Scilab Textbook Companion for Fluidization Engineering by K. Daizo And O. Levenspiel[1]

Created by
Subash G
B.Tech
Chemical Engineering
SASTRA University
College Teacher
Dr. P.R.Naren
Cross-Checked by

May 20, 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: Fluidization Engineering

Author: K. Daizo And O. Levenspiel

Publisher: Butterworth-Heinemann, Massachusetts

Edition: 2

Year: 1991

ISBN: 81-312-0035-3

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes 4

3 Fluidization and Mapping of Regimes 6

4 The Dense Bed 14

5 Bubbles in Dense Beds 22

6 Bubbling Fluidized Beds 26

7 Entrainment and Elutriation from Fluidized Beds 38

8 High velocity Fluidization 48

9 Solid Movement Mixing Segregation and Staging 54

10 Gas Dispersion and Gas Interchange in Bubbling Beds 61

11 Particle to Gas Mass and Heat Transfer 69

12 Conversion of Gas in Catalytic Reactions 80

13 Heat Transfer between Fluidized Beds and Surfaces 92

14 The RTD and Size Distribution of Solids in Fluidized Beds 100

15 Circulation Systems 110

16 Design for Physical Operations 120

17 Design of Catalytic Reactors **132**

18 The Design of Noncatalytic Gas Solid Reactors **145**

List of Scilab Codes

Exa 3.1	Size Measure of Nonuniform Solids	6
Exa 3.2	Estimation of Minimum Fluidizing Velocity	7
Exa 3.3	Estimation of Terminal Velocity of Falling Particles	9
Exa 3.4	Prediction of Flow Regimes	10
Exa 4.1	Design of a Perforated Plate Distributor	14
Exa 4.2	Design of a Tuyere Distributor	17
Exa 4.3	Power Requirement for a Fluidized Coal Combustor	19
Exa 5.1	Characteristics of a Singe Bubble	22
Exa 5.2	Initial Bubble Size at a Distributor	23
Exa 6.1	Bubble Size and Rise Velocity in Geldart A Beds	26
Exa 6.2	Bubble Size and Rise Velocity in Geldart B Beds	27
Exa 6.3	Scale down of a Commercial Chlorinator	30
Exa 6.4	Reactor Scale up for Geldart A Catalyst	32
Exa 6.5	Reactor Scale up for Geldart B Catalyst	36
Exa 7.1	Entrainment from fine particle beds with high freeboard	38
Exa 7.2	Entrainment from large particle beds with high freeboard	39
Exa 7.3	Entrainment from beds with a wide size distribution of solids	40
Exa 7.4	kstar from steady state experiments	42
Exa 7.5	Comparing predictions for kstar	43
Exa 7.6	Entrainment from a short vessel	46
Exa 8.1	Performance of a Fast Fluidized Vessel	48
Exa 9.1	Vertical Movement of Solids	54
Exa 9.2	Horizontal Drift Of Solids	56
Exa 9.3	Design of Baffle Plates	58
Exa 10.1	Estimate Interchange Coefficients in Bubbling Beds	61
Exa 10.2	Compare the Relative Importance of Kbc and Kce	63

Exa 10.3 Compare Interchange Rates for Adsorbed and Nonadsorbed Gases 66
Exa 11.1 Fitting Reported Mass Transfer Data with the Bubbling Bed Model 69
Exa 11.2 The Effect of m on Bubble Emulsion Interchange . . . 71
Exa 11.3 Fitting Reported Heat Transfer Data with the Bubbling Bed Model 74
Exa 11.4 Heating a Particle in a Fluidized Bed 76
Exa 12.1 Fine Particle Geldart A Bubbling Bed Reactor 80
Exa 12.2 Commercial Sized Phthalic Anhydride Reactor 82
Exa 12.3 Bubbling Bed Reactor for Intermediate Sized Reactor 85
Exa 12.4 Reaction in the Slow Bubble Regime 87
Exa 12.5 Conversion in the Freeboard of a Reactor 89
Exa 13.1 h on a Horizontal Tube Bank 92
Exa 13.2 Effect of Gas Properties on h 94
Exa 13.3 Effect of Particle Size on h 96
Exa 13.4 Freeboard Heat Exchange 97
Exa 14.1 Flow with Elutriation 100
Exa 14.2 Flow with Elutriation and Change in Density of Solids 102
Exa 14.3 Single Size Feed of Shrinking Particles 105
Exa 14.4 Wide Size Distribution of Shrinking Particle 106
Exa 14.5 Elutriation and Attrition of Catalyst 107
Exa 15.1 Circulation Rate when Deactivation Controls 110
Exa 15.2 Circulation Rate when Heat Duty Controls 111
Exa 15.3 Aeration of Fine Particle Downcomer 113
Exa 15.4 Circulation in Side by Side Beds 115
Exa 15.5 Steam Seal of a Coarse Particle Downcomer 117
Exa 16.1 Single Stage Limestone Calciner 120
Exa 16.2 Multistage Limestone Calciner 122
Exa 16.3 Multistage Adsorber 125
Exa 16.4 Dryer Kinetics and Scale up 126
Exa 16.5 Solvent Recovery from Polymer Particles 128
Exa 17.1 Reactor Development Program 132
Exa 17.2 Design of a Commercial Acrylonitrile Reactor 137
Exa 17.3 Reactor Regenerator with Circulating Catalyst Catalytic Cracking 139
Exa 18.1 Kinetics of Zinc Blende Roasting 145
Exa 18.2 Kinetics of Carbon Burning 146

Exa 18.3 Roasting Kinetics from Flowing Solids Data 148
Exa 18.4 Scale up of a Reactor with Flowing Solids 150
Exa 18.5 Design of a Roaster for Finely Ground Ore 153
Exa 18.6 Design of a Roaster for Coarse Ore 158

List of Figures

10.1 Estimate Interchange Coefficients in Bubbling Beds 64

11.1 Fitting Reported Mass Transfer Data with the Bubbling Bed Model 72

11.2 Fitting Reported Heat Transfer Data with the Bubbling Bed Model 77

14.1 Flow with Elutriation and Change in Density of Solids . . . 105

Chapter 3

Fluidization and Mapping of Regimes

Scilab code Exa 3.1 Size Measure of Nonuniform Solids

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-3, Example 1, Page 68
//Title: Size Measure of Nonuniform Solids
//
    ================================================================

clear
clc

//INPUT
weight = [0;60;150;270;330;360];// Weight in grams
    for the oversized particles
psize = [50;75;100;125;150;175];//PSD in micrometers

//CALCULATION
len = length(psize); // To obtain the size of input
```

```
array
// Computation of sauter mean diameter for the given PSD
i = 1;
while i<len
        dpi(i)=(psize(i,:)+ psize(i+1,:))/2;
        weightf(i)=(weight(i+1)-weight(i))/weight(6);
        dp(i)=weightf(i)/dpi(i);
        i=i+1;
end
dpbar=1/sum(dp);//Calculation of average particle daimeter Eq.(15)

//OUTPUT
mprintf('\n The Sauter mean diameter of the material with the given particle size distribution = %f micrometer',dpbar);

//=================================================END OF PROGRAM
=====================================================================
```

Scilab code Exa 3.2 Estimation of Minimum Fluidizing Velocity

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-3, Example 2, Page 76
//Title: Estimation of Minimum fluidizing velocity
//
=====================================================================
```

```
clear
clc

//INPUT
ephsilon=0.55;//Void fraction of bed
rhog=0.0012;//Density of gas in g/cc
myu=.00018;//Viscosity of gas in g/cm s
dpbar=0.016;//Mean diameter of solids in centimeter
phis=0.67;//Sphericity of solids
rhos=2.6;//Density of solids in g/cc
g=980;//Acceleration due to gravity in square cm/s^2

//CALCULATION
//Computation of umf using the simplified equation for small particles
umf=((dpbar^2)*(rhos-rhog)*g*(ephsilon^3)*(phis^2))/(150*myu*(1-ephsilon));//Simplified equation to calculate minimum fluidizing velocity for small particles Eq.(21)
Re=(dpbar*umf*rhog)/myu;//To calculate Reynolds number for particle

//Computation of umf if neither void fraction of bed nor sphericity is known
c1=28.7; c2=0.0494;//Value of constants from Table 4, page 70
umf1=(myu/(dpbar*rhog))*(((c1^2)+((c2*(dpbar^3)*rhog*(rhos-rhog)*g)/(myu^2)))^0.5-c1);//Equation to calculate minimum fluidizing velocity for coarse particles Eq.(25)
err=((umf-umf1)/umf)*100;//Calculation of error from experimental value

//OUTPUT
if Re<20 then
    mprintf('\nThe particle Reynolds no = %f',Re)
    printf('\nThe simplified equation used for calculating minimum fluidizing velocity is
```

```
valid.');
end
mprintf('\nThe minimum fluidizing velocity by simplified equation for small particles = %fcm/s',umf);
mprintf('\nThe minimum fluidizing velocity by equation for coarse partilces = %fcm/s',umf1);
mprintf('\nThis value is %f percent below the experimentally reported value.',err);

//==========================================END OF PROGRAM ==========================================
```

Scilab code Exa 3.3 Estimation of Terminal Velocity of Falling Particles

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-3, Example 3, Page 82
//Title: Estimation of terminal velocity of falling particles
//==========================================================================================

clear
clc

//INPUT
rhog=1.2e-3;//Density of air in g/cc
myu=1.8e-4//Viscosity of air in g/cm s
dpbar=0.016//Mean diameter of solids in centimeter
phis=0.67;//Sphericity of solids
```

```
rhos=2.6;//Density of solids in g/cc
g=980//Acceleration due to gravity in square cm/s^2

//CALCULATION
dpstar=dpbar*((rhog*(rhos-rhog)*g)/myu^2)^(1/3);//
    Calculation of dimensionless particle size Eq
    .(31)
utstar=((18/(dpstar^2))+(2.335-(1.744*phis))/(dpstar
    ^0.5))^-1;//Calculation of dimensionless gas
    velocity Eq.(33)
ut=utstar*((myu*(rhos-rhog)*g)/rhog^2)^(1/3);//
    Calculation of terminal velocity of falling
    particles Eq.(32)


//OUTPUT
mprintf('\nThe dimensionless particle size = %f',
    dpstar);
mprintf('\nThe dimensionless gas velocity = %f',
    utstar);
mprintf('\nThe terminal velocity of falling
    particles = %fcm/s', ut);

//==========================================END OF PROGRAM
    ==============================================
```

Scilab code Exa 3.4 Prediction of Flow Regimes

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-3, Example 4, Page 91
```

```
//Title: Prediction of flow regime
//

clear
clc

//INPUT
rhos=1.5;//Density of Solid in g/cc
uo1=40; uo2=80;//Superficial gas velocity in cm/s
dp1=0.006; dp2=0.045;//Particle size in centimeter
rhog1=1.5E-3; rhog2=1E-3; //Density of gas in g/cc
myu1=2E-4; myu2=2.5E-4;//Viscosity of air in g/cm s
g=980;//Acceleration due to gravity in square cm/s^2

//CALCULATION
//for smaller particles
dpstar1=dp1*((rhog1*(rhos-rhog1)*g)/myu1^2)^(1/3);//Calculation of dimensionless particle diamter Eq.(31)
uostar1=uo1*((rhog1^2)/((myu1)*(rhos-rhog1)*g))^(1/3);
uostar2=uo2*((rhog1^2)/((myu1)*(rhos-rhog1)*g))^(1/3);//Calculation of dimensionless superficial gas velocity Eq.(32)

//for larger particles
dpstar2=dp2*((rhog2*(rhos-rhog2)*g)/myu2^2)^(1/3);//Calculation of dimensionless particle diamter Eq.(31)
uostar3=uo1*((rhog2^2)/((myu2)*(rhos-rhog2)*g))^(1/3);
uostar4=uo2*((rhog2^2)/((myu2)*(rhos-rhog2)*g))^(1/3);//Calculation of dimensionless superficial gas velocity Eq.(32)


//OUTPUT
```

```
printf('\nFor particle of size %f centimeter',dp1);
mprintf('\nThe dimensionless particle diameter = %f',dpstar1);
mprintf('\nThe dimensionless superficial gas velocity = %fcm/s(for superficial gas velocity of %fcm/s)',uostar1,uo1);
mprintf('\nThe dimensionless superficial gas velocity = %fcm/s(for superficial gas velocity of %fcm/s)',uostar2,uo2);
mprintf('\n\nFrom Fig.16(page 89)comparing u*=%f vs dp*=%f',uostar1,dpstar1);
mprintf('\nFor Superficial gas velocity =%f \nMode of Fluidization:Onset of turbulent fluidization in an ordinary bubbling bed',uo1);
mprintf('\nFrom Fig.16(page 89)comparing u* =%f vs dp* =%f',uostar2,dpstar1);
mprintf('\nFor Superficial gas velocity =%f \nMode of Fluidization:Fast fluidization(requires a circulating solid system)',uo2);
printf('\n\nFor particle of size %f centimeter',dp2)
mprintf('\nThe dimensionless particle diameter = %f',dpstar2);
mprintf('\nThe dimensionless superficial gas velocity = %fcm/s(for superficial gas velocity of %fcm/s)',uostar3,uo1);
mprintf('\nThe dimensionless superficial gas velocity = %fcm/s(for superficial gas velocity of %fcm/s)',uostar4,uo2);
mprintf('\n\nFrom Fig.16(page 89)comparing u*=%f vs dp*=%f',uostar3,dpstar2);
mprintf('\nFor Superficial gas velocity =%f \nMode of Fluidization:Bublling Fluidization',uo1);
mprintf('\nFrom Fig.16(page 89)comparing u* =%f vs dp* =%f',uostar4,dpstar2);
mprintf('\nFor Superficial gas velocity =%f \nMode of Fluidization:Bubbling Fluidization',uo2);

//=========================================END OF PROGRAM
```

Chapter 4

The Dense Bed

Scilab code Exa 4.1 Design of a Perforated Plate Distributor

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-4, Example 1, Page 106
//Title: Design of a Perforated Plate Distributor
//
    ==========================================================

clear
clc

//INPUT
dt=4;//Vessel diameter in m
Lmf=2;//Length of the bed in m
ephsilonmf=0.48;//Void fraction of bed
rhos=1500;//Density of solid in kg/m^3
rhog=3.6;//Density of gas in kg/m^3
myu=2E-5;//Viscosity of gas in kg/m s
po=3;//Pressure of inlet gas in bar
uo=0.4;//Superficial velocity of gas in m/s
```

