

```
air",VH);

//part(ii)
Ca=1005;
Cb=1884;
Cs=Ca+Cb*y_bar;                //humid heat  in j/kg dry
     air degree celcius
printf("\n we get humid heat as \t\t :%f j/kg dry
   air degree celcius ",Cs);

//part(iii)
d=2502300;                         //latent heat in j/kg
H=Cs*(Tg-0)+y_bar*d;           //enthalpy for refrence
   temperature of 0 degree
printf("\n we get Enthalpy H as \t\t:%f j/kg dry air
   ",H);

//part(iv)
dewpoint=23.5;                 //dew point temperature in
     degree celcius
printf("\n the dew point temperature \t :%f degree
   celcius",dewpoint);

//end
```

Scilab code Exa 5.6 properties when dry bulb temperature is 55

```

clear;
clc;
printf("\t Example 5.6\n");

//part(i)
y=.048;                           //humidity kmol water
   vapour/kmol dry air
```

```
y_bar=y*(18/28.84);            //(from chart) absolute humidity
printf("\n we get absolute humidity as  :%f kg of water/kg dry air",y_bar);
printf("\n we get percentage humidity as(from chart) :%f percent",25.5);
y_bar=y*(18/28.84);            //relative humidity
Ma=18;                         //molecular weight
Mb=28.84;                      //molecular weight
Tg=55;                         //temperature of mixture
pt=1.013*10^5;                 //total pressure in pascal
VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt];        //humid volume in m^3mixture/kg of dry air
printf("\n we get VH  as \t :%f m^3/kg dry air",VH);

//part(ii)
Ca=1005;
Cb=1884;
Cs=Ca+Cb*y_bar;                //humid heat  in j/kg dry air degree celcius
printf("\n we get humid heat as \t :%f j/kg dry air degree celcius ",Cs);

//part(iii)
d=2502300;                     //latent heat in j/kg
H=Cs*(Tg-0)+y_bar*d;           //enthalpy for refrence temperature of 0 degree
printf("\n we get H as \t :%f j/kg dry air",H);

//end
```

Scilab code Exa 5.7 Calculation of final temperature

```
clear;
clc;
printf("\t Example 5.7\n");
        //given o.03 kg of water vapour/kg of dry
           air is contacted with water at an
           adiabatic temperature and humidified and
           cooled to 70 percent saturtion

//from pyschometric chart
ft=46;                      //final temperature in degree
   celcius
printf("\n final temperature is (from chart):%f
   degree celcius",ft);
y_bar=.0475;            // humidity of air
printf("\n the humidity of air(from chart) :%f kg
   water vapour /kg dry air",y_bar);

//end
```

Scilab code Exa 5.8 Calculation of molal humidity

```

clear;
clc;
printf("\t Example 5.8\n");
pa1=4.24              //data:    vapour pressure of
   water at 30degree = 4.24 kpa
pa2=1.70              //         vapour pressure of
   water at 30degree = 1.70 kpa

//part(i)
pt=100;                               //total pressure
ys_bar=pa1/(pt-pa1);                    //kg water vapour/
   kg dry air
rh=.8;                                //relative humidity
```

```
pa3=rh*pa1;                           //partial pressure
y_bar=pa3*(18/28.84)/(pt-pa3);                    //molal humidity
printf("\n the molal humidity:%f kg/kg dry air",y_bar);

//part(ii)
//under these conditions the air will be saturated at 15 degree as some water is condensed
pa=1.7;
pt=200;
ys=pa/(pt-pa);
ys_bar=ys*(18/28.84);
printf("\n the molal humidity if pressure doubled and temp. is 15 :%f kg/kg dry air",ys_bar);

//part(iii)
Ma=18;                              //molecular weight
Mb=28.84;                           //molecular weight
Tg=30;                              //temperature of mixture
rh=(pa/pa2)*100;                    //relative humidity in pecentage
pt=10^5;                   //total pressure in pascal
VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt];          //humid volume in m^3mixture/kg of dry air
printf("\n we get humid volume VH  as \t :%f m^3/kg of dry air",VH);
w=100/VH;                           //100 m^3 of original air
wo= w*y_bar;                        //water present in original air
wf= w*ys_bar;                      //water present finally
wc=wo-wf;                           //water condensed from 100m^3 of original sample
printf("\n the weight water condensed from 100m^3 of original sample:%f kg",wc);

```

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

//end
```

Scilab code Exa 5.9 Calculation of relative humidity and humid volume

```
// Calculation of relative humidity ,humid volume enthalpy and heat required if 100m^3 of this air is heated to 110 degree
clear;
clc;
printf("\t Example 5.9\n");

//part(i)
y_bar=.03;                          // humidity inkg water /kg dry air
pt=760;                             //total pressure in pascal
pa2=118;                            //final pressure
y=y_bar/(18/28.84);               //humidity kmol water vapour/kmol dry air
pa=(y*pt)/(y+1);                  //partial pressure
rh=pa/pa2;                          //relative humidity
sh=pa2/(pt-pa2);                    //saturated humidity
ph=(y/sh)*100;                        //percentage humidity
printf("\n percentage humidity is :%f",ph);
```

```

///part(ii)
Ma=18;                              //molecular weight
Mb=28.84;                            //molecular weight
Tg=55;                               //temperature of mixture
pt=1.013*10^5;                   //total pressure in pascal
VH=8315*[(1/Mb)+(y_bar/Ma)]*[(Tg+273)/pt];           //humid volume in m^3mixture/kg of dry air
printf("\n we get VH humid volume as  :%f m^3/kg dry air",VH);


//part(iii)
Ca=1005;
Cb=1884;
Cs=Ca+Cb*y_bar;                        //humid heat  in j/kg dry air degree celcius
printf("\n we get humid heat as \t :%f j/kg dry air degree celcius ",Cs);
d=2502300;                            //latent heat in j/kg
H=Cs*(Tg-0)+y_bar*d;               //enthalpy for refrence temperature of 0 degree
printf("\n we get H enthalpy as \t :%f j/kg",H);