```
uorm=40;//Maximum allowable jet velocity from holes in m/s
g=9.80;//Acceleration due to gravity in m/s^2
gc=1;
pi=3.1428;

//CALCULATION
//Computation of minimum allowable pressure drop through the distributor
deltapb={(1-ephsilonmf)*(rhos-rhog)*g*Lmf}/gc;//Calculation of pressure drop in bed using Eqn.(3.17)
deltapd=0.3*deltapb;//Calculation of pressure drop in distributor using Eqn.(3)

//Computation of orifice coefficient
Ret=(dt*uo*rhog)/myu;
if    Ret>=3000 then Cd=0.60;
elseif    Ret>=2000 then Cd=0.61;
elseif    Ret>=1000 then Cd=0.64;
elseif    Ret>=500  then Cd=0.68;
elseif    Ret>=300  then Cd=0.70;
elseif    Ret>=100  then Cd=0.68;
end

//Computation of gas velocity through orifice
uor=Cd*((2*deltapd)/rhog)^0.5;//Calculation of gas velocity through orifice by using Eqn.(12)
f=(uo/uor)*100;//Calculation of fraction of open area in the perforated plate


//Computation of number of orifices per unit area of distributor
dor=[0.001;0.002;0.004];//Different orifice diameters in m
n=length(dor);
i=1;
```

```
while i<=n
    Nor(i)=(uo*4)/(pi*uor*(dor(i))^2);//Calculation of number of orifices by using Eqn.(13)
    i=i+1;
end

//OUTPUT
mprintf('\nThe pressure drop in bed:%fPa',deltapb);
mprintf('\nThe minimum allowable pressure drop in distributor:%fPa',deltapd);
if uor<uorm then mprintf('\nThe gas veleocity of %fm/s is satisfactory',uor);
    else mprintf('\nThe gas veleocity of %fm/s is not satisfactory',uor);
end
if f<10 then mprintf('\nThe fraction of open area of %f percent is allowable',f);
    else mprintf('\nThe fraction of open area of %f percent is not allowable',f);
end
printf('\nDiameter of orifice(m)');
printf('\tNumber of orifices per unit area(per sq.m)');
j=1;
while j<=n
    mprintf('\n%f',dor(j));
    mprintf('\t\t%f',Nor(j));
    j=j+1;
end
printf('\nThis number can be rounded off.');
printf('\nSince orifices that are too small are liable to clog and those that are too large cause uneven distribution of gas, we choose orifice of diameter %fm',dor(2));

//==========================================================END OF PROGRAM
```

Scilab code Exa 4.2 Design of a Tuyere Distributor

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-4, Example 2, Page 108
//Title: Design of a Tuyere Distributor
//
   ==========================================================

clear
clc

//INPUT
lor=0.1;//Minimum allowable tuyere spacing in m
uorm=30;//Maximum allowable jet velocity from the
   tuyere in m/s
uo=0.4;//Superficial velocity of gas in m/s
uor=30.2;//Gas velocity through orifice,from Exa 1,
   in m/s
Cd=0.6;//Dicharge coefficient from Exa 1
rhog=3.6//Density of gas in kg/m^3
pi=3.1428;

//CALCULATION
Nor=1/(lor^2);//Calculation of number of orifices
   per unit area by assuming minimum spacing for
   tuyeres
dor={(4/pi)*(uo/uor)*(1/Nor)}^0.5;//Calculation of
   diameter of inlet orifiec by using Eqn.(13)

//Computation of diameter of hole for different
```

```
number of holes per tuyere
q=(lor^2)*uo;//Volumetric flow rate in m^3/s
Nh=[8;6;4];//Different number of holes per tuyere
n=length(Nh);
i=1;
while i<=n
    dh(i)=((((q/Nh(i))*(4/pi))/uorm)^0.5);//Calculation of diameter of holes
    i=i+1;
end
deltaph=(rhog/2)*((uor/Cd)^2);

//OUTPUT
printf('\nNumber of holes(number of holes/tuyeres)');
printf('\tDiameter of hole(m)');
j=1;
while j<=n
    mprintf('\n%f',Nh(j));
    mprintf('\t\t\t\t\t%f',dh(j));
    j=j+1;
end
printf('\nThe design chosen is as follows');
printf('\n\tTuyeres are as shown in Fig.2(b),page 97');
mprintf('\n\tNumber of holes = %f(Since rectangular pitch is chosen for tuyeres)',Nh(2));
mprintf('\n\tDiameter of hole = %fm',dh(2));
mprintf('\n\tDiameter of incoming high-pressure-drop orifice = %fm ID',dor);
printf('\nChecking the pressure drop in tuyeres');
mprintf('\nSince pressure drop of %fPa gives sufficiently high distributor pressure drop as seen in Exa.1, use of inlet orifice can be dispensed.',deltaph);

//==================================================END OF PROGRAM
```

Scilab code Exa 4.3 Power Requirement for a Fluidized Coal Combustor

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-4, Example 3, Page 110
//Title: Power Requirement for a Fluidized Coal
    Combustor(FBC)
//
    ==========================================================

clear
clc

//INPUT
deltapd=[3;10]//Distributor pressure drop in kPa
deltapd2=10;//Distributor pressure drop in kPa
po=101;//Entering air pressure in kPa
To=20;//Entering air temperature in degree C
y=1.4;//Fugacity of air
deltapb=10;//Pressure drop in bed in kPa
p3=103;//Pressure at the bed exit in kPa
F=8;//Feed rate of coal in tons/hr
H=25;//Gross heatig value of coal in MJ/kg
Fa=10;//Air required at standard condition in nm^3/
    kg
etac=0.75;//Efficiency of compressor
etap=36;//Efficiency of plant in %

//CALCULATION
//Calculation of volumetric flow rate of air
```

```
vo=((F*1000)*Fa*((To+273)/273))/3600;

//Case(a) Distributor Pressure drop = 3kPa and Case(
   b) Distributor Pressure drop = 10kPa
n=length(deltapd);
i=1;
while i<=n
    p2(i)=p3+deltapb;//Calculation of pressure at
       the entrance of the bed
    p1(i)=p2(i)+deltapd(i);//Calculation of pressure
        before entering the bed
    ws(i)=(y/(y-1))*po*vo*((p1(i)/po)^((y-1)/y)-1)
       *(1/etac);//Calculation of power required for
        the compressor by Eqn.(18) & Eqn.(20)
    i=i+1;
end

//Case(c) 50% of the required bypassed to burn the
   volatile gases. Distributor Pressure drop = 3kPa
//No change in pressure drop from case(a)
v1=vo/2;//New volumetric flow rate of air
ws1=ws(1)/2;//Power required for blower for primary
   air
ws2=(y/(y-1))*po*v1*((p3/po)^((y-1)/y)-1)*(1/etac);
   //Power required for blower for bypassed air
wst=ws1+ws2;//Total power required for the two
   blowers
p=((ws(1)-wst)/ws(1))*100;//Saving in power when
   compared to case(a)

//OUTPUT
printf('\nCase(a)');
mprintf('\n\tVolumetric flow rate of air = %f m^3/hr
   ',vo);
mprintf('\n\tPower required for compressor = %f kW',
   ws(1));
printf('\nCase(b)');
mprintf('\n\tVolumetric flow rate of air = %f m^3/hr
```

```
',vo);
mprintf('\n\tPower required for compressor = %f kW',ws(2));
printf('\nCase(c)');
mprintf('\n\tVolumetric flow rate of air = %f m^3/hr',v1);
mprintf('\n\tPower required for compressor for primary air = %f kW',ws1);
mprintf('\n\tPower required for blower for bypassed air = %f kW',ws2);
mprintf('\n\tTotal power required for the two blowers = %f kW',wst);
mprintf('\n\tPower saved compared to case(a) = %f percent',p);

//===========================================END OF PROGRAM===========================================================
```

Chapter 5

Bubbles in Dense Beds

Scilab code Exa 5.1 Characteristics of a Singe Bubble

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-5, Example 1, Page 126
//Title: Charactersitics of a Single Bubble
//
    ==========================================================================

clear
clc

//INPUT
dt=60;//ID of tube in cm
dp=300;//Size of particles of bed in micrometers
umf=3;//Velocity at minimum fluidization condition
    in cm/s
ephsilonmf=0.5;//Void fraction of bed at minimum
    fluidization condition
db=5;//Diameter of bubble in cm
g=980;//Acceleration due to gravity in cm/s^2
```

```
//CALCULATION
//Computation of rise velocity of bubble
if (db/dt)<0.125    then ubr=(0.711*((g*db)^0.5));//Rise velocity by Eqn.(3)
elseif (db/dt)<0.6  then ubr=(0.711*((g*db)^0.5))*1.2*exp(-1.49*(db/dt));//Rise velocity by Eqn.(4)
end

//Computation of cloud thickness
Rb=db/2;//Radius of bubble
uf=umf/ephsilonmf;//Velocity of emulsion gas
Rc=Rb*((ubr+(2*uf))/(ubr-uf))^(1/3);//Radius of cloud by Eqn.(6)

//OUTPUT
mprintf('\nThe rise velocity of the bubble=%fcm/s',ubr);
mprintf('\nThe cloud thickness=%fcm',Rc-Rb);
mprintf('\nFrom Fig.8(page 124)comparing fw vs dp, for dp = %f micrometer, wake fraction = 0.24',dp);

//==========================================END OF PROGRAM==========================================
```

Scilab code Exa 5.2 Initial Bubble Size at a Distributor

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

```

```
//Chapter-5, Example 2, Page 132
//Title: Initial Bubble Size at a Distributor
//==========================================================================
clear
clc

//INPUT
uo=15;//Superificial gas velocity in cm/s
umf=1;////Velocity at minimum fluidization condition in cm/s
lor=2;//Pitch of perforated plate in cm
g=980;//Acceleration due to gravity in cm/s^2
//CALCULATION
//Case(a) For porous plate
dbo1=(2.78/g)*(uo-umf)^2;//Initial bubble size using Eqn.(19)

//Case(b) For Perforated plate
Nor=(2/sqrt(3))*(1/lor)^2;//Number of orifices in cm^-2
dbo2=(1.30/(g^0.2))*((uo-umf)/Nor)^0.4;//Initial bubble size using Eqn.(15) assuming inital bubble size is smaller than hole spacing

//OUTPUT
printf('\nCase(a) For porous plate');
printf('\n\tInitial bubble size=%fcm',dbo1);
printf('\nCase(b) For Perforated plate');
printf('\n\tInitial bubble size=%fcm',dbo2);
printf('\n\tSince %f<%f, the equation used is correct.',dbo2,lor);

//==========================================END OF PROGRAM==========================================
```

Chapter 6

Bubbling Fluidized Beds

Scilab code Exa 6.1 Bubble Size and Rise Velocity in Geldart A Beds

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-6, Example 1, Page 150
//Title: Bubble Size and Rise Velocity in Geldart A
   Beds
//
   ==========================================================

clear
clc

//INPUT
z=0.5;//Height of bed in m
dt=0.5;//ID of tube in m
rhos=1.6;//Density of catalyst in g/cm^3
dpbar=60;//Averge catalyst diameter in micrometer
umf=0.002;//Velocity at minimum fluidization
   condition in m/s
uo=0.2;//Superficial velocity in m/s
```

```
dor=2;//Diameter of orifice in mm
lor=20;//Pitch of perforated plate in mm
g=9.80;//g=980;//Acceleration due to gravity in m/s^2

//CALCULATION
//Method 1. Procedure using Eqn.(10) & Eqn.(11)
db=(0.035+0.040)/2;//Bubble size at z=0.5m from Fig.7(a) & Fig.7(b)
ub1=1.55*((uo-umf)+14.1*(db+0.005))*(dt^0.32)+0.711*(g*db)^0.5;//Bubble velocity using Eqn.(10) & Eqn.(11)

//Method 2. Werther's procedure
si=0.8;//From Fig.6 for Geldart A solids
ub2=si*(uo-umf)+(3.2*(dt^(1/3)))*(0.711*(g*db)^0.5);//Bubble velocity using Eqn.(9)

//OUTPUT
printf('\nMethod 1. Procedure using Eqn.(10) & Eqn.(11)');
mprintf('\n\tDiameter of the bubble=%fm',db);
mprintf('\n\tRise velocity of the bubble=%fm/s',ub1);
printf('\nMethod 2. Werthers procedure');
mprintf('\n\tDiameter of the bubble=%fm',db);
mprintf('\n\tRise velocity of the bubble=%fm/s',ub2);

//========================================END OF PROGRAM============================================
```

Scilab code Exa 6.2 Bubble Size and Rise Velocity in Geldart B Beds

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-6, Example 2, Page 151
//Title: Bubble Size and Rise Velocity in Geldart B
   Beds
//
==========================================================================

clear
clc

//INPUT
z=0.5;//Height of bed in m
dt=0.5;//ID of tube in m
rhos=2.6;//Density of catalyst in g/cm^3
dpbar=100;//Averge catalyst diameter in micrometer
umf=0.01;//Velocity at minimum fluidization
   condition in m/s
uo=0.45;//Superficial velocity in m/s
dor=2;//Diameter of orifice in mm
lor=30;//Pitch of perforated plate in mm
g=9.80;//Acceleration due to gravity in m/s^2
pi=3.142857;

//CALCULATION
//Part(a).Bubble Size
Nor=(2/sqrt(3))*(1/lor^2);
dbo=5.5;

//Method 1.Werther's procedure for finding bubble
   size
z1=[0;5;10;20;30;50;70];
n=length(z1);
i=1;
while i<=n
    db(i)=0.853*((1+0.272*(uo-umf)*100)^(1/3))
```

```
*(1+0.0684*z1(i))^1.21;
    i=i+1;
end
db1=0.163;//Since bubble size starts at dbo=5.5cm at z=0, we shift the curve accordingly to z=0.5m

//Method 2.Mori and Wen's procedure for finding bubble size
dbm=0.65*((pi/4)*((dt*100)^2)*(uo-umf)*100)^0.4;
db2=dbm-(dbm-dbo)*exp(-0.3^(z/dt));

//Part(b).Bubble Velocity
//Method 1.Procedure using Eqn.(12)
ub1=1.6*((uo-umf)+1.13*db1^0.5)*(dt^1.35)+(0.711*(g*db1)^0.5);

//Method 2.Werther's Procedure
si=0.65;
ub2=si*(uo-umf)+2*(dt^0.5)*(0.711*(g*db1)^0.5);

//Using Eqn.(7) & Eqn.(8)
ubr1=0.711*(g*db1)^0.5;
ubr2=0.711*(g*db2/100)^0.5
ub3=uo-umf+ubr1;
ub4=uo-umf+ubr2;

//OUTPUT
printf('\nBubble Size');
mprintf('\nInitial bubble size from Fig.5.14 for %fm/s = %fcm',uo-umf,dbo);
printf('\n\n\tMethod 1.Werthers procedure for finding bubble size');
printf('\n\t\tHeight of bed(cm)');
printf('\t\t\tBubble size(cm)');
m=length(z1);
j=1;
while j<=m
    mprintf('\n\t\t%f',z1(j));
```

```
    mprintf('\t\t\t\t%f',db(j));
    j=j+1;
end
printf('\n\n\tMethod 2.Mori and Wens procedure for finding bubble size');
mprintf('\n\t\tMaximum expected bubble size=%fcm',dbm);
mprintf('\n\t\tBubble size=%fcm',db2);
printf('\nBubble Velocity');
printf('\n\n\tMethod 1.Procedure using Eqn.(12)');
mprintf('\n\t\tBubble velocity=%fm/s',ub1);
printf('\n\n\tMethod 2.Werthers procedure');
mprintf('\n\t\tBubble velocity=%fm/s',ub2);
printf('\nComparing the above results with the expressions of the simple two-phase theory');
printf('\n\tWerthers bubble size');
mprintf('\tBubble rise velocity=%fm/s\tBubble velocity=%fm/s',ubr1,ub3);
printf('\n\tMori & Wens bubble size');
mprintf('\tBubble rise velocity=%fm/s\tBubble velocity=%fm/s',ubr2,ub4);

//==========================================END OF PROGRAM==============================================================
```

Scilab code Exa 6.3 Scale down of a Commercial Chlorinator

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-6, Example 3, Page 153
//Title: Scale-down of a Commercial Chlorinator
```

```
//
clear
clc

//INPUT
dpbar=53;//Average particle size in micrometer
s=[1;2];//Size of Bermuda rock in cm
rhosbar=3200;//Average solid density of the coke-zircon mixture in kg/m^3
ephsilonm=0.5;//Void fraction for fixed bed
ephsilonf=0.75;//Void fraction for bubbling bed
rhogbar=0.64;//Average density of gas in kg/m^3
uo=14;//Superficial gas velocity in cm/s
myu=5E-5;//Viscosity of gas in kg/m s
T=1000;//Temperature in degree C
P=1;//Pressure in atm
dt=91.5;//ID of bed in cm
sh=150;//Slumped height in cm

//CALCULATION
rhog2=1.2;//Density of ambient air
myu2=1.8E-5;//Viscosity of ambient air
rhos2=rhog2*(rhosbar/rhogbar);//For the requirement of constant density ratio
m=((rhogbar*myu2)/(rhog2*myu))^(2/3);//Scale factor by usin Eqn.(16)
u2=(m^0.5)*uo;//Superficial gas velocity by using Eqn.(17)
//OUTPUT
printf('\nFor the model use');
mprintf('\n\tBed of ID %fcm\n\tSlumped bed height of %fcm\n\tPacked bed distributor consisting of %f-%fmm rock',m*dt,m*sh,m*s(1),m*s(2));
mprintf('\nFluidizing gas: ambient air at %fatm',P);
mprintf('\nSolids: \tzirconia, Average particle size=%fmicrometers',m*dpbar);
```

```
mprintf('\nEntering gas:\tSuperficial velocity=%fcm/
    s',u2);

//==========================================END OF PROGRAM
    ===============================================================
```

Scilab code Exa 6.4 Reactor Scale up for Geldart A Catalyst

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-6, Example 4, Page 159
//Title: Reactor Scale-up for Geldart A Catalyst
//
    ===============================================================

clear
clc

//INPUT
dtb=20;//ID of bench-scale reactor
dtp=1;//ID of pilot reactor
dpbar=52;//Average particle size in micrometer
ephsilonm=0.45;//Void fraction for fixed bed
ephsilonmf=0.50;//Void fraction at minimum
    fluidization condition
ephsilonmb=0.60;//Void fraction
uo=30;//Superficial gas velocity in cm/s
Lmb=2;//Length of fixed bed in m
umf=0.33;//Velocity at minimum fluidization
    condition in cm/s
umb=1;//Velocity at in cm/s
```

```
db=3;//Equilibrium bubble size in cm
g=9.80;//Acceleration due to gravity in m/s^2
pi=3.142857;

//CALCULATION
ubr=0.711*(g*db/100)^0.5;//Rise velocity of bubble using Eqn.(7)

//Bubble velocity for the bench unit
ubb1=1.55*(((uo-umf)/100)+14.1*((db/100)+0.005))*((dtb/100)^0.32)+ubr;//Bubble velocity using Eqn.(11)
si=1;
ubb2=si*((uo-umf)/100)+(3.2*((dtb/100)^(1/3)))*ubr;//Bubble velocity using Eqn.(9)
ubb=(ubb1+ubb2)/2;//Average bubble velocity

//Bubble velocity for the pilot unit
ubp1=1.55*(((uo-umf)/100)+14.1*((db/100)+0.005))*(dtp^0.32)+ubr;//Bubble velocity using Eqn.(11)
si=1;
ubp2=si*((uo-umf)/100)+(3.2*(dtp^(1/3)))*ubr;//Bubble velocity using Eqn.(9)
ubp=(ubp1+ubp2)/2;//Average bubble velocity

//Rise velocity of upflowing emulsion
ueb=ubb-ubr;//For the bench unit
uep=ubp-ubr;//For the pilot unit

//Scale-Up Alternative 1.
dteb=20;//Effective bubble diameter
dib=[5;10;15;20];//Different outside diameters
n=length(dib);
i=1;
while i<=n
    li(i)=sqrt(((pi*dib(i)*dteb)/4)+((pi/4)*(dib(i))^2));//Pitch using Eqn.(13)
    i=i+1;
```

```
end

//Scale-Up Alternative 2.
Lmp=Lmb*(ubp/ubb);//Static bed height of commercial unit
dtep=100;//Effective bubble diameter
dip=[10;15;20;25];//Different outside diameters
m=length(dip);
i=1;
while i<=m
    lip(i)=sqrt(((pi*dip(i)*dtep)/4)+(pi/4)*dip(i)); //Pitch using Eqn.(13)
    i=i+1;
end

//Height of Bubbling beds
//For bench unit
deltab=((uo/100)-(umb/100))/(ubb-(umb/100));//Fraction of bed in bubbles using Eqn.(28)
ephsilonfb=deltab+(1-deltab)*ephsilonmb;//Void fraction of bubbling bed using Eqn.(20)
Lfb=Lmb*(1-ephsilonm)/(1-ephsilonfb);//Hieght of bubbling bed usnig Eqn.(19)
//For pilot unit
deltap=((uo/100)-(umb/100))/(ubp-(umb/100));//Fraction of bed in bubbles using Eqn.(28)
ephsilonfp=deltap+(1-deltap)*ephsilonmb;//Void fraction of bubbling bed using Eqn.(20)
Lfp=Lmp*(1-ephsilonm)/(1-ephsilonfp);//Hieght of bubbling bed usnig Eqn.(19)

//OUTPUT
mprintf('\nRise velocity of bubble=%fm/s',ubr);
printf('\nFor the bench unit');
mprintf('\n\tWith Eqn.(11), Rise velocity=%fm/s',ubb1);
mprintf('\n\tWith Werthers procedure, Rise velocity=%fm/s',ubb2);
```

```
mprintf('\n\tAverage rise velocity=%fm/s',ubb);
mprintf('\n\tRise velocity of upflowing emulsion=%fm/s',ueb);
printf('\nFor the pilot unit');
mprintf('\n\tWith Eqn.(11), Rise velocity=%fm/s',ubp1);
mprintf('\n\tWith Werthers procedure, Rise velocity=%fm/s',ubp2);
mprintf('\n\tAverage rise velocity=%fm/s',ubp);
mprintf('\n\tRise velocity of upflowing emulsion=%fm/s',uep);
printf('\nScale-Up Alternative 1.');
printf('\n\tOuter diameter of tube(cm)');
printf('\tPitch(cm)');
n=length(dib);
j=1;
while j<=n
    mprintf('\n\t\t%f',dib(j));
    mprintf('\t\t\t%f',li(j));
    j=j+1;
end
printf('\n\tSuitable arrangement');
mprintf('\n\t\tOuter Diameter=%fcm\tPitch:Diameter ratio=%f',dib(2),(li(2)/dib(2)));
printf('\nScale-Up Alternative 2.');
mprintf('\n\tStatic bed height for commercial unit=%fm',Lmp);
printf('\n\tOuter diameter of tube(cm)');
printf('\tPitch(cm)');
n=length(dip);
j=1;
while j<=n
    mprintf('\n\t\t%f',dip(j));
    mprintf('\t\t\t%f',lip(j));
    j=j+1;
end
printf('\n\tSuitable arrangement');
mprintf('\n\t\tOuter Diameter=%fcm\tPitch:Diameter
```

```
ratio=%f',dip(2),(lip(2)/dip(2)));
printf('\n\n\t\t\t\tFraction of bed in bubbles\tVoid fraction of bed\tStatic bed height(m)\tHeight of bubbling bed(m)');
printf('\n\t\t\t\t------------------------------------------------------------------------');
mprintf('\nBench unit\tID=%fm\t%f\t\t\t%f\t\t%f\t\t%f',dtb/100,deltab,ephsilonfb,Lmb,Lfb);
mprintf('\nCommercial unit\tID=%fm\t%f\t\t\t%f\t\t%f\t\t%f',dtp,deltap,ephsilonfp,Lmp,Lfp);

//==================================END OF PROGRAM==================================================
```

Scilab code Exa 6.5 Reactor Scale up for Geldart B Catalyst

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-6, Example 5, Page 161
//Title: Reactor Scale-up for Geldart B Catalyst
//==========================================================================================

clear
clc

//INPUT
dtb=20;//ID of bench-scale reactor
dtp=1;//ID of pilot reactor
dpbar=200;//Average particle size in micrometer
```

```
ephsilonmf=0.50;//Void fraction at minimum
    fluidization condition
ephsilonmb=0.50;//Void fraction
uo=30;//Superficial gas velocity in cm/s
Lmb=2;//Length of fixed bed in m
umf=3;//Velocity at minimum fluidization condition
    in cm/s
umb=3;//Velocity at in cm/s
g=9.80;//Acceleration due to gravity in m/s^2
pi=3.142857;

//CALCULATION
//In the small bench unit
c=1;
ubb=c*((uo-umf)/100)+0.35*(g*(dtb/100))^0.5;//
    Velocity using Eqn.(5.22)
zsb=60*(dtb)^0.175;//Height using Eqn.(5.24)

//In the large pilot unit
ubp=c*((uo-umf)/100)+0.35*(g*dtp)^0.5;//Velocity
    using Eqn.(5.22)
zsp=60*(dtp*100)^0.175;//Height using Eqn.(5.24)

//OUTPUT
printf('\nCondition at which bubbles transform into
    slugs');
mprintf('\nFor tha small bench unit\n\t\tVelocity=
    %fm/s\n\t\tHeight above distributor plate=%fm',
    ubb,zsb/100);
mprintf('\nFor tha large pilot unit\n\t\tVelocity=
    %fm/s\n\t\tHeight above distributor plate=%fm',
    ubp,zsp/100);

//==========================================END OF PROGRAM
    ========================================================
```

Chapter 7

Entrainment and Elutriation from Fluidized Beds

Scilab code Exa 7.1 Entrainment from fine particle beds with high freeboard

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-7, Example 1, Page 179
//Title: Entrainment from Fine Particle Beds with
    High Freeboard
//
    ==========================================================

clear
clc

//INPUT
rhog=5.51;//Density of gas in kg/m^3
rhos=1200;//Density of solid in kg/m^3
dpbar=130;//Average size of particles in micrometer
uo=0.61;//Superficial gas velocity in m/s
```

```
g=9.80;//Acceleration due to gravity in m/s^2

//CALCULATION
//Assuming that freeboard in higher than TDH, computation of entrainment rate by Zenz & Weil's method
x=(uo^2)/(g*(dpbar*10^-6)*rhos^2);//Calculation of value of x-axis for Fig.(6), page 175
y=1.2;// Value of y-axis from Fig.(6)
Gsstar=y*rhog*uo;//Computation of rate of entrainment

//OUTPUT
mprintf('\nRate of entrainment=%fkg/m^2s',Gsstar);

//==========================================================END OF PROGRAM==========================================================================================
```

Scilab code Exa 7.2 Entrainment from large particle beds with high freeboard

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-7, Example 2, Page 180
//Title: Entrainment from Large Particle Beds with High Freeboard
//==========================================================================================================

clear
clc
```

```
//INPUT
x=0.2;//Fraction of fines in the bed
Gsstar=4.033320//Rate of entrainment in kg/m^2s(from
    Exa.1)

//CALCULATION
Gsstar1=x*Gsstar;//Rate of entrainment by Eqn.(3)

//OUTPUT
mprintf('\nRate of entrainment=%fkg/m^2s',Gsstar1);

//==========================================END OF PROGRAM
    ============================================================
```

Scilab code Exa 7.3 Entrainment from beds with a wide size distribution of solids

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-7, Example 3, Page 181
//Title: Entrainment from Beds with a Wide Size
    Distribution of Solids
//
    ============================================================================

clear
clc

//INPUT
rhog=5.51;//Density of gas in kg/m^3
```

```
rhos=1200;//Density of solid in kg/m^3
uo=0.61;//Superficial gas velocity in m/s
g=9.80;//Acceleration due to gravity in m/s^2
dp=[10;30;50;70;90;110;130];//Diameter of particle in micrometer
p=[0;0.0110;0.0179;0.0130;0.0058;0.0020;0];
pi=3.142857;
dt=6;