//part(iv)
v=100;                                //volume of air
mass=v/VH;                            //mass of dry air
Tg=110;                               //temperature of mixture
d=2502300;                            //latent heat in j
H_final=Cs*(Tg-0)+y_bar*d;        //enthalpy for refrence temperature of 0 degree
H_added=(H_final-H)*102.25;        //HEAT added  in kj
printf("\n we get heat added as \t :%f kj",H_added/1000);
//end
```

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

```

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

```
i=i+1;
end                     //end of lop
ES=[60.735 101.79 166.49 278.72 354.92]          //
   enthalpy of eqll condition

plot(T,E,"o--");
plot(T,ES,"+-");
title("Fig.5.10(b),Temperature-Enthalpy plot");
xlabel("X-- Temperature, degree celcius");
ylabel("Y-- Enthalpy ,kj/kg");
legend("operating line","Enthalpy at saturated cond"
   )

//locate (30,71.09) the operating conditions at the
   bottom of the tower and draw the tangent to the
   curve
Hg1=71.09;                      //point on the oper.
   line(incoming air)
Hg2=253;                        //point after drawing
    the tangent
slope=(Hg2-Hg1)/(T1-T2);        //we gt slope of the
   tangent
    //slope = (L*Cl/G)_min
Cl=4.18;
G_min=L*60*Cl/slope;                      //tangent
   gives minimum value of the gas flow rate
G_actual=G_min*1.3;                      //since actual
   flow rate is 1.3 times the minimum
slope2=L*Cl*60/G_actual;                //slope of
   operating line
Hg2_actual=slope2*(T1-T2)+Hg1;      //actual
   humidityat pt 2
Ggas=10000;                          //minimum gas rate
    in kg/hr*m^2
Area1=G_actual/Ggas;               //maximum area of
   the tower(based on gas)
Gliq=12000;                         //minimum liquid
   rate in kg/hr*m^2
```

```
Area2=60*L/Gliq;                    //maximum area of the tower(based on liquid)
printf("\n \n the maximum area of the tower(based on gas) is :%f m^2",Area1);
printf("\n the maximum area of the tower(based on liquid) is :%f m^2",Area2);
dia=(Area1*4/3.14)^0.5;             //diameter of the tower in m

//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 equl. line we get conditions of gas and equl. values are tabulated below as follows


//table
T=[20 30 40 50 55]                 //differnt temperature for different enthalpy calculation
//enthaly
H_bar=[101.79 133.0 166.49 210.0 278.72]          //H_bar i.e. at equl.
Hg=[71.09 103.00 140.00 173.00 211.09]           //Hg i.e. of operating line
i=1;
while(i<6)            //looping for different enthalpy calculation of operating line
y(i)=1/(H_bar(i)-Hg(i));
printf("\n the enhalpy at :%f is :%f",T(i),y(i));
i=i+1;
end                   //end of lop
xset('window',1);
plot(Hg,y,"o-");
xtitle(" Fig.5.10(c) Example 10 (1/(Hf-Hg)) vs Hg","X-- Hg --->","Y-- 1/(Hf-Hg) ---->");

//area under this curve gives Ntog =4.26
```

```
Ntog=4.26;                  //no. of transfer unit
Gs=10000;                    //gas flow rate
Htog=Gs/Ky_a;                // height of transfer unit
height=Ntog*Htog;            //height of the tower
printf("\n \nthe tower height is :%f m",height);


//make up water is based onevaporation loss(E),blow down loss(B),windage loss(W)                    M = E + B + W
W=.2/100 *L*60;                 //windage loss(W)
B=0;                          //blow down loss neglected
E=G_actual*(.064-.016);       //assuming air leaves fully saturated
M = E + B + W;                  //make up water is based onevaporation loss(E),blow down loss(B),windage loss(W)
printf("\n make up water is based onevaporation loss(E),blow down loss(B),windage loss(W) is :%f kg/hr",M);
//end
```

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

```

clear;
clc;
printf("\t Example 5.11\n");
//air leaves at 19 degree at fully saturated condition

T1=30;                //temperature at the inlet in
```

```
degree celcius
T2=17;                //temperature at the exit in degree celcius
f=100000;             //flow rate of water in kg/hr
hi=.004;              //humidity of incoming air in kg/kg of dry air
hl=.015;              //humidity of leaving air in kg/kg of dry air
Hi=18.11;             //enthalpy of incoming air in kg/kg of dry air
Hl=57.16;              //enthalpy of leaving air in kg/kg of dry air
//w=mdry*(hl-hi) = mdry*0.011;
                                    ------equn 1st
//mass of water evaporated

//making energy balance:  total heat in = total heat out
//heat in entering water + heat in entering air = heat in leaving water + heat in leaving air
//100000*1*(30-0) + mdry*Hi = (100000-w)*1*(17-0) + mdry*Hl  ----eqn 2nd

//substituting eqn 1st in 2nd we get;
a=14.4;                                       //cross sectional area of the tower in m^2
mdry=(T1*f-T2*f)/(Hl-Hi-T2*.011);        //mass of dry air
velocity=mdry/a;                              //air velocity in kg/m^2* hr
x=mdry*.011;                                 //make up water needed in kg/hr
printf("\n the make up water needed is :%f kg /hr",x);
printf("\n the velocity of air is as :%f kg/hr",velocity);
//end
```

Scilab code Exa 5.12 Calculation of the length of the chamber

```

clear;
clc;
printf("\t Example 5.12\n");
//horizontal spray with recirculated water . air is cooled and humidified to 34 and leaves at 90 percent saturation

T1=65;                    //dry bulb temperature at the inlet in degree celcius
f=3.5;                //flow rate of air in m^3/s
hi=1.017;                  //humidity of incoming air in kg/kg of dry air
hl=.03;                  //humidity of leaving air in kg/kg of dry air
k=1.12;                  //mass transfer coefficient in kg/m^3*s
y1=.017;                 //molefraction at recieving end
y2=.03;                  //molefraction at leaving end

//substituting eqn 1st in 2nd we get;
a=2;                                       //cross sectional area of the tower in m^2
d=1.113;                                       //density o fair in kg/m^3
m=(f*d)                                        //mass flow rate  of  air
gs=m/hi;                                      //air velocity in kg/m^2* hr
ys_bar=.032;
//for recirculation humidifier
z=log((ys_bar-y1)/(ys_bar-y2))*gs/k;            //
```

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

//end
```

Chapter 6

Drying

Scilab code Exa 6.1 Calculation of the solid temp

```

clear;
clc;
printf("\t Example 6.1\n");

    //air leaves the pre-heater of the dryer at 325K

H1=.005;                              //humidity of
   incoming air per kg of dry air
T1=25;                                //wet bulb
   temperature
    //moisture is removed along constant wet bulb
       temp. till 60per R.H is reached
// from the chart ,humidity of ai rleaving first
   shelf =.016 kg water /kg dry air.