//CALCULATION
n=length(dp);
i=1;
while i<=n
    x(i)=(uo^2)/(g*(dp(i)*10^-6)*rhos^2);//Computation of value of x-axis for Fig.(6), page 175)
    i=i+1;
end
y=[40;12;6;3.2;2.;1.3;1];//Value of y-axis corresponding to each value of x-axis
y1 = y .* p;
i=1;
k=0;
while i<n
    y1(i)=(y(i)*p(i));
    k=k+((0.5)*(dp(i+1)-dp(i))*(y1(i+1)+y1(i)));//Integration using Trapezoidal rule
    i=i+1;
end
rhosbar=k*rhog;//Computation of solid loading
te=(pi/4)*(dt^2)*rhosbar*uo;//Computation of total entrainment

//OUTPUT
mprintf('\nSolid loading =%fkg/m^3',rhosbar);
mprintf('\nTotal Entrainment =%fkg/s',te);

//=========================================END OF PROGRAM
```

Scilab code Exa 7.4 kstar from steady state experiments

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-7, Example 4, Page 181
//Title: k* from Steady State Experiments
//
   ==========================================================

clear
clc

//INPUT
dp=[40;60;80;100;120];//Diameter of particle in
   micrometer
uo=0.381;//Superficial gas velocity in m/s

//CALCULATION
Gs=0.9;//Rate of entrainment in kg/m^2 s from Fig.3(
   a)
pb=(1/100)*[0.45;1.00;1.25;1.00;0.60];//Size
   distribution for bed particles from Fig.3(b)
pe=(1/100)*[1.20;2.00;1.25;0.45;0.10];//Size
   distribution for entrained particles from Fig.3(b
   )
n=length(dp);
i=1;
while i<=n
    ki(i)=(Gs*pe(i))/pb(i);//Calculation of ki*
```

```
        using Eqn.(13)
    i=i+1;
end

//OUTPUT
printf('\ndpi(micrometer)');
printf('\t100pb(dpi)(micrometer^-1)');
printf('\t100pe(dpi)(micrometer^-1)');
printf('\tki*(kg/m^2 s)');
j=1;
while j<=n
    mprintf('\n%f',dp(j));
    mprintf('\t%f',100*pb(j));
    mprintf('\t\t\t%f',100*pe(j));
    mprintf('\t\t\t%f',ki(j));
    j=j+1;
end

//==========================================END OF PROGRAM
    ================================================================
```

Scilab code Exa 7.5 Comparing predictions for kstar

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-7, Example 5, Page 181
//Title: Comparing Predictions for k*
//
    ==========================================================================
clear
```

```
clc

//INPUT
rhog=1.217;//Density of gas in kg/m^3
myu=1.8E-5;//Viscosity of gas in kg/m s
umf=0.11;//Velocity at minimum fluidization condition in m/s
rhos=2000;//Density of solid in kg/m^3
uo=1.0;//Superficial gas velocity in m/s
g=9.80;//Acceleration due to gravity in m/s^2
dp=[30;40;50;60;80;100;120];//Diameter of particle in micrometer
uti=[0.066;0.115;0.175;0.240;0.385;0.555;1.0];//Terminal velocity of particles in m/s

//CALCULATION
n=length(dp);
i=1;
while i<=n
    //Using Yagi & Aochi's correlation
    Ret(i)=(rhog*(uti(i))*dp(i)*10^-6)/myu;
    kistar1(i)=((myu*((uo-uti(i))^2))/(g*(dp(i)*10^-6)^2))*(0.0015*(Ret(i)^0.5)+(0.01*(Ret(i)^1.2)));
    //Using Wen & Hasinger's correlation
    kistar2(i)=(((1.52E-5)*((uo-uti(i))^2)*rhog)/(g*dp(i)*10^-6)^0.5)*(Ret(i)^0.725)*((rhos-rhog)/rhog)^1.15;
    //Using Merrick & Highley's correlation
    kistar3(i)=uo*rhog*(0.0001+130*exp(-10.4*((uti(i)/uo)^0.5)*((umf/(uo-umf))^0.25)));
    //Using Geldart's correlation
    kistar4(i)=23.7*uo*rhog*exp(-5.4*(uti(i)/uo));
    //Using Zenz & Weil's procedure
    x1(i)=(uo^2)/(g*(dp(i)*10^-6)*rhos^2);//Computation of value of x-axis for Fig.(6), page 175)
    y1=[12.2;8.6;6.4;4.9;2.75;1.8;1.2];//Value of y-
```

```
axis corresponding to each value of x-axis
    kistar5(i)=y1(i)*rhog*uo;
    //Using Gugnoni & Zenz's procedure
    x2(i)=(uo-uti(i))/((g*dp(i)*10^-6)^0.5);//Computation of value of x-axis for Fig.(6), page 175)
    y=[5.8;5.4;3.2;2.8;1.3;0.6;0];//Value of y-axis corresponding to each value of x-axis
    kistar6(i)=y(i)*rhog*uo;
    i=i+1;
end

i=1;
printf('dp(micrometer)');
printf('\tYagi & Aochi');
printf('\tWen & Hashinger');
printf('\t\tMerrick & Highley');
printf('\tGeldart et al.');
printf('\t\tZenz & Well');
printf('\t\tGugnoni & Zenz');
while i<=n
    mprintf('\n%f',dp(i));
    mprintf('\t%f',kistar1(i));
    mprintf('\t%f',kistar2(i));
    mprintf('\t\t%f',kistar3(i));
    mprintf('\t\t%f',kistar4(i));
    mprintf('\t\t%f',kistar5(i));
    mprintf('\t\t%f',kistar6(i));
    i=i+1;
end

//Note: There is huge deviation of the calculated answer and the answer given in the textbook for the correlation of Merrick & Highley.  There is a contradiction in the correlation used in the problem and the one given in page 179.
//We tried to retrieve the original paper i.e. D. Merrick and J.Highley, AICHE J., 6, 220(1960).
```

```
But the effort was not fruitful.

//=============================================END OF PROGRAM
=================================================================
```

Scilab code Exa 7.6 Entrainment from a short vessel

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-7, Example 6, Page 190
//Title: Entrainment from a Short Vessel Ht<TDH
//
    ================================================================================

clear
clc

//INPUT
dpbar=60;//Average size of particles in micrometer
rhog=1.3;//Density of gas in kg/m^3
rhos=1500;//Density of solid in kg/m^3
umf=0.003;//Velocity at minimum fluidization
    condition in m/s
uo=0.503;//Superficial gas velocity in m/s
g=9.80;//Acceleration due to gravity in m/s^2
Hf=2;//Height at which the cyclone inlet is to be
    located in m

//CALCULATION
y=(uo^2)/(g*(dpbar*10^-3)*rhos^2);//Calculation of
    value of y-axis for Fig.(6), page 175
```

```
x=1;//Value of x-axis from Fig.(6), page 175
Gsstar=x*rhog*uo;//Computation of rate of entrainment
Gsuo=5.0;//Ejection rate pf particles in kg/m^2 s from Fig.(11), page 188
a=0.72/uo;//From Fig.(12), page 189
Gs=Gsstar+(Gsuo-Gsstar)*exp(-a*Hf);
p=((Gs-Gsstar)/Gsstar)*100;

//OUTPUT
mprintf('\nRate of entrainment from short bed=%fkg/m^2s',Gs);
mprintf('\nThis entrainment is %f percent higher than it would be if the gas exit were at the TDH',p);

//==================================================END OF PROGRAM==================================================
```

Chapter 8

High velocity Fluidization

Scilab code Exa 8.1 Performance of a Fast Fluidized Vessel

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-8, Example 1, Page 206
//Title: Performance of a Fast Fluidized Vessel
//
   ==========================================================

clear
clc

//INPUT
Lmf=2.4;//Length of bed at minimum fluidized
   condition in m
uo=[2;4;6];//Superficial gas velocity in m/s
GsII=100;//Solid circulation rate in kg/m^2 s for
   Mode II
uoIII=4;//Superficial gas velocity in m/s for Mode
   III
GsIII=[42;50;100;200;400];//Solid circulation rate
```

```
in kg/m^2 s for Mode III
GsIV=[70;100;120];//Solid circulation rate in kg/m^2
    s for Mode IV
dt=0.4;//Column diamter in m
Ht=10;//Height of column in m
rhos=1000;//Density of solid in kg/m^3
dpbar=55;//Particle diameter in micrometer
ephsilonmf=0.5;//Void fraction at minimum
   fluidization condition

//CALCULATION
//Mode I
ephsilonstar=0.01;//Saturation carrying capacity of
   gas
ephsilonsd=[0.2;0.16;0.14];//Solid holdup in lower
   dense region from Fig.8(b) for various uo
n=length(uo);
i=1;
Hfguess=2;//Guess value of height
while i<=n
    a(i)=3/uo(i);//Decay constant
    function[fn]=solver_func(Hf)//Function defined
       for solving the system
        fn=Lmf*(1-ephsilonmf)-((ephsilonsd(i)-(
           ephsilonstar+(ephsilonsd(i)-ephsilonstar)
           *exp(-a(i)*Hf)))/a(i))-Ht*ephsilonsd(i)+
           Hf*(ephsilonsd(i)-ephsilonstar);
    endfunction
    [Hf(i)]=fsolve(Hfguess,solver_func,1E-6);//Using
        inbuilt function fsolve for solving Eqn.(10)
        for Hf
    Hd(i)=Ht-Hf(i);//Height of lower densce region
    ephsilonse(i)=ephsilonstar+(ephsilonsd(i)-
       ephsilonstar)*exp(-a(i)*Hf(i));//Solid holdup
        at exit
    GsI(i)=rhos*uo(i)*ephsilonse(i);//Solid
       circulation rate from Eqn.(4)
    i=i+1;
```

```
end

//Mode II
i=1;
Hfguess2=2;//Guess value of height
while i<=n
    ephsilonseII(i)=GsII/(rhos*uo(i));//Solid holdup
        at exit
    function[fn]=solver_func1(Hf)//Function defined
       for solving the system
        fn=ephsilonseII(i)-ephsilonstar-(ephsilonsd(
           i)-ephsilonstar)*exp(-a(i)*Hf);//From Eqn
           .(7)
    endfunction
    [HfII(i)]=fsolve(Hfguess2,solver_func1,1E-6);//
       Using inbuilt function fsolve for solving Eqn
       .(10) for Hf
    HdII(i)=Ht-HfII(i);//Height of lower dense
       region
    //Length of bed minimum fluidization condtion
    LmfII(i)=(1-ephsilonmf)^-1*[((ephsilonsd(i)-
       ephsilonseII(i))/a(i))+Ht*ephsilonsd(i)-HfII(
       i)*(ephsilonsd(i)-ephsilonstar)];
    i=i+1;
end

//Mode III
aIII=3/uoIII;//Decay constant
ephsilonsdIII=0.16;//Solid holdup at lower dense
   region
i=1;
m=length(GsIII);
Hfguess3=2;//Guess value of height
while i<=m
    ephsilonseIII(i)=GsIII(i)/(rhos*uoIII);//Solid
       holdup at exit
    function[fn]=solver_func2(Hf)//Function defined
       for solving the system
```

```
        fn=ephsilonseIII(i)-ephsilonstar-(ephsilonsdIII-ephsilonstar)*exp(-aIII*Hf);//From Eqn.(7)
    endfunction
    [HfIII(i)]=fsolve(Hfguess3,solver_func2,1E-6);//Using inbuilt function fsolve for solving Eqn.(10) for Hf
    HdIII(i)=Ht-HfIII(i);//Height of lower dense region
    //Length of bed at minimum fluidization condition
    LmfIII(i)=(1-ephsilonmf)^-1*[((ephsilonsdIII-ephsilonseIII(i))/aIII)+Ht*ephsilonsdIII-HfIII(i)*(ephsilonsdIII-ephsilonstar)];
    i=i+1;
end

//Mode IV
i=1;
Hfguess4=2;//Guess value of height
while i<=n
    aIV(i)=3/uo(i);//Decay constant
    ephsilonseIV(i)=GsIV(i)/(rhos*uo(i));//Solid holdup at exit
    function[fn]=solver_func3(Hf)//Function defined for solving the system
        fn=ephsilonseIV(i)-ephsilonstar-(ephsilonsd(i)-ephsilonstar)*exp(-aIV(i)*Hf);//From Eqn.(7)
    endfunction
    [HfIV(i)]=fsolve(Hfguess4,solver_func3,1E-6);//Using inbuilt function fsolve for solving Eqn.(10) for Hf
    HdIV(i)=Ht-HfIV(i);//Height of lower dense region
    //Length of bed at minimum fluidization condition
    LmfIV(i)=(1-ephsilonmf)^-1*[((ephsilonsd(i)-
```

```
        ephsilonseIV(i))/aIV(i))+Ht*ephsilonsd(i)-HfIV(i)*(ephsilonsd(i)-ephsilonstar)];
    i=i+1;
end

//OUTPUT
printf('\nMode I');
printf('\n\tuo(m/s)\t\tephsilonse(-)\tHf(m)\t\tHd(m)\t\tGs(kg/m^2 s)');
i=1;
while i<=n
    mprintf('\n\t%f\t%f\t%f\t%f\t%f',uo(i),ephsilonse(i),Hf(i),Hd(i),GsI(i));
    i=i+1;
end
printf('\nMode II');
printf('\n\tuo(m/s)\t\tephsilonse(-)\tHf(m)\t\tHd(m)\t\tLmf(m))');
i=1;
while i<=n
    mprintf('\n\t%f\t%f\t%f\t%f\t%f',uo(i),ephsilonseII(i),HfII(i),HdII(i),LmfII(i));
    i=i+1;
end
printf('\nMode III');
printf('\n\tGs(kg/m^ s)\tephsilonse(-)\tHf(m)\t\tHd(m)\t\tLmf(m)');
i=1;
while i<=m
    mprintf('\n\t%f\t%f\t%f\t%f\t%f',GsIII(i),ephsilonseIII(i),HfIII(i),HdIII(i),LmfIII(i));
    i=i+1;
end
printf('\nMode IV');
printf('\n\tuo(m/s)\t\tGs(kg/m^2 s)\tephsilonse(-)\tHf(m)\t\tLmf(m)');
i=1;
```