//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 .

```

```
    //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 .

//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 doed not correspond toactual one)

printf("\n  the solid temp. correspond to wbt and they are 23, 27,32 and 34  degree respectively");

        //part(ii)
Ybar=.032;                    //kg water/kg dry air// final moist air condotions
T2=42;                        //dry bulb temperature
Mair=28.84;                   //molecular weight of air
Mwater=18;                      //molecular weight of water
pt=1.013*10^5;                //total pressure in pascal
Vh=8315*((1/Mair)+(Ybar/Mwater))*((T2+273)/pt);
r=300;                        //flow rate of moist air leaving the dryer
a=r*60/Vh;                   //amount of dry air leaving /hr
w=a*(Ybar-0.005);             // water removed /hr
printf("\n the water removed /hr is :%fkg /hr",w);

//end
```

Scilab code Exa 6.2 time for drying

```

clear;
clc;
printf("\t Example 6.2\n");
//table    X*100,(kgmoisture/kg dry solid)   N*100 (
    kg moisture evaporated /hr*m^2)

//                   35
                                          30
//                   25
                                          30
//                   20
                                          30
//                   18
                                          26.6
//                   16
                                          23.9
//                   14
                                          20.8
//                   12
                                          18
//                   10
                                          15
//                   9
                                           9.7
//                   8
                                           7
//                   7
                                           4.3
//                   6.4
                                         2.511111


Ls=262.5;                         //mass of bone dry solid
     ais the drying surface
A=262.5/8;                       //both upper surafce and
```

```
lower surface are exposed
Nc=0.3;                        //in kg/m^2*hr
x2=.06;                        //moisture content on wet basis  finally after drying
x1=.25;                        //moisture content on wet basis  finally after drying
Xcr=0.20;                      //crtical moisture content
X1=x1/(1-x1);                              //moisture content on dry basis  intially
X2=x2/(1-x2);                              //moisture content on dry basis  finally after drying
Xbar=0.025;                                //equillibrium moisture

t1=Ls/(A*Nc) *(X1-Xcr);         //so for constant rate period

//for falling rate period we find time graphically
p = [.20 .18 .16 .14 .12 .10 .09 .08 .07 .064];
a = [3.3 5.56 6.25 7.14 8.32 10.00 11.11 12.5 14.29 15.625];

plot(p,a,"o-");
title("Fig.6.18 Example2 1/N vs X for fallling rate period");
xlabel("X--- Moisture content , X(kg/kg)");
ylabel("Y--- 1/N, hr,m^2/kg");

Area=1.116;                     //area under the curve
t2=Area *Ls/A;                  //falling rate period we find time graphically
ttotal=t1+t2;                   //total time for drying
printf("\n the total time for drying the wet slab on wet basis is :%f min",ttotal);
//end
```

Scilab code Exa 6.3.a plots of drying rate curve

```

clear;
clc;
printf("\t Example 6_3_a\n");
    //part(i)
//table wt of wet slab,kg  -- 5.0   4.0   3.6   3.5   3.4  3.06  2.85
//       drying rate,kg/m^2s-- 5.0   5.0   4.5   4.0   3.5  2.00  1.00
//       X,Dry basis          -- 1.0   0.6   .44   0.4   .36  .224  0.14
// equillibrium relation is given under
p = [1.0   0.6   .44   0.4   .36   .224   0.14];
a = [5.0   5.0   4.5   4.0   3.5   2.00   1.00];

i=1;                              //looping for calc. of 1/N
while(i<8)                        //looping begins
t(i)=1/(a(i));
i=i+1;
end                               //as 1/N plot is needed

plot(p,a,"o-");
title("Fig.6.19(a) Example3  Drying Rate curve");
xlabel("X-- Moisture content, X(kg/kg) ----->");
ylabel("Y-- Drying Rate, N(kg/hr.m^2 ----->");
xset('window',1);
plot(p,t,"o-");
title("Fig.6.19(b) Example3 1/N vs X");
xlabel("X-- Moisture content, X(kg/kg) ---->");
ylabel("Y-- 1/N, hr,m^2/kg ---->");
//from X=0.6 to 0.44 ,falling rate is non linear and
```

```
    from X=.44 to .14 falling rate is linear

printf("\n from the graph we get critical moisture
   content as 0.6 kg moisture/kg dry solid");

//end
```

Scilab code Exa 6.3.b total time for drying

```


clear;
clc;
printf("\t Example 6_3_b\n");


    //part(ii)
w1=5;                              //wet of wet solid
c1=.5/(1-.5);                      //moisture content
   per kg wet solid
w2=5*0.5;                          //moisture for 5kg
   wet solid
w3=w1-w2;                          //weight of dry
   solid
xbar=0.05;                         //equillibrium
   moisture content
Xbar=xbar/(1-xbar);                //equillibrium
   moisture content
Ls=2.5;                           //mass of bone dry
   solid ais the drying surface
A=5;                             //both upper surafce
   and lower surface are exposed
Nc=0.6;                           //in kg/m^2*hr
//from X=0.6 to 0.44 ,falling rate is non linear and
    from X=.44 to .14 falling rate is linear
```

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

//end
```

Scilab code Exa 6.4 time for drying the sheets

```

clear;
clc;
printf("\t Example 6.4\n");

//table X— .30   .20   .18   .15   .14   .11   .07   .05
//       N— 1.22  1.22  1.14  .90   .80   .56   .22   .05
//let Ls/A=p
```

```
p=48;                       //mass of bone dry solid ais the drying surface
v=1.5*1.5*.5;               //volume of material
Nc=1.22;                    //in kg/m^2*hr
Xcr=0.2;                    //crtical moisture content
X1=0.25;                  //moisture content on dry basis  intially
X2=0.08;                   //moisture content on dry basis  finally after drying
Xbar=0.025;               //equillibrium moisture

//tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-Xbar)));

t1=p/(Nc) * (X1-Xcr);     //time taken for constant drying rate period
//table X-- .18      .15      .14    .11      .07      .05
//     1/N-- .8772    1.11    1.25    1.7857   4.545    20

// equillibrium relation is given under
p = [.18 .15 .14 .11 .07 .05];
a = [.8772 1.11 1.25 1.7857 4.545 20];

plot(p,a,"o-");
title("Fig.6.20 Example4 1/N vs X for fallling rate period");
xlabel("X-- Moisture content, X(kg/kg) ---->");
ylabel("Y-- 1/N, hr,m^2/kg ---->");

a=14*.025*1;           //area under the curve
t2=a*48;              //time taken for varying drying period
ttotal=t1+t2;         //total time taken
printf("\n total time for drying the material from 25 to 8 percent moisture under same drying conditions is :%f hr",ttotal);

```

```
//end
```

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

```

clear;
clc;
printf("\t Example 6_5_a\n");

//table 6.5.1
//S.NO.          Time (Hr)                    weight of
   wet material(kg)
//                 0.0
                                            5.314
//                 0.4
                                            5.238
//                 0.8
                                            5.162
//                 1.0
                                            5.124
//                 1.4
                                            5.048
//                 1.8
                                            4.972
//                 2.2
                                            4.895
//                 2.6
                                            4.819
//                 3.0
                                            4.743
//                 3.4
                                            4.667
//                 4.2
                                            4.524
//                 4.6
```

```
                                    4.468
//                    5.0                                   4.426
//                    6.0                                   4.340
//                 infinite                             4.120

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]
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]
//part(i)
x=4.120;                               //weight of the dried material
   printf("\n moisture content (dry basis) ");
i=1;                                   //looping starts
while(i<16)                           //calculation of moisture content
  p(i)=(w(i)-x)/x;
  printf("\n      :%f",p(i));
i=i+1;
end
    printf("\n   \n  Drying rate kg/hr*m^2");
i=2;
while(i<15)
     a(i)=(p(i-1)-p(i))*4.12/(t(i)-t(i-1));
     printf("\n       :%f  ",a(i));
     i=i+1;
end
a(1)=.19;
a(15)=0;
printf("\n\n from the above data it is clear that critical moisture content Xcr=0.11");
plot(p,a,"o-");
title("Fig.6.19(a) Example3  Drying Rate curve");
xlabel("X-- Moisture content, X(kg/kg) ---->");
ylabel("Y-- Drying Rate, N(kg/hr.m^2 ---->");
```

```
//end
```

Scilab code Exa 6.5.b amount of air required

```

clear;
clc;
printf("\t Example 6_5_b\n");

//part(ii)
w1=4.934;                          //weight after two
   hours
w0=5.314;                          //initial weight
w2=w0-w1;                          // water evaporated in 2
   hrs
H1=.01;                            //humidty of incoming air
H2=.03;                            //humidity of leaving air
yout=.03;
yin=.01;
Gs=w2/(yout-yin);         //water carried away
printf("\n the amount of air required in 2hours is :
   %f kg",Gs);
//end
```

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

```

clear;
clc;
printf("\t Example 6_5_c\n");

//part(iii)
```

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

Scilab code Exa 6.6 time saved in drying

```

clear;
clc;
printf("\t Example 6.6\n");
            //wooden cloth is dried from 100 to 10 and then th efinal moisture content is changed to 16 percent from 10;

Xcr=0.55;               //crtical moisture content
X1=1;                 //moisture content on dry basis intially
```

```
X2=.1;                //moisture content on dry basis
   finally after drying
Xbar=.06;             //equillibrium moisture
    //since eqn 1 is tobe divided by eqn 2 so let
       the value of Ls/A*Nc be = 1 as it will be
       cancelled
p=1;                   //let Ls/A*Nc be =p
p=poly([0],'p');        //calc. of time 1
tbar=1;                  //since the eqns are
   independent of tbar
t1=roots(tbar- p*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar)))); //———eqn1
X2bar=.16;
p=poly([0],'p');        //calc. of time 2
t2=roots(tbar- p*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2bar-Xbar))));//———eqn2

    //let t1/t2 be = k
k=t1/t2;
ans=1/k-1;             //reduction in time for drying
printf("\n the reduction in time for drying is :%f
   percent",ans*100);
//end
```

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

```

clear;
clc;
printf("\t Example 6.7\n");
            //assume rate of drying in the falling
               rate period is directly prporional to
                the free moisture content

//Ls/ A*Nc is unknown;
```

```
Xcr=0.14;                //crtical moisture content
X1=.3/(1-.3);            //moisture content on dry
   basis intially
X2=0.1/(1-0.1);          //moisture content on dry
   basis finally after drying
Xbar=.04;               //equillibrium moisture
tbar=5;                 //time needed to dry from 30 to
    6 percent on bone dry basis

    //let Ls / A*Nc be = p
p=poly([0],'p');           //calc. of Ls / A*Nc be = p
   value
x=roots(tbar-p * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar))));
printf("\n the value of Ls/ A*Nc is :%f",x);

//new X1 AND X2 are now given as follows
X1=0.3/(1-.3);                //new moisture content
   on dry basis intially
X2=0.064;                   //new moisture content on
   dry basis finally after drying
tbar=x * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-
   Xbar)));
printf("\n the time for drying the sheets from 30 to
    6  percent moisture under same drying conditions
    is :%f hr",tbar);

//end
```