```
while i<=n
    mprintf('\n\t%f\t%f\t%f\t%f\t%f',uo(i),GsIV(i),
       ephsilonseIV(i),HfIV(i),LmfIV(i));
    i=i+1;
end

//=========================================END OF PROGRAM
```

Chapter 9

Solid Movement Mixing Segregation and Staging

Scilab code Exa 9.1 Vertical Movement of Solids

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-9, Example 1, Page 218
//Title: Vertical Movement of Solids
//
    ==========================================================

clear
clc

//INPUT
umf=0.015;//Velocity at minimum fluidization
    condition in m/s
ephsilonmf=0.5;//Void fraction at minimum
    fluidization condition
uo=0.1;//Superficial gas velocity in m/s
```

```
delta=0.2;//Bed fraction in bubbles
db=0.06;//Equilibrium bubble size in m
dt=[0.1;0.3;0.6;1.5];//Various vessel sizes in m
ub=[0.4;0.75;0.85;1.1];//Bubble velocity in m/s
Dsv=[0.03;0.11;0.14;0.23];//Reported values of vertical dispersion coefficient

//CALCULATION
n=length(ub);
i=1;
fw1=2;//Wake fraction from Hamilton et al.
fw2=0.32;//Wake fraction from Fig.(5.8)
fw=(fw1+fw2)*0.5;//Average value of wake fraction
while i<=n
    Dsv1(i)=12*((uo*100)^0.5)*((dt(i)*100)^0.9);//Vertical distribution coefficient from Eqn.(3)
    Dsv2(i)=(fw^2*ephsilonmf*delta*db*ub(i)^2)/(3*umf);//Vertical distribution coefficient from Eqn.(12)
    i=i+1;
end

//OUTPUT
printf('\n\t\tVertical dispersion coefficient(m^2/s)');
printf('\nVessel Size(m)');
printf('\tFrom Experiment');
printf('\tFrom Eqn.(3)');
printf('\tFrom Eqn.(12)');
i=1;
while i<=n
    mprintf('\n%f',dt(i));
    mprintf('\t%f',Dsv(i));
    mprintf('\t%f',Dsv1(i)/10^4);
    mprintf('\t%f',Dsv2(i));
    i=i+1;
end
```

```
//==================================================END OF PROGRAM==================================================

```

Scilab code Exa 9.2 Horizontal Drift Of Solids

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-9, Example 2, Page 222
//Title: Horizontal Drift Of Solids
//==========================================================================================================

clear
clc

//INPUT
Lmf=0.83;//Length of bed at minimum fluidization condition in m
dp=450;//Average particle size in micrometer
ephsilonmf=0.42;//Void fraction at minimum fluidization condition
umf=0.17;//Velocity at minimum fluidization condition in m/s
uo=[0.37;0.47;0.57;0.67];//Superficial gas velocity in m/s
Dsh=[0.0012;0.0018;0.0021;0.0025];//Horizontal Drift Coefficient from Experiment in m^2/s
db=[0.10;0.14];//Equilibrium bubble size in m
g=9.81;//Acceleration due to gravity in m/s^2
```

```
//CALCULATION
n=length(uo);
m=length(db);
j=1;
i=1;
k=1;
alpha=0.77;//Since we are not dealing with Geldart A or AB solids
uf=umf/ephsilonmf;
for j = 1:m
        for i = 1:n
            ubr(k)=0.711*(db(j)*g)^0.5;//Rise velocity of a single bubble in m/s
            ub(k)=uo(i)-umf+ubr(k);//Rise velocity of bubbles in a bubbling bed
            delta(k)=(uo(i)-umf)/(ub(k)+umf);//Bed fraction in bubbles
            if ubr(i)>uf then Dshc(k)=(3/16)*(delta(k)/(1-delta(k)))*((alpha^2*db(j)*ubr(k)*[(((ubr(k)+2*uf)/(ubr(k)-uf))^(1/3))-1]));//Horizontal Distribution coeff. from Eqn.(14)
            else Dsh(k)=(3/16)*(delta/(1-delta))*(alpha^2*umf*db/ephsilonmf);//Horizontal Distribution coeff. from Eqn.(15)
            end
            Dshc(k)=(3/16)*(delta(k)/(1-delta(k)))*((alpha^2*db(j)*ubr(k)*[(((ubr(k)+2*uf)/(ubr(k)-uf))^(1/3))-1]));//Horizontal Distribution coeff. from Eqn.(14)
            i=i+1;
            k=k+1;
        end
    i=1;
```

```
    j=j+1;
end

//OUTPUT
i=1;
j=1;
k=1;
while k<=m*n
    mprintf('\nSnce we do not have ub=%fm/s>>uf=%fm/s we use Eqn.(14).',ub(k),uf)
    printf('\nGas Velocity(m/s)');
    printf('\tHorizontal Drift Coefficient Calculated(m^2/s)');
    printf('\tHorizontal Drift Coefficient from Experiment(m^2/s)');
    while j<=m
        mprintf('\ndb=%fm',db(j));
        while i<=n
            mprintf('\n%f',uo(i));
            mprintf('\t\t%f',Dshc(k));
            mprintf('\t\t\t\t\t%f',Dsh(i));
            i=i+1;
            k=k+1;
        end
    i=1;
    j=j+1;
    end
end

//=====================================END OF PROGRAM
```

Scilab code Exa 9.3 Design of Baffle Plates

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-9, Example 3, Page 232
//Title: Design of Baffle Plates
//
   ==========================================================

clear
clc

//INPUT
Gsup=1.5;//Solid interchange rate in kg/m^2plate s
dor=19.1;//Orifice diameter in mm
dp=210;//Particle size in micrometer
uo=0.4;//Superficial gas velocity in m/s
fopen=[0.12;0.17;0.26];//Open area fraction
pi=3.14;

//CALCULATION
n=length(fopen);
i=1;
while i<=n
    uor(i)=uo/fopen(i);//Gas velocity through the
       orifice
    ls1(i)=Gsup/fopen(i);//Flux of solids through
       the holes
    i=i+1;
end
ls2=[12;20;25];//Flux of solids through holes from
   Fig.13(c) for different uor values
fopen1=0.12;//Open area fraction which gives
   reasonable fit
lor=sqrt(((pi/4)*dor^2)/fopen1);//Orifice spacing

//OUTPUT
```

```
printf('\nfopen');
printf('\t\tuor(m/s)');
printf('\tls from Eqn.(18)');
printf('\tls from Fig.13(c)');
i=1;
while i<=n
    mprintf('\n%f',fopen(i));
    mprintf('\t%f',uor(i));
    mprintf('\t%f',ls1(i));
    mprintf('\t\t%f',ls2(i));
    i=i+1;
end
mprintf('\n\nFor square pitch, the orifice spacing
   should be %fmm',lor);

//====================================END OF PROGRAM
   ======================================================
```

Chapter 10

Gas Dispersion and Gas Interchange in Bubbling Beds

Scilab code Exa 10.1 Estimate Interchange Coefficients in Bubbling Beds

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-10, Example 1, Page 253
//Title: Estimate Interchange Coefficients in
   Bubbling Beds
//
   ==========================================

clear
clc

//INPUT
umf=[0.01;0.045];//Velocity at minimum fluidization
   condition in m/s
ephsilonmf=[0.5;0.5];//Void fraction at minimum
   fluidization condition
```

```
D=[2E-5;7E-5];//Diffusion coefficient of gas in m^2/s
g=9.81;//Acceleration due to gravity in m/s^2

//CALCULATION
db=[5;10;15;20];
n=length(umf);
m=length(db)'
for i = 1:n
    for j = 1:m
            Kbc(i,j)=4.5*(umf(i)/db(j))+5.85*((D(i)^0.5*g^0.25)/db(j)^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(27)
            Kce(i,j)=6.77*((D(i)*ephsilonmf(i)*0.711*(g*db(j))^0.5)/db(j)^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(34)
            Kbe(i,j)=(Kbc(i,j)*Kce(i,j))/(Kbc(i,j)+Kce(i,j));//Gas interchange coefficient between bubble and emulsion from Eqn.(14)
    end;
end

//OUTPUT
i=1;
j=1;
k=1;
while k<=m*n
    printf('\n\t\tKbc for fine particles and He');
    printf('\tKbc for coarse particles and ozone');
    printf('\tKbe for fine particles and He');
    printf('\tKbe for coarse particles and ozone');
    while j<=m
        mprintf('\ndb=%fm',db(j)*10^-2);
        while i<=n
            mprintf('\t%f',Kbc(k));
```

```
            mprintf('\t\t\t%f',Kbe(k));
            i=i+1;
            k=k+1;
            printf('\t\t\t');
        end
    i=1;
    j=j+1;
    end
end
Kbe=Kbe';
Kbc=Kbc';
plot2d("ll",db,[Kbc Kbe]);
xtitle('Plot of Kbc,Kbe vs db','db',['Kbc','Kbe']);
printf('\nComparing the points with the plot of Kbc,
   Kbe vs db in Fig.(12), we can conlcude the
   following:');
printf('\nKbc for fine particles and helium: line 2
   in Fig.(12)');
printf('\nKbc for coarser particles and ozone: line
   3 in Fig.(12)');
printf('\nKbe for fine particles and helium: line 4
   in Fig.(12)');
printf('\nKbe for coarser particles and ozone: line
   5 in Fig.(12)');

//============================================END OF PROGRAM
   ======================================================
```

Scilab code Exa 10.2 Compare the Relative Importance of Kbc and Kce

```
//Kunii D., Levenspiel O., 1991. Fluidization
```

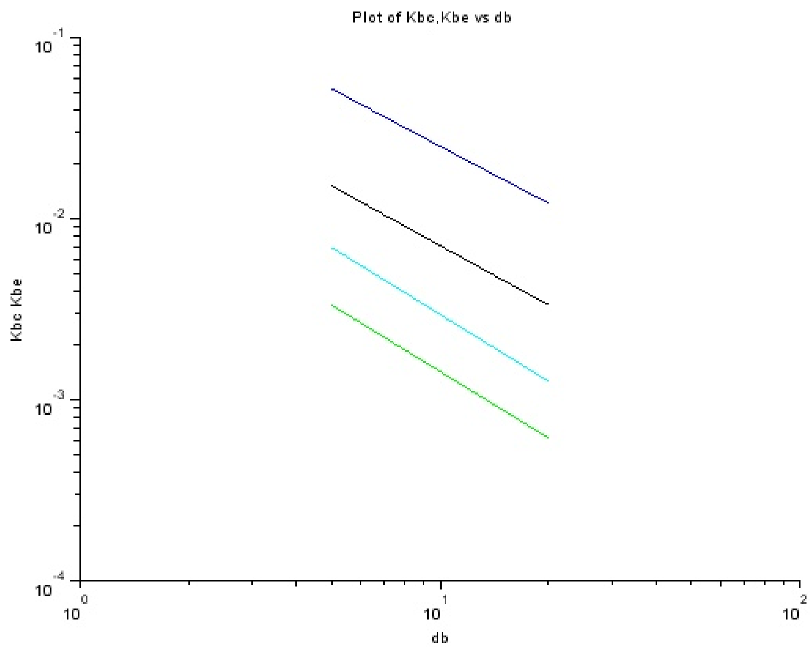

Figure 10.1: Estimate Interchange Coefficients in Bubbling Beds

```
Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-10, Example 2, Page 254
//Title: Compare the Relative Importance of Kbc and Kce
//==========================================================================

clear
clc

//INPUT
D=0.69;//Diffusion coefficient of gas in cm^2/s
umf=1.0;//Velocity at minimum fluidization condition in cm/s
ephsilonmf=0.5;//Void fraction at minimum fluidization condition
db=[5;15];//Equilibrium bubble size in cm
g=980;//Acceleration due to gravity in cm/s^2

//CALCULATION
n=length(db);
i=1;
while i<=n
    Kbc(i)=4.5*(umf/db(i))+5.85*((D^0.5*g^0.25)/db(i)^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(27)
    Kce(i)=6.77*((D*ephsilonmf*0.711*(g*db(i))^0.5)/db(i)^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(34)
    Kbe(i)=(Kbc(i)*Kce(i))/(Kbc(i)+Kce(i));//Gas interchange coefficient between bubble and emulsion from Eqn.(14)
    e(i)=(Kce(i)-Kbe(i))/Kbe(i);//Error when minor resistance is ignored
    i=i+1;
```

```
end

//OUTPUT
printf('\ndb(cm)');
printf('\t\tCalculated Kbc');
printf('\tCalculated Kce');
printf('\t\tKbe from Eqn.(14)');
printf('\tErron when minor resistance is ignored (in
    percentage)');
i=1;
while i<=n
    mprintf('\n%f',db(i));
    mprintf('\t%f',Kbc(i));
    mprintf('\t%f',Kce(i));
    mprintf('\t\t%f',Kbe(i));
    mprintf('\t\t%f',e(i)*100);
    i=i+1;
end