Scilab code Exa 6.8.a time for drying the material

```

clear;
clc;
```

```
printf("\t Example 6_8_a\n");

Ls=1000;                //mass of bone dry solid ais
   the drying surface
A=55;                   //both upper surafce and lower
    surface are exposed
v=.75;                  //velocity of air
Nc=.3*10^-3;            //in kg/m^2*s
x2=.2;                  //moisture content on wet
   basis  finally after drying
Xcr=0.125;               //crtical moisture content
X1=0.15;                //moisture content on dry
   basis  intially
X2=0.025;               //moisture content on dry
   basis  finally after drying
Xbar=0.0;          //equillibrium moisture

tbar=(Ls/(A*Nc))*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar)));

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);

//end
```

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

```

clear;
clc;
printf("\t Example 6_8_b\n");

//tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-
   Xbar)));
```

```
//part(i)
        // assuming only surface evaporation and
           assuming air moves parellel to  surface

//Nc=G^0.71;             G=V*d
//so NC = k* V^.71
Ls=1000;                //mass of bone dry solid ais
   the drying surface
A=55;                   //both upper surafce and lower
    surface are exposed
v=.75;                  //velocity of air
Nc=.3*10^-3;            //in kg/m^2*s
x2=.2;                  //moisture content on wet
   basis  finally after drying
Xcr=0.125;               //crtical moisture content
X1=0.15;                //moisture content on dry
   basis  intially
X2=0.025;               //moisture content on dry
   basis  finally after drying
Xbar=0.0;          //equillibrium moisture
tbar=3.8077;         //time to dry material ,
   calculated from previous part
V1=.75;                  //old velocity
V2=4;                    //new velocity
Nc2=Nc*(V2/V1)^.71;      //in kg/m^2*s
t2=(Ls/(A*Nc2))*((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar)));     //if air velocity is increased
   to 4
t=tbar-t2/3600;              //time saved

printf("\n the time saved , if air velocity is
   increased to 4 m/s: %f",t);

//end
```

Scilab code Exa 6.9 time for drying the fibreboard

```

clear;
clc;
printf("\t Example 6.9\n");

            //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

x1=.75;                 //moisture content on wet basis
xbar=0.1;                 //equilllibrium moisture on dry basis
xcr=0.6;                 //critical moisture content
Ls=0.90;                 //mass of bone dry solid ais the drying surface
A=0.3*0.3*2;             //both upper surafce and lower surface are exposed
//A*Nc=10^-4;
x2=.2;                  //moisture content on wet basis finally after drying
Xcr=0.6/0.4;            //crtical moisture content
X1=3;                   //moisture content on dry basis intially
X2=0.25;                //moisture content on dry basis finally after drying
Xbar=0.1/0.9;          //equillibrium moisture
tbar=Ls/(10^-4) * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-Xbar)));
printf("\n the time for drying the sheets from 75 to 25 percent moisture under same drying conditions
```

```
    is  :%f hr",tbar/3600);

//end
```

Scilab code Exa 6.10 time for drying the moist material

```

clear;
clc;
printf("\t Example 6.10\n");
            //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

//Ls/ A*Nc is unknown;

Xcr=0.16;              //crtical moisture content
X1=.33;                //moisture content on dry
   basis intially
X2=0.09;               //moisture content on dry
   basis finally after drying
Xbar=.05;             //equillibrium moisture
tbar=7;               //time needed to dry from 33 to
    9 percent on bone dry basis

    //let Ls / A*Nc be = p
p=poly([0],'p');          //calc. of Ls / A*Nc be = p
   value
x=roots(tbar-p * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar))));

//new X1 AND X2 are now given as follows
X1=0.37;               //new moisture content on dry
   basis intially
```

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

//end
```

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

```

clear;
clc;
printf("\t Example 6.11\n");

d=.22;                        //density of dry pulp in g/cc;
x1=.65;                     //moisture content on wet basis
x2=.3;                     //moisture content on wet basis
Ls=2.5;                     //mass of bone dry solid ais the drying surface in kg
A=1.5*1.5*2;                 //both upper surafce and lower surface are exposed
v=1.5*1.5*.5;                //volume of material
Nc=1.4;                      //in kg/m^2*hr
Xcr=1.67;                   //crtical moisture content
X1=x1/(1-x1);                      //moisture content on dry basis  intially
X2=x2/(1-x2);                  //moisture content on dry basis  finally after drying
Xbar=0.0;                  //equillibrium moisture
```

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

//end
```

Scilab code Exa 6.12 time for drying the slab

```

clear;
clc;
printf("\t Example 6.12\n");

d=.22;                    //density of dry pulp in g
   /cc;

Ls=1.125*10^-2*.22*10^3;                   //mass of
   bone dry solid ais the drying surface
A=1.5*1.5*2;              //both upper surafce and
   lower surface are exposed
v=1.5*1.5*.5;             //volume of material
Nc=1.4;                   //in kg/m^2*hr
x2=.2;                    //moisture content on wet
   basis  finally after drying
Xcr=0.46;                 //crtical moisture content
X1=0.15;                   //moisture content on dry
   basis  intially
X2=0.085;                 //moisture content on dry
   basis  finally after drying
```

```
Xbar=0.025;          //equillibrium moisture

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

//end
```

Scilab code Exa 6.13 time for drying the filter cake

```

clear;
clc;
printf("\t Example 6.13\n");

//Ls/ A*Nc is unknown;

Xcr=0.14;              //crtical moisture content
x1=0.3;                //moisture content on wet basis
x2=0.1;                //moisture content on wet basis
X1=x1/(1-x1);                //moisture content on dry basis intially
X2=x2/(1-x2);               //moisture content on dry basis finally after drying
Xbar=0.04;             //equillibrium moisture
tbar=5;               //time needed to dry from 30 to
```

```
    10 percent on bone dry basis

    //let Ls / A*Nc be = p
p=poly([0],'p');          //calc. of Ls / A*Nc be = p
   value
x=roots(tbar-p * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)
   /(X2-Xbar))));

//new X1 AND X2 are now given as follows
x1=.3;                   //new moisture content on wet
   basis
x2=0.06;                   //new moisture content on
   wet basis
X1=x1/(1-x1);                  //new moisture content on
    dry basis intially
X2=x2/(1-x2);                  //new moisture content on
    dry basis finally after drying
tbar=x * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-
   Xbar)));
printf("\n the time for drying the sheets from 30 to
    10  percent moisture under same drying
   conditions is :%f hr",tbar);

//end
```

Scilab code Exa 6.14 time for drying the sheets

```

clear;
clc;
printf("\t Example 6.14\n");

d=450;                    //density of dry pulp in
   kg/m^3;
thickness=0.05;           //thickness in m^2
```

```
Ls=d*thickness;              //mass of bone dry solid ais the drying surface
A=1;                         //area in m^2
v=1*5*10^-2;                 //volume of material
Nc=4.8;                      //in kg/m^2*hr
xcr=.2;
xbar=0.02;
x1=.45;                      //new moisture content on wet basis
x2=0.05;                      //new moisture content on wet basis
X1=x1/(1-x1);                    //new moisture content on dry basis intially
X2=x2/(1-x2);                    //new moisture content on dry basis finally after drying
Xbar=xbar/(1-xbar);                        //crtical moisture content
Xcr=xcr/(1-xcr);          //equillibrium moisture

//tbar=(Ls/(A*Nc))*((Xcr-Xbar)*log((Xcr-Xbar)/(X2-Xbar)));
        // but initial moisture is more than Xcr, so there is constant rate drying period and only falling rate peroid is observed
tbar=Ls/(A*Nc) * ((X1-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(X2-Xbar)));
printf("\n the time for drying the sheets from 45 to 5 percent moisture under same drying conditions is :%f min",tbar);

//end
```

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

```
clear;
clc;
printf("\t Example 6.15\n");

t1=20;                    //ambient air temperature
t2=70;                    //exhaust air temperature
r1=150;                   //evaporation of water
r2=.25;                   //outlet solid moisture content
t3=15;                  //inlet solid temperature
t4=65;                    //outlet solid temperature
p=5;                     //power demand in KW
h=18;                    //heat loss in kj

h1=1;                         //mean specific heat of dry air in kj/kg*K
h2=1.25;                        //mean specific heat of dry material in kj/kg*K
h3=4.18;                        //mean specific heat of moisture in kj/kg*K
e=2626;                        //enthalpy of saturated water vapour in kj/kg

             //basis is 1hr
a1=r1*h3*(t4-t3);              //heat required for heating 150 kg of water from 15 to 65
a2=r1*e;                       //heat required for 150 kg water evaporation
a3=2000*h1*(t2-t1);            //heat required for heating air from 20 to 70
a4=r2*h3*(t4-t3);              //heat required for heating moisture from 15 to 65
a5=120*h2*(t4-t3);             //heat required for heating dry solid from 15 to 65
hlost=h*3600;                   //heat lost in kj
total=(a2+a3+a4+a5+hlost)/3600;     //total heat lost
printf("\n :%f  kW of heat required for 2000kg/hr of dry air",total);
```

```
ans1=a2+a1;              //heat needed for evaporation
printf("\n heat needed fro evaporation is :%f",ans1
   /3600);
ans2=(ans1/3600)/total;     //fraction of this heat
   needed for evaporation
printf("\n fraction of this heat needed for
   evaporation:%f",ans2);


//end
```

Scilab code Exa 6.16 surface area of the roller

```

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

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

//end
```

Scilab code Exa 6.17 time for drying the sheets

```

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

```
v2=.6*1.2*.012;                    //volume of the new solid
w2=1.8;                            //weight of the old solid
w3=864*10^-5*1.8*10^-5/54;    //weight of the bone dry solid

    //Nc is prporional to =(t-ts) = (G)^0.71---- whrere G is the mass flow rate
v1=3;                            //old velocity
Tg=52;                           //old dry bulb temperature
Tw=21;                           //wet bulb temperature
H=.002;                          //humidity
SH=0.015;                        //saturated humidity
vnew=5                          //new velocity
Tgnew=66;                       //new DBT
Twnew=24;                       //new WBT
Hnew=.004;                    //new humidity
SH=.020;                     //new satuurated humidity

                //hence drying rate of air under new condition
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/m^2*s
DT=Ls/(A*Nc) * ((Xo-Xcr)+(Xcr-Xbar)*log((Xcr-Xbar)/(Xfinal-Xbar)));
printf("\n the time for drying the sheets from 25 to 2 percent moisture under same drying conditions is :%f hours",DT/3600);


//end
```

Chapter 7

Crystallisation

Scilab code Exa 7.1 total weight of the solution

```

clear;
clc;
printf("\t Example 7.1\n");

//let x be the weight of water in the quantity of solution needed

c=.498;                    //solute content afetr crystallisation
W1=111;                    //molecular weight of CaCl2
W2=219;                    //molecular weight of CaCl2.6 H2O
M1=(108/W2)*100;           //water present in 100kg of CaCl2.6H2O
M2=(W1/W2)*100;           //CaCl2 present in 100kg of CaCl2.6H20
//t=M2+c*x;               //total weight entering the solubility
//x+49.3;                 total water solubility used
//s*(x+49.3)/100          //total Cacl2 after solubility
```

```
x=poly([0],'x');          //calc. x the weight of
    crystal
t=roots((M2+c*x)-(x+49.3)*.819);
printf("\nthe weight of water in the quantity of
    solution needed :%f kg",t);

h=(c)*t;              //weight of CaCl2 corresponding
    to weight water
tw=t+h;                // total weight of the solution
printf("\nthe  total weight of the solution is :%f
    kg",tw);
//end
```

Scilab code Exa 7.2 percentage saturation and yield

```

clear;
clc;
printf("\t Exercise 7.1\n");