//=========================================END OF PROGRAM
    ==============================================================
```

Scilab code Exa 10.3 Compare Interchange Rates for Adsorbed and Non-adsorbed Gases

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-10, Example 3, Page 255
//Title: Compare Interchange Rates for Adsorbed and
    Nonadsorbed Gases
//
```

```

clear
clc

//INPUT
Kbe=[0.028;0.05];//Reported range for gas interchange coefficient between bubble and emulsion
uo=0.30;//Superficial gas velocity in m/s
db=0.13;//Equilibrium bubble size in m
m=7;
ephsilonmf=0.5;//Void fraction at minimum fluidization condition
umf=0.0018;//Velocity at minimum fluidization condition in m/s
D=[9E-6;22E-6];//Diffusion coefficient of gas in m^2/s
g=9.81;//Acceleration due to gravity in m/s^2

//CALCULATION
n=length(Kbe);
i=1;
while i<=n
    Kbem(i)=(6/db)*Kbe(i);//Gas interchange coefficient between bubble and emulsion from Eqn.(19)
    Kbc(i)=4.5*(umf/db)+5.85*((D(i)^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(27)
    Kce(i)=6.77*((D(i)*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(34)
    Kbe(i)=(Kbc(i)*Kce(i))/(Kbc(i)+Kce(i));//Gas interchange coefficient between bubble and emulsion from Eqn.(14)
    c(i)=(Kbem(i)/Kbe(i));
```

```
    i=i+1;
end

//OUTPUT
printf('\nKbe from Eqn.(19)');
printf('\tKbc from Eqn.(27)');
printf('\tKce from Eqn.(34)');
printf('\tKbe from Eqn.(14)');
printf('\tComparison of Kbe from Eqn.(19) and that
   from Eqn.(14)');
i=1;
while i<=n
    mprintf('\n%f',Kbem(i));
    mprintf('\t\t%f',Kbc(i));
    mprintf('\t\t%f',Kce(i));
    mprintf('\t\t%f',Kbe(i));
    mprintf('\t\t%f',c(i));
    i=i+1;
end

//========================================END OF PROGRAM
```

Chapter 11

Particle to Gas Mass and Heat Transfer

Scilab code Exa 11.1 Fitting Reported Mass Transfer Data with the Bubbling Bed Model

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-11, Example 1, Page 265
//Title: Fitting Reported Mass Transfer Data with
   the Bubbling Bed Model
//
   ==========================================================================

clear
clc

//INPUT
db=0.37;//Equilibrium bubble size in cm
dp=0.028;//Particle size in cm
rhos=1.06;//Density of solids in g/cc
```

```
ephsilonmf=0.5;//Void fraction at minimum fluidization condition
phis=0.4;//Sphericity of solids
gammab=0.005;//Ratio of volume of dispersed solids to that of bubble phase
rhog=1.18E-3;//Density of air in g/cc
myu=1.8E-4;//Viscosity of gas in g/cm s
D=0.065;//Diffusion coefficient of gas in cm^2/s
Sc=2.35;//Schmidt number
etad=1;//Adsorption efficiency factor
y=1;
umf=1.21;//Velocity at minimum fluidization condition in cm/s
ut=69;//Terminal velocity in cm/s
g=980;//Acceleration due to gravity in square cm/s^2
uo=[10;20;30;40;50];//Superficial gas velocity in cm/s

//CALCULATION
n=length(uo);
i=1;
Rept=(dp*ut*rhog)/myu;
Shstar=2+(0.6*(Rept^0.5)*(Sc^(1/3)));//Sherwood no. from Eqn.(1)
Kbc=4.5*(umf/db)+5.85*((D^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
ubr=0.711*(g*db)^0.5;//Rise velocity of the bubble
while i<=n
    x(i)=(uo(i)-umf)/(ubr*(1-ephsilonmf));//The term delta/(1-epshilonf) after simplification
    Shbed(i)=x(i)*[(gammab*Shstar*etad)+((phis*dp^2*y)/(6*D))*Kbc];//Sherwood no. from Eqn.(11)
    Rep(i)=(dp*uo(i)*rhog)/myu;//Reynolds of the particle
    i=i+1;
end

```

```
//OUTPUT
printf('\nThe desired result is the relationship
   between Shbed and Rep  The points gives a
   straight line of the form y=mx+c');
printf('\nRep');
printf('\t\tShbed');
i=1;
while i<=n
    printf('\n%f',Rep(i));
    printf('\t%f',Shbed(i));
    i=i+1;
end
plot(Rep,Shbed);
xlabel("Rep");
ylabel("Shbed");

//=====================================END OF PROGRAM
   ========================================================
```

Scilab code Exa 11.2 The Effect of m on Bubble Emulsion Interchange

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-11, Example 2, Page 267
//Title: The Effect of m on Bubble-Emulsion
   Interchange
//
   ========================================================
```

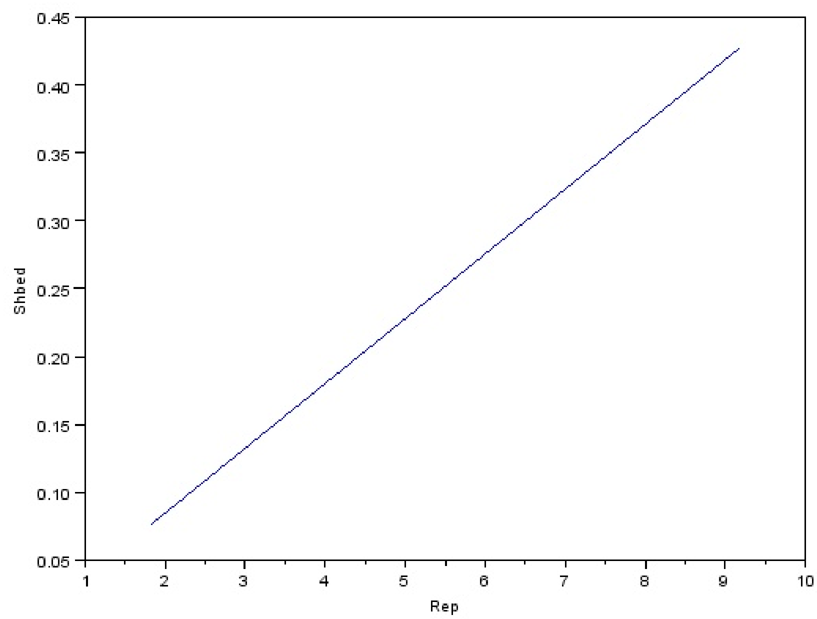

Figure 11.1: Fitting Reported Mass Transfer Data with the Bubbling Bed Model

```

clear
clc

//INPUT
umf=0.12;//Velocity at minimum fluidization condition in cm/s
uo=40;//Superficial gas velocity in cm/s
ub=120;//Velocity of the bubble in cm/s
D=0.7;//Diffusion coefficient of gas in cm^2/s
abkbe1=1;//Bubble-emuslion interchange coefficient for non absorbing particles(m=0)
abkbe2=18;//Bubble-emuslion interchange coefficient for highly absorbing particles(m=infinity)
g=980;//Acceleration due to gravity in square cm/s^2

//CALCULATION
//For non absorbing particles m=0,etad=0
Kbc=(ub/uo)*(abkbe1);
dbguess=2;//Guess value of db
function[fn]=solver_func(db)//Function defined for solving the system
    fn=abkbe1-(uo/ub)*(4.5*(umf/db)+5.85*(D^0.5*g^0.25)/(db^(5/4)));//Eqn.(10.27)
endfunction
[d]=fsolve(dbguess,solver_func,1E-6);//Using inbuilt function fsolve for solving Eqn.(10.27) for db
//For highly absorbing particles m=infinity, etad=1
M=abkbe2-(uo/ub)*Kbc;
//For intermediate condition
alpha=100;
m=10;
etad=1/(1+(alpha/m));//Fitted adsorption efficiency factor from Eqn.(23)
abkbe3=M*etad+(uo/ub)*Kbc;

//OUTPUT
mprintf('\nFor non absorbing particles:\n\tDiameter
```

```
of bubble=%fcm\n\tBubble-cloud interchange coefficient=%fs^-1',d,Kbc);
mprintf('\nFor highly absorbing partilces:\n\tM=%f',M);
mprintf('\nFor intermediate condition:\n\tFitted adsorption efficiency factor:%f\n\tBubble-emuslion interchange coefficient:%fs^-1',etad,abkbe3);

//==============================================END OF PROGRAM==================================================================
```

Scilab code Exa 11.3 Fitting Reported Heat Transfer Data with the Bubbling Bed Model

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-11, Example 3, Page 273
//Title: Fitting Reported Heat Transfer Data with the Bubbling Bed Model
//==========================================================================================

clear
clc

//INPUT
rhos=1.3;//Density of solids in g/cc
phis=0.806;//Sphericity of solids
gammab=0.001;//Ratio of volume of dispersed solids
```

```
to that of bubble phase
rhog=1.18E-3;//Density of air in g/cc
Pr=0.69;//Prandtl number
myu=1.8E-4;//Viscosity of gas in g/cm s
Cpg=1.00;//Specific heat capacity of gas in J/g K
ephsilonmf=0.45;//Void fraction at minimum fluidization condition
kg=2.61E-4;//Thermal concuctivity of gas in W/cm k
dp=0.036;//Particle size in cm
umf=6.5;//Velocity at minimum fluidization condition in cm/s
ut=150;//Terminal velocity in cm/s
db=0.4;//Equilibrium bubble size in cm
etah=1;//Efficiency of heat transfer
uo=[10;20;30;40;50];//Superficial gas velocity in cm/s
g=980;//Acceleration due to gravity in square cm/s^2

//CALCULATION
Nustar=2+[((dp*ut*rhog)/myu)^0.5*Pr^(1/3)];//Nusselt no. from Eqn.(25)
Hbc=4.5*(umf*rhog*Cpg/db)+5.85*((kg*rhog*Cpg)^0.5*g^0.25/db^(5/4));//Total heat interchange across the bubble-cloud boundary from Eqn.(32)
ubr=0.711*(g*db)^0.5;//Rise velocity of the bubble from Eqn.(6.7)
n=length(uo);
i=1;
while i<=n
    x(i)=(uo(i)-umf)/(ubr*(1-ephsilonmf));//The term delta/(1-epshilonf) after simplification
    Nubed(i)=x(i)*[gammab*Nustar*etah+(phis*dp^2/(6*kg))*Hbc];//Nusselt no. from Eqn.(36)
    Rep(i)=(dp*uo(i)*rhog)/myu;//Reynolds of the particle
    i=i+1;
end

```

```
//OUTPUT
printf('\nThe desired result is the relationship between Nubed and Rep which is in the form of a straight line y=mx+c');
printf('\nRep');
printf('\t\tNubed');
i=1;
while i<=n
    printf('\n%f',Rep(i));
    printf('\t%f',Nubed(i));
    i=i+1;
end
plot(Rep,Nubed);
xlabel("Rep");
ylabel("Nubed");

//==========================================================END OF PROGRAM==========================================================
```

Scilab code Exa 11.4 Heating a Particle in a Fluidized Bed

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-11, Example 4, Page 274
//Title: Heating a Particle in a Fluidized Bed
//==========================================================================================================================

```

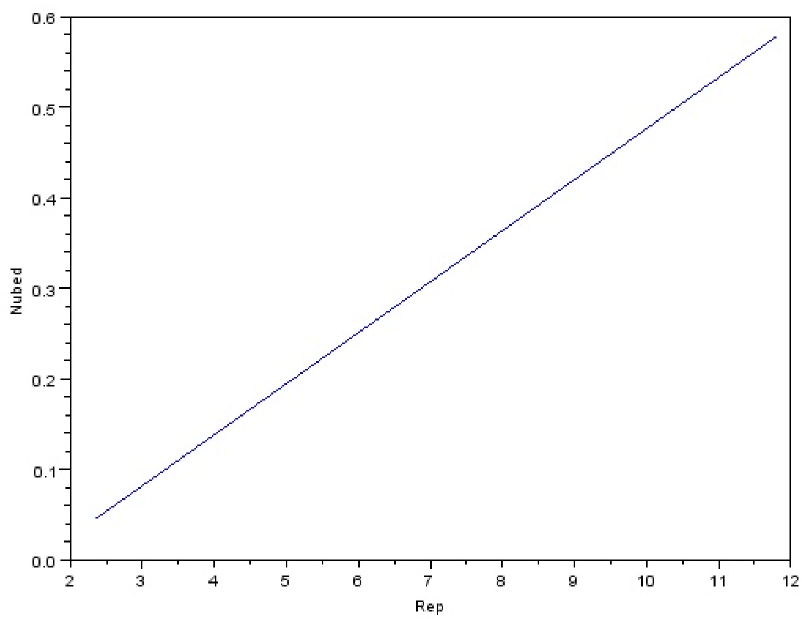

Figure 11.2: Fitting Reported Heat Transfer Data with the Bubbling Bed Model

```
clear
clc

//INPUT
rhog=1.2;//Density of air in kg/m^3
myu=1.8E-5;//Viscosity of gas in kg/m s
kg=2.6E-2;//Thermal concuctivity of gas in W/m k
dp=1E-4;//Particle size in m
rhos=8920;//Density of solids in kg/m^3
Cps=390;//Specific heat capacity of the solid in J/
    kg K
ephsilonf=0.5;//Void fraction of the fluidized bed
umf=0.1;//Velocity at minimum fluidization condition
     in m/s
uo=0.1;//Superficial gas velocity in m/s
pi=3.14

//CALCULATION
to=0;//Initial temperature of the bed
T=100;//Temperature of the bed
t=0.99*T;//Particle temperature i.e. when it
    approaches 1% of the bed temperature
mp=(pi/6)*dp^3*rhos;//Mass of the particle
A=pi*dp^2;//Surface area of the particle
Rep=(dp*uo*rhog)/myu;//Reynold's no. of the particle
Nubed=0.0178;//Nusselt no. from Fig.(6)
hbed1=(Nubed*kg)/dp;//Heat transfer coefficient of
    the bed
t1=(mp*Cps/(hbed1*A))*log((T-to)/(T-t));//Time
    needed for the particle approach 1 percentage of
    the bed temperature in case(a)
hbed2=140*hbed1;//Since from Fig.(6) Nup is 140
    times Nubed
t2=(mp*Cps/(hbed2*A))*log((T-to)/(T-t));//Time
    needed for the particle approach 1 percentage of
    the bed temperature in case(b)

//OUTPUT
```

```
printf('\nCase(a):Using the whole bed coefficient from Fig.(6)');
mprintf('\n\tTime needed for the particle approach 1 percentage of the bed temperature is %fs',t1);
printf('\nCase(b):Uisng the single-particle coefficient of Eqn.(25),also shown in Fig.(6)');
mprintf('\n\tTime needed for the particle approach 1 percentage of the bed temperature is %fs',t2);

//==================================END OF PROGRAM==================================
```

Chapter 12

Conversion of Gas in Catalytic Reactions

Scilab code Exa 12.1 Fine Particle Geldart A Bubbling Bed Reactor

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-12, Example 1, Page 293
//Title: Fine Particle (Geldart A) Bubbling Bed
   Reactor
//
   ==========================================================

clear
clc

//INPUT
Kr=10;//rate constant in m^3 gas/m^3 cat s
D=2E-5;//Diffusion coefficient of gas in m^2/s
dpbar=68;//Average partilce size in micrometers
ephsilonm=0.5;//Void fraction of fixed bed
```

```
gammab=0.005;//Ratio of volume of dispersed solids to that of bubble phase
ephsilonmf=0.55;//Void fraction at minimum fluidization condition
umf=0.006;//Velocity at minimum fluidization condition in m/s
db=0.04;//Equilibrium bubble size in m
Lm=0.7;//Length of the bed in m
uo=0.1;//Superficial gas velocity in m/s
dbed=0.26;//Diameter of the bed in m
g=9.81;//Acceleration due to gravity in square m/s^2

//CALCULATION
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from Eqn.(6.7)
ub=uo-umf+ubr;//Velocity of bubbles in bubbling beds in Eqn.(6.8)
Kbc=4.5*(umf/db)+5.85*((D^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
Kce=6.77*((D*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)
delta=uo/ub;//Fraction of bed in bubbles from Eqn.(6.29)
fw=0.6;//Wake volume to bubble volume from Fig.(5.8)
gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw);//Volume of solids in cloud to that of the bubble from Eqn.(6.36)
gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-gammac;//Volume of solids in emulsion to that of the bubble from Eqn.(6.35)
ephsilonf=1-(1-delta)*(1-ephsilonmf);//Void fraction of fixed bed from Eqn.(6.20)
Lf=(1-ephsilonm)*Lm/(1-ephsilonf);//Length of fixed bed from Eqn.(6.19)
Krtou=Kr*Lm*(1-ephsilonm)/uo;//Dimensionless reaction rate group from Eqn.(5)
```

```
Kf=gammab*Kr+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)
    +(1/(gammae*Kr))))));//Raction rate for fluidized
     bed from Eqn.(14)
XA=1-exp(-1*Kf*Lf/ub);//Conversion from Eqn.(16)

//OUTPUT
mprintf('\nThe dimnesionless reaction rate group: %f
    ',Krtou);
mprintf('\nThe reaction rate for fluidized bed: %fs
    ^-1',Kf);
mprintf('\nConversion: %f',XA);

//==========================================END OF PROGRAM
    ==============================================================
```

Scilab code Exa 12.2 Commercial Sized Phthalic Anhydride Reactor

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-12, Example 2, Page 298
//Title: Commercial-Sized Phthalic Anhydride Reactor
//
    ==========================================================================


clear
clc

//INPUT
umf=0.005;//Velocity at minimum fluidization
    condition in m/s
```

```
ephsilonm=0.52;//Void fraction of fixed bed
ephsilonmf=0.57;//Void fraction at minimum fluidization condition
DA=8.1E-6;//Diffusion coefficient of gas in m^2/s
DR=8.4E-6;//Diffusion coefficient of gas in m^2/s
Lm=5;//Length of the bed in m
dte=1;//Diameter of tube in m
Kr1=1.5;//rate constant in m^3 gas/m^3 cat s
Kr3=0.01;//rate constant in m^3 gas/m^3 cat s
gammab=0.005;//Ratio of volume of dispersed solids to that of bubble phase
uo=0.45;//Superficial gas velocity in m/s
db=0.05;//Equilibrium bubble size in m from Fig.(6.8)
ub=1.5;//Velocity of bubbles in bubbling bed in m/s from Fig.(6.11(a))
g=9.81;//Acceleration due to gravity in square m/s^2

//CALCULATION
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from Eqn.(6.7)
KbcA=4.5*(umf/db)+5.85*((DA^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
KceA=6.77*((DA*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)
KbcR=4.5*(umf/db)+5.85*((DR^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
KceR=6.77*((DR*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)
delta=uo/ub;//Fraction of bed in bubbles from Eqn.(6.29)
fw=0.6;//Wake volume to bubble volume from Fig.(5.8)
gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw);//Volume of solids in cloud to that of the
```

```
bubble from Eqn.(6.36)
gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-gammac;//Volume of solids in emulsion to that of the bubble from Eqn.(6.35)
ephsilonf=1-(1-delta)*(1-ephsilonmf);//Void fraction of fixed bed from Eqn.(6.20)
Lf=(1-ephsilonm)*Lm/(1-ephsilonf);//Length of fixed bed from Eqn.(6.19)
Krtou=Kr1*Lm*(1-ephsilonm)/uo;//Dimensionless reaction rate group from Eqn.(5)
Kr12=Kr1;//Since the reactions are a special case of Denbigh scheme
Kr34=Kr3;
Kf1=(gammab*Kr12+1/((1/KbcA)+(1/(gammac*Kr12+1/((1/KceA)+(1/(gammae*Kr12)))))))*(delta/(1-ephsilonf));//Rate of reaction 1 for fluidized bed from Eqn.(14)
Kf3=(gammab*Kr34+1/((1/KbcR)+(1/(gammac*Kr34+1/((1/KceR)+(1/(gammae*Kr34)))))))*(delta/(1-ephsilonf));//Rate of reaction 2 for fluidized bed from Eqn.(14)
Kf12=Kf1;
Kf34=Kf3;
KfA=[[KbcR*KceA/gammac^2+(Kr12+KceA/gammac+KceA/gammae)*(Kr34+KceR/gammac+KceR/gammae)]*delta*KbcA*Kr12*Kr34/(1-ephsilonf)]/[[(Kr12+KbcA/gammac)*(Kr12+KceA/gammae)+Kr12*KceA/gammac]*[(Kr34+KbcR/gammac)*(Kr34+KceR/gammae)+Kr34*KceR/gammac]];//Rate of raection with respect to A from Eqn.(35)
KfAR=Kr1/Kr12*Kf12-KfA;//Rate of reaction from Eqn.(34)
tou=Lf*(1-ephsilonf)/uo;//Residence time from Eqn.(5)
XA=1-exp(-Kf1*tou);//Conversion of A from Eqn.(26)
XR=1-((KfAR/(Kf12-Kf34))*[exp(-Kf34*tou)-exp(-Kf12*tou)]);//Conversion of R from Eqn.(27)
SR=(1-XR)/XA;//Selectivity of R
```

```
//OUTPUT

mprintf('\nRate of reaction 1 for fluidized bed:%f',Kf1);
mprintf('\nRate of reaction 2 for fluidized bed:%f',Kf3);
mprintf('\nRate of reaction 1 with respect to A:%f',KfA);
mprintf('\nThe Conversion of Napthalene:%f percentage',XA*100);
mprintf('\nThe selectivity of Phthalic anhydride:%f percentage',SR*100);

//==========================================END OF PROGRAM==============================================
```

Scilab code Exa 12.3 Bubbling Bed Reactor for Intermediate Sized Reactor

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-12, Example 3, Page 302
//Title: Bubbling Bed Reactor for Intermediate Sized Reactor
//==========================================================================================

clear
clc
```

```

//INPUT
Kr=3;//rate constant in m^3 gas/m^3 cat s
db=0.12;//Equilibrium bubble size in m
D=9E-5;//Diffusion coefficient of gas in m^2/s
dpbar=68;//Average partilce size in micrometers
ephsilonm=0.42;//Void fraction of fixed bed
uo=0.4;//Superficial gas velocity in m/s
Lm=0.8;//Length of the bed in m
ephsilonmf=0.45;//Void fraction at minimum fluidization condition
umf=0.21;//Velocity at minimum fluidization condition in m/s
gammab=0;//Ratio of volume of dispersed solids to that of bubble phase
g=9.81;//Acceleration due to gravity in square m/s^2

//CALCULATION
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from Eqn.(6.7)
ub=uo-umf+ubr;//Velocity of bubbles in bubbling beds in Eqn.(6.8)
ubstar=ub+3*umf;//Rise velocity of the bubble gas from Eqn.(45)
delta=(uo-umf)/(ub+umf);//Fraction of bed in bubbles from Eqn.(6.46)
Kbe=4.5*(umf/db);//Interchange coefficient between bubble and emulsion from Eqn.(47)
Lf=Lm*(1-ephsilonm)/((1-delta)*(1-ephsilonmf));//Length of fixed bed
phi=[(Kr/Kbe)^2*{(1-ephsilonmf)-gammab*(umf/ubstar)}^2+((delta/(1-delta))+umf/ubstar)^2+2*(Kr/Kbe)*{(1-ephsilonmf)-gammab*(umf/ubstar)}*((delta/(1-delta))-umf/ubstar)]^0.5;//From Eqn.(52)
q1=0.5*Kr/umf*{(1-ephsilonmf)+gammab*(umf/ubstar)}+0.5*Kbe/umf*{((delta/(1-delta))+umf/ubstar)-phi};//From Eqn.(50)
q2=0.5*Kr/umf*{(1-ephsilonmf)+gammab*(umf/ubstar)
```

```
}+0.5*Kbe/umf*{((delta/(1-delta))+umf/ubstar)+phi};//From Eqn.(50)
si1=0.5-0.5*((1-delta)/delta)*[umf/ubstar-Kr/Kbe*{(1-ephsilonmf)-gammab*(umf/ubstar)}-phi];//From Eqn.(51)
si2=0.5-0.5*((1-delta)/delta)*[umf/ubstar-Kr/Kbe*{(1-ephsilonmf)-gammab*(umf/ubstar)}+phi];//From Eqn.(51)
XA=1-(delta/(1-delta))*(1/(uo*phi))*[(1-si2)*{si1*delta*ubstar+(1-delta)*umf}*exp(-q1*Lf)+(si1-1)*{si2*delta*ubstar+(1-delta)*umf}*exp(-q2*Lf)];//Conversion from Eqn.(49)
Krtou=Kr*Lm*(1-ephsilonm)/uo;//Dimensionless reaction rate group from Eqn.(5)

//OUTPUT
mprintf('\nCOmparing the values of 1-XA = %f and Krtou = %f with Fig.(6), we can conlcude that this operating condition is shown as point A in Fig.(3)',1-XA,Krtou);
printf('\nLine 2 gives the locus of conversions for different values of the reaction rate group for this fluidized contacting');

//==================================================END OF PROGRAM
==========================================================================
```

Scilab code Exa 12.4 Reaction in the Slow Bubble Regime

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

```

```
//Chapter-12, Example 4, Page 305
//Title: Reaction in the Slow Bubble Regime
//==========================================================================

clear
clc

//INPUT
uo=0.25;//Superficial gas velocity in m/s
db=0.025;//Equilibrium bubble size in m
Kr=1.5;//rate constant in m^3 gas/m^3 cat s
umf=0.21;//Velocity at minimum fluidization condition in m/s
Lm=0.8;//Length of the bed in m
ephsilonm=0.42;//Void fraction of fixed bed
g=9.81;//Acceleration due to gravity in square m/s^2

//CALCULATION
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from Eqn.(6.7)
ub=uo-umf+ubr;//Velocity of bubbles in bubbling beds in Eqn.(6.8)
delta=(uo-umf)/(ub+2*umf);//Fraction of bed in bubbles from Eqn.(55) since ub/umf<<1
XA=1-exp(-Kr*Lm*((1-ephsilonm)/uo)*(umf/uo)*(1-delta));//Conversion from Eqn.(57)
Krtou=Kr*Lm*(1-ephsilonm)/uo;//Dimensionless reaction rate group from Eqn.(5)


//OUTPUT
mprintf('\nComparing the values of 1-XA = %f and Krtou = %f with Fig.(6), we can conlcude that this operating condition is shown as point B in Fig.(3)',1-XA,Krtou);
printf('\nLine 3 gives the locus of conversions for
```

```
different values of the reaction rate group for
this fluidized contacting');

//=========================================================END OF PROGRAM
=====================================================================
```

Scilab code Exa 12.5 Conversion in the Freeboard of a Reactor

```
//Kunii D., Levenspiel O., 1991. Fluidization
Engineering(II Edition). Butterworth-Heinemann,
MA, pp 491

//Chapter-12, Example 5, Page 307
//Title: Conversion in the Freeboard of a Reactor
//
==========================================================================

clear
clc

//INPUT
uo=0.3;//Superficial gas velocity in m/s
Lf=1.1;//Length of fixed bed in m
Hf=1.2;//Length of freeboard in m
db=0.04;//Equilibrium bubble size in m
umf=0.006;//Velocity at minimum fluidization
condition in m/s
ephsilonmf=0.55;//Void fraction at minimum
fluidization condition
gammab=0.005;//Ratio of volume of dispersed solids
to that of bubble phase
Kr=10;//rate constant in m^3 gas/m^3 cat s
```

```
D=2E-5;//Diffusion coefficient of gas in m^2/s
g=9.81;//Acceleration due to gravity in square m/s^2

//CALCULATION
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from Eqn.(6.7)
ub=uo-umf+ubr;//Velocity of bubbles in bubbling beds in Eqn.(6.8)
Kbc=4.5*(umf/db)+5.85*((D^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
Kce=6.77*((D*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)
delta=uo/ub;//Fraction of bed in bubbles from Eqn.(6.29)
ephsilonf=1-(1-delta)*(1-ephsilonmf);//Void fraction of fixed bed from Eqn.(6.20)
fw=0.6;//Wake volume to bubble volume from Fig.(5.8)
gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw);//Volume of solids in cloud to that of the bubble from Eqn.(6.36)
gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-gammac;//Volume of solids in emulsion to that of the bubble from Eqn.(6.35)
Kf=(gammab*Kr)+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)+(1/(gammae*Kr))))));//Raction rate for fluidized bed from Eqn.(14)
XA=1-exp(-1*Kf*Lf/ub);//Conversion at the top of dense bed from Eqn.(16)
etabed=(Kf*delta)/(Kr*(1-ephsilonf));//Reactor efficiency from Eqn.(22)
a=0.6/uo//Since uoa = 0.6s^-1 from Fig.(5)
adash=6.62;//From Fig.(5)
XA1=1-1/(exp(((1-ephsilonf)*Kr/(uo*a))*[(1-exp(-a*Hf))-((1-etabed)/(1+(adash/a)))*(1-exp(-(a+adash)*Hf))]));//Conversion from Eqn.(64)
XA2=1-(1-XA1)*(1-XA);//Conversion at the exit from
```

```
    Eqn.(64)

//OUTPUT
printf('\nThe conversion:');
mprintf('\n\tAt the top pf the dense bed: %f
    percentage',XA*100);
mprintf('\n\tAt the reactor exit: %f percentage',XA2
    *100);

//Disclaimer: The value of kf deviate from the one
    given in textbook, where as it is close to the
    value obtained by manual calculation.
//=============================================END OF PROGRAM
    ====================================================================
```