//part(i)
w1=1000;                  //weight of solution to be
    cooled
s1=104.1;                 //solubility at 50 degree per
     100 kg of water
s2=78.0;                 //solubility at 10 degree per
    100 kg of water
a2=45;                 //percentage of sodium nitrate
    in the solution per 100kg of solution

x1=s1/(100+s1)*100;                  //percentage of
    saturated solution at 50 degree
tw=(a2/(100-a2))/(x1/(100-x1));      //the
    percentage saturation
printf("\nthe  percentage saturation is :%f percent"
```

```
,tw*100);

//part(ii)
//let x be the weight of NaNO3 crystal formed after crystallisation
x=poly([0],'x');          //calc. x the weight of crystal
t=roots((w1*a2/100)-(x+(w1-x)*s2/(100+s2)));
printf("\n the weight of NaNO3 crystal formed after crystallisation :%f kg",t);

//part(iii)
yield=t/(a2*w1/100);             //yield = weight of NaNO3 crystal formed/weight of NaNO3
printf("\n the percentage yield is:%f percent",yield*100);
//end
```

Scilab code Exa 7.3 temperature to which solution should be cooled

```

clear;
clc;
printf("\t Example 7.3\n");

s1=19.75;                  //solubility at 70 degree per 100 gm of water
s2=16.5;                   //solubility at 50 degree per 100 gm of water
s3=12.97;                  //solubility at 30 degree per 100 gm of water
s4=9.22;                   //solubility at 10 degree per 100 gm of water
s5=7.34;                   //solubility at 0 degree per 100 gm of water
```

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

//end
```

Scilab code Exa 7.4 percentage saturation and yield

```

clear;
clc;
printf("\t Example 7.4\n");
//part(i)
a1=146;                //solubility at 70 degree
a2=121;                //solubility at 10 degree
t1=58;                 // percentage of solute
   content
t2=40.66;
x1=a1/(100+a1) *100;         //percentage of
   saturated solution at 50 degree
tw=(t1/42)/(x1/t2);    //the  percentage saturation
printf("\nthe  percentage saturation is :%f percent"
   ,tw*100);

//part(ii)
p1=2000*.58;                //weight of solute in 200
```

```
    kg of solution 2000*.58
//let x be the weight of  crystal formed after
    crystallisation
x=poly([0],'x');          //calc. x the weight of
    crystal
t=roots((1160)-(x+(1055.02-.547*x)));
printf("\n the weight of NaNO3 crystal formed after
    crystallisation :%f kg",t);

//part(iii)
yield=t/p1;               //yield = weight of NaNO3
    crystal formed/weight of NaNO3
printf("\n the percentage yield is:%f percent",yield
    *100);
//end
```

Scilab code Exa 7.5 the weight of Na2SO4 hydrate crystal

```

clear;
clc;
printf("\t Example 7.5\n");

p1=.3;                      //percentage of the solute
    in the solution
w1=1000;                    //weight of the solution
    taken
w2=142;                    //molecular weight of Na2SO4
    .
M1=(w2/(180+w2));           //solute (Na2SO4) present
    in the Na2CO3.10H2O solution
s1=40.8;                 //solubility of Na2SO4 at 30
    degree per 100 gm of water
s2=9.0;                 //solubility of Na2SO4 at 10
    degree per 100 gm of water
```

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

//end
```

Scilab code Exa 7.6 percentage yield of Na2CO3 hydrated crysta

```

clear;
clc;
printf("\t Example 7.6\n");

s1=12.5;                    //solubility of Na2CO3 at 10 degree per 100 gm of water
p1=.3;                      //percentage of the solute in the solution
w1=2000;                    //weight of the solution taken
w2=106;                     //molecular weight of Na2CO3.
M1=(w2/(180+w2));           //solute (Na2CO3) present in the Na2CO3.10H2O solution
//let x be the quantity of Na2CO3.10H2O crystal formed
x=poly([0],'x');            //calc. x the weight of crystal
t=roots(w1*p1-M1*x-(w1-x)*(s1/(100+s1)));
printf("\n the weight of quantity of Na2CO3.10H2O  :%f kg",t);
```

```
//in the book the ans is wrong, they have calculated 2000*0.3-2000*12.5/112.5 as =x(miscalculation)

p=(286/106)*w1*p1;            //weight of Na2CO3.10H2O crystal present in the original solution
yield=t/p;                    //percentage yield
printf("\n percentage yield :%f percent",yield*100);
//end
```

Scilab code Exa 7.7 percentage yield of K2CO3 hydrated crystal

```

clear;
clc;
printf("\t Example 7.7\n");

s1=139.8;                   //solubility at 80 degree per 100 gm of water
s2=110.5;                   //solubility at 20 degree per 100 gm of water
w2=174.2;                      //molecular weight of K2CO3.10H2O
M1=(138/w2)*100;            //water present in 100kg of K2CO3.10H2O
//let x be the quantity of Na2CO3.10H2O
x=poly([0],'x');             //calc. x the weight of crystal
t=roots(500*(139.8/239.8)-.7921*x-(500-x)*110.5/210.5);
printf("\n the weight of quantity of K2CO3.10H2O formed  :%f kg",t);

p=(174/138)*500*(139.8/239.8);        //weight of crystal present in the original solution
yield=t/p;                    //percentage yield
```

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

Scilab code Exa 7.8 percentage yield FeSO4 hydrate crystal

```

clear;
clc;
printf("\t Example 7.8\n");

s1=20.51;                  //solubility at 10 degree per 100 gm of water
w2=277.85;                      //molecular weight of FeSO4.7H2O

//let x be the quantity of Na2CO3.10H2O
x=poly([0],'x');            //calc. x the weight of crystal
t=roots(900*.4-.5465*x-(900-x)*20.5/120.5);
printf("\n the weight of quantity of FeSO4.7H2O formed  :%f kg",t);

p=(277.85/151.85)*900*(0.4);         //weight of crystal present in the original solution
yield=t/p;                      //percentage yield
printf("\n percentage yield :%f percent",yield*100);
//end
```

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

```

clear;
```

```
clc;
printf("\t Example 7.2\n");

//part(i)
a1=229.7;                    //solubility at 60 degree
a2=174.7;                    //solubility at 60 degree
t1=68;                      // percentage of sodium nitrate
t2=30.34;
x1=a1/329.7 *100;          //percentage of saturated solution at 50 degree
tw=(t1/32)/(x1/t2);     //the  percentage saturation
printf("\nthe  percentage saturation is :%f percent",tw*100);

//part(ii)
//let x be the weight of Cesium chloride crystal formed after crystallisation
x=poly([0],'x');          //calc. x the weight of crystal
t=roots(1000*.68-(x+(1000-x)*174.7/274.7));
printf("\n the weight of CaCl2 crystal formed after crystallisation :%f kg",t);

//part(iii)
yield=t/680;              //yield = weight of CaCl2 crystal formed/weight of CaCl2
printf("\n the percentage yield of Cesium chloride is:%f percent",yield*100);
//end
```

Scilab code Exa 7.10 weight of Na2CO3hydrate needed to dissolve Na2CO3

```

clear;
```

```
clc;
printf("\t Example 7.10\n");

s1=38.8;                  //solubility at 30 degree per 100 gm of water
s2=12.5;                  //solubility at 10 degree per 100 gm of water
w2=296;                    //molecular weight of Na2CO3.10H2O
per=116/w2 *100;           //percentage solute in Na2CO3.10H2O

//let x be the quantity of Na2CO3.10H2O
w=200;                       //original solotion weight
m1=w*(s2/(s2+100));          //weight of Na2CO3.10H2O needed to dissolve Na2CO3 present in the original solotion
w3=w-m1;                     //weight of water
//w4=m1+per/100;                weight of Na2CO3 after dissolution
x1=s1/(s1+100);              //weight fraction of solute after dissolution
printf("\n the weight of quantity of Na2CO3.10H2O formed  :%f kg",w3);

//for the total solution after dissolution
x=poly([0],'x');           //calc. x the weight of crystal
t=roots((m1+per*x/100)-((m1+per*x/100)+(w3+.609*x))*x1);
printf("\nweight of Na2CO3.10H2O needed to dissolve Na2CO3 present in the original solution  %f kg",t);

//end
```

Scilab code Exa 7.11 weight of Na2CO3 hydrated formed

```

clear;
clc;
printf("\t Example 7.11\n");

s1=35;                    //percentage of solution
x1=6000;                  //weight of Na2CO3 solution
s2=21.5;                  //solubility at 20 degree per
   100 gm of water
w2=296;                     //molecular weight of Na2CO3
   .10H2O
per=116/w2 *100;            //percentage solute in
   Na2CO3.10H2O
w1=s1*x1;                //weight of solute
w3=x1*0.04;              //weight of solution lost by
   vaporisation
//let x be the quantity of Na2CO3.10H2O formed
//making material balance
x=poly([0],'x');            //calc. x the weight of
   crystal
t=roots(2100-(.391*x)-(6000-240-x)*(21.5/121.5));
printf("\n the weight of Na2CO3.10H2O crystal formed
    after crystallisation :%f kg",t);



//end
```

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

```
clear;
clc;
printf("\t Example 7.12\n");

                    //FeSO4.7H2O
C=1000;                              //crystal formed in kg
hf=26.002;                           //enthalpy of the feed at 80 degree in cal/g
hl=-1.33;                            //enthalpy of the saturated sol at 30 degree in cal/g
hc=-50.56;                           //enthalpy of crystal
xf=40/(100+40);
xm=30/(100+30);
xc=151.84/277.85;            //151.84 is the weight of FeSO4
        //component balance
//     F*xf = M*xm + C*xc      -------eqn 1st
//     F = M + 10000 + V       -------eqn 2nd
//     F*Hf = V*Hv + M*Hm +C*Hc------eqn 3rd
Hf=26.002;                           //enthalpy of the feed at 80 degree in cal/g
Hv=612;                      //
Hm=-1.33;                    //enthalpy of the saturated sol at 30 degree in cal/g
Hc=-50.56;                   //enthalpy of crystal leaving the crystalliser

//solving these we gt
a=[1 -1 -1;.286 -.231 0;26.002 1.33 -612]
b=[10000;5470;-505600]
x=inv(a)*b;                          //solving out the values using matrices
t1=x(1);                             //3 solution of the eqn
t2=x(2);
t3=x(3);
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);
//end
```

Scilab code Exa 7.13 cooling water requirement

```

clear;
clc;
printf("\t Example 7.13\n");

C=800;                              //crystal formed in kg/hr
t2=49;                               //temp. of the entering fed
t1=27;                              //temp. of the product
t3=21;                              //temp. of the leaving cooling water
t4=15;                             //temp. of the enetring cooling water
U=175;                             //overall heat transfer coefficient
F=140*151.85/277.85;           //feed concentration
xf=F/240;                      //concentration in feed solution
P=74*151.85/277.85;            //product concentration
xm=P/174;                      //concentration of FeSO4 in product solution
xc=151.85/277.85;            //
        //mass balance    F = M+C     ----eqn 1st
        //sloute balance F*xf = M*xm + C*xc ----eqn 2nd
//solving these we get
F=800*.3141/0.0866;            //feed conc.
M=F-C;                         //product concentration
        //making energy balance
```

```
        //heat to be removed by cooling water =heat to be removed from solution + heat of crystallization
cp=.7;                      //specific heat capacity
dt=(t2-t1);                 //change in temp.
dh=15.8;                    //heat of crystallization
Q=F*cp*dt+dh*C;             //heat to be removed by cooling water
cp=1;                       //specific heat capacity of water
dt=(t3-t4);                      //change in temp.
mw=Q/(cp*dt);               //cooling water needed
printf("\n cooling water requiement is :%f kg/hr",mw);
    //Q=U*A*(dtlm)
dtlm=((t2-t3)-(t1-t4))/(log((t2-t3)/(t1-t4)));//log mean temp. difference
A=Q/(U*dtlm);                       //area of the crystallizer section
l=A/1.3;
printf("\n length of crystallliser sections needed is :%f m",l);

//end
```