Chapter 13

Heat Transfer between Fluidized Beds and Surfaces

Scilab code Exa 13.1 h on a Horizontal Tube Bank

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-13, Example 1, Page 331
//Title: h on a Horizontal Tube Bank
//
    ==========================================================

clear
clc

//INPUT
dp=57;//Particle size in micrometer
rhos=940;//Density of solids in kg/m^3
Cps=828;//Specific heat capacity of the solid in J/
    kg K
ks=0.20;//Thermal conductivity of solids in W/m k
```

```
kg=0.035;//Thermal concuctivity of gas in W/m k
umf=0.006;//Velocity at minimum fluidization
    condition in m/s
ephsilonmf=0.476;//Void fraction at minimum
    fluidization condition
do1=0.0254;//Outside diameter of tube in m
L=1;
uo=[0.05;0.1;0.2;0.35];//Superficial gas velocity in
     m/s
nw=[2;3.1;3.4;3.5];//Bubble frequency in s^-1
g=9.81;//Acceleration due to gravity in square m/s^2


//CALCULATION
dte=4*do1*L/2*L;//Hydraulic diameter from Eqn.(6.13)
db=(1+1.5)*0.5*dte;//Rise velocity of the bubble
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from
    Eqn.(6.7)
phib=0.19;//From Fig.(15) for ks/kg=5.7
ke=ephsilonmf*kg+(1-ephsilonmf)*ks*[1/((phib*(ks/kg)
    )+(2/3))];//Effective thermal conductivity of bed
     from Eqn.(3)
n=length(uo);
i=1;
while i<=n
    ub(i)=uo(i)-umf+ubr;//Velocity of bubbles in
        bubbling beds in Eqn.(6.8)
    delta(i)=uo(i)/ub(i);//Fraction of bed in
        bubbles from Eqn.(6.29)
    h(i)=1.13*[ke*rhos*(1-ephsilonmf)*Cps*nw(i)*(1-
        delta(i))]^0.5;//Heat transfer coefficinet
        from Eqn.(18)
    i=i+1;
end

//OUTPUT
printf('\nSuperficial gas velocity(m/s)');
printf('\tHeat transfer coefficient(W/m^2 k)');
```

```
i=1;
while i<=n
    mprintf('\n%f',uo(i));
    mprintf('\t\t\t%f',h(i));
    i=i+1;
end

//==========================================END OF PROGRAM
    =============================================================
```

Scilab code Exa 13.2 Effect of Gas Properties on h

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-13, Example 2, Page 332
//Title: Effect of Gas Properties on h
//
    ==========================================================================


clear
clc

//INPUT
dp=80;//Particle size in micrometer
rhos=2550;//Density of solids in kg/m^3
Cps=756;//Specific heat capacity of the solid in J/
    kg K
ks=1.21;//Thermal conductivity of solids in W/m k
kg=[0.005;0.02;0.2];//Thermal concuctivity of gas in
     W/m k
```

```
ephsilonmf=0.476;//Void fraction at minimum fluidization condition

//CALCULATION
delta=0.5*(0.1+0.3);//For a gently fluidized bed
nw=3;//Bubble frequency in s^-1 from Fig.(5.12) at about 30cm above the distributor
n=length(kg);
i=1;
while i<=n
    x(i)=ks/kg(i);//To find different values of ks/kg
    i=i+1;
end
phib=[0.08;0.10;0.20];//From Fig.(15) for different values of ks/kg
i=1;
while i<=n
    ke(i)=ephsilonmf*kg(i)+(1-ephsilonmf)*ks*[1/((phib(i)*(ks/kg(i)))+(2/3))];//Effective thermal conductivity of bed from Eqn.(3)
    h1(i)=1.13*[ke(i)*rhos*(1-ephsilonmf)*Cps*nw*(1-delta)]^0.5;//Heat transfer coefficinet from Eqn.(18)
    i=i+1;
end

//OUTPUT
printf('\nThermal conductivity of Gas(W/m K))');
printf('\tMax. heat transfer coefficient(W/m^2 k)');
i=1;
while i<=n
    mprintf('\n%f',kg(i));
    mprintf('\t\t\t\t%f',h1(i));
    i=i+1;
end

//==========================================END OF PROGRAM
```

Scilab code Exa 13.3 Effect of Particle Size on h

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-13, Example 3, Page 332
//Title: Effect of Particle Size on h
//==========================================================================

clear
clc

//INPUT
rhos=2700;//Density of solids in kg/m^3
Cps=755;//Specific heat capacity of the solid in J/kg K
ks=1.2;//Thermal conductivity of solids in W/m k
kg=0.028;//Thermal concuctivity of gas in W/m k
ephsilonmf=0.476;//Void fraction at minimum fluidization condition
dp1=10E-3;//Particle size for which h=hmax in m
hmax=250;//Max. heat transfer coefficient in W/m^2 K
nw=5;//Bubble frequency in s^-1
delta=0.1;//Fraction of bed in bubbles
deltaw=0.1;//Fraction of bed in bubbles in wall region
dp=2E-3;//Diameter of particle in m

```

```
//CALCULATION
x=ks/kg;
phib=0.11;
phiw=0.17;
ke=ephsilonmf*kg+(1-ephsilonmf)*ks*[1/((phib*(ks/kg))+(2/3))];//Effective thermal conductivity of bed from Eqn.(3)
hpacket=1.13*[ke*rhos*(1-ephsilonmf)*Cps*nw/(1-deltaw)]^0.5;//Heat transfer coefficient for the packet of emulsion from Eqn.(11)
ephsilonw=ephsilonmf;//Void fraction in the wall region
kew=ephsilonw*kg+(1-ephsilonw)*ks*[(phiw*(ks/kg)+(1/3))^-1];//Effective thermal conductivity in the wall region with stagnant gas from Eqn.(4)
y=(2*kew/dp1)+(hmax*hpacket)/(((1-deltaw)*hpacket)-hmax);//Calculating the term alphaw*Cpg*rhog*uo from Eqn.(16) by rearranging it
h=(1-deltaw)/((2*kew/dp+y*(dp/dp1)^0.5)^-1+hpacket^-1);//Heat transfer coeeficient from Eqn.(11) by using the value of y

//OUTPUT
mprintf('\nThe heat transfer coefficient for paricle size of %fm = %fW/m^2 K',dp,h);

//=========================================END OF PROGRAM
```

Scilab code Exa 13.4 Freeboard Heat Exchange

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann,
```

```
    MA, pp 491

//Chapter-13, Example 4, Page 334
//Title: Freeboard Heat Exchange
//
    ==========================================================================

clear
clc

//INPUT
Hf=4;//Height of freeboard in m
uo=2.4;//Superficial gas velocity in m/s
ho=350;//Heat transfer coefficient at the bottom of
    freeboard region in W/m^2 K
hg=20;//Heat transfer coefficient in equivalent gas
    stream, but free of solids in W/m^2 K

//CALCULATION
zf=[0;0.5;1;1.5;2;2.5;3;3.5;Hf];//Height above the
    top of the dense bubbling fluidized bed
hr=0;//Assuming heat transfer due to radiation is
    negligible
a=1.5/uo;//Since decay coefficient from Fig.(7.12),
    a*uo=1.5s^-1
n=length(zf);
i=1;
while i<=n
    h(i)=(hr+hg)+(ho-hr-hg)*exp(-a*zf(i)/2);//Heat
        transfer coefficient from Eqn.(24) for zf=Hf
    i=i+1;
end
hbar=(hr+hg)+2*(ho-hr-hg)*(1-exp(-a*Hf/2))/(a*Hf);//
    Mean heat transfer coefficient for the 4-m high
    freeboard from Eqn.(26)

//OUTPUT
```

```
printf('\nThe required relationship is h(W/m^2 K) vs
    . zf(m) as in Fig.(9a)');
printf('\nHeight above the dense bubbling fluidized
    bed(m))');
printf('\tHeat transfer coefficient(W/m^2 k)');
i=1;
while i<=n
    mprintf('\n%f',zf(i));
    mprintf('\t\t\t\t\t%f',h(i));
    i=i+1;
end
mprintf('\n\nThe mean heat transfer coefficient for
    the 4-m high freeboard =%fW/m^2 K',hbar);

//=========================================END OF PROGRAM
    ==============================================================
```

Chapter 14

The RTD and Size Distribution of Solids in Fluidized Beds

Scilab code Exa 14.1 Flow with Elutriation

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-14, Example 1, Page 343
//Title: Flow with Elutriation
//
   ==========================================================================

clear
clc

//INPUT
Fo=2.7;//Feed rate in kg/min
Fof=0.9;//Feed rate of fines in feed in kg/min
Foc=1.8;//Feed rate of coarse in feed in kg/min
W=17;//Bed weight in kg
kf=0.8;//Elutriation of fines in min^-1
```

```
kc=0.0125;//Elutriation of coarse in min^-1

//CALCULATION
F1guess=1;//Guess value of F1
function[fn]=solver_func(F1)//Function defined for solving the system
    fn=F1-(Fof/(1+(W/F1)*kf))-(Foc/(1+(W/F1)*kc));//Eqn.(17)
endfunction
[F1]=fsolve(F1guess,solver_func,1E-6);//Inbuilt function fsolve to solve for F1
F1f=Fof/(1+(W/F1)*kf);//Flow rate of fines in entrained streams from Eqn.(16)
F1c=Foc/(1+(W/F1)*kc);//Flow rate of coarse in entrained streams from Eqn.(16)
F2f=Fof-F1f;//Flow rate of fines in overflow streams from Eqn.(9)
F2c=Foc-F1c;//Flow rate of coarse in overflow streams from Eqn.(9)
tbarf=1/((F1/W)+kf);//Mean residence time of fines from Eqn.(12)
tbarc=1/((F1/W)+kc);//Mean residence time of coarse from Eqn.(12)

//OUTPUT
mprintf('\nFlow rate in entrained stream:\n\tFines:%fkg/min\n\tCoarse:%fkg/min',F1f,F1c);
mprintf('\nFlow rate in overflow stream:\n\tFines:%fkg/min\n\tCoarse:%fkg/min',F2f,F2c);
mprintf('\nMean residence time:\n\tFines:%fmins\n\tCoarse:%fmins',tbarf,tbarc);

//========================================END OF PROGRAM========================================
```

Scilab code Exa 14.2 Flow with Elutriation and Change in Density of Solids

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-14, Example 2, Page 344
//Title: Flow with Elutriation and Change in Density of Solids
//==========================================================================================

clear
clc

//INPUT
dt=4;//Diameter of reactor in m
ephsilonm=0.4;//Void fraction of static bed
rhos=2500;//Density of solid in the bed in kg/m^3
Lm=1.2;//Height of static bed in m
Fo=3000;//Feed rate in kg/hr
beta1=1.2;//Increase in density of solids
dp=[3;4;5;6;7;8;9;10;11;12;3;14;16;18;20;22;24;26;28;30]*10^-2; //Size of particles in mm
po=[0;0.3;0.8;1.3;1.9;2.6;3.5;4.4;5.7;6.7;7.5;7.8;7.5;6.3;5.0;3.6;2 //Size distribution of solids in mm^-1
k=[0;10;9.75;9.5;8.75;7.5;6.0;4.38;2.62;1.20;0.325;0;0;0;0;0;0;0;0 //Elutriation constant in s^-1
pi=3.14;
```

```

//CALCULATION
W=(pi/4*dt^2)*Lm*(1-ephsilonm)*rhos;//Weight of solids in bed
n=length(dp);
i=1;
F1guess=1000;//Guess value for F1
F1c=2510:10:2700;
while i<=n
    function[fn]=solver_func(F1)//Function defined for solving the system
        if k(i)==0  then    x(i)=0; break
                    else    x(i)=(po(i)/(W*k(i)/F1))*log(1+(W*k(i)/F1));
        end
    fn=F1/(Lm*Fo)-x(i);
    endfunction
    [F1(i)]=fsolve(F1guess,solver_func,1E-6);//Using inbuilt function fsolve for solving Eqn.(20) for F1
    c(i)=F1c(i)/(Lm*Fo);
    if F1(i)==0 then a(i)=0;
    else     a(i)=(po(i)/(W*k(i)/F1(i)))*log(1+(W*k(i)/F1(i)));
    end
    i=i+1;
end
plot(F1,a,F1,c);
xtitle('F1 vs a,c','F1','a,c');
F1n=2500;//The point were both the curves meet
F2=beta1*Fo-F1n;//Flow rate of the second leaving stream
j=1;
m=length(dp);
while j<=m
    p1(j)=(1/F1n)*((Fo*po(j))/(1+(W/F1n)*k(j)));//Size distribution of stream 1 in mm^-1 from Eqn.(16)
```

```
    p2(j)=k(j)*W*p1(j)/F2;//Size distribution of stream 2 in mm^-1 from Eqn.(7)
    if p1(j)==0 & p2(j)==0 then tbar(j)=0;
    else  if  p1(j)==0 then tbar(j)=(W*p1(j))/(F2*p2(j));
        else if  p2(j)==0 then tbar(j)=(W*p1(j))/(F1n*p1(j));
            else tbar(j)=(W*p1(j))/(F1n*p1(j)+F2*p2(j));//Average time in hr from Eqn.(11)
            end
        end
    end
    j=j+1;
end

//OUTPUT
printf('\nFlow rate of stream 1:%fkg/hr',F1n);
printf('\nFlow rate of stream 2:%fkg/hr',F2);
j=1;
mprintf('\ntbar(hr)');
while j<=m
    mprintf('\n%f',tbar(j));
    j=j+1;
end

//==================================================END OF PROGRAM
```

```
//DISCLAIMER: The value obtained for tbar is deviating highly form the one given in textbook. However, the value obtained by manual calculation is close to the  ones obtained from the program.
```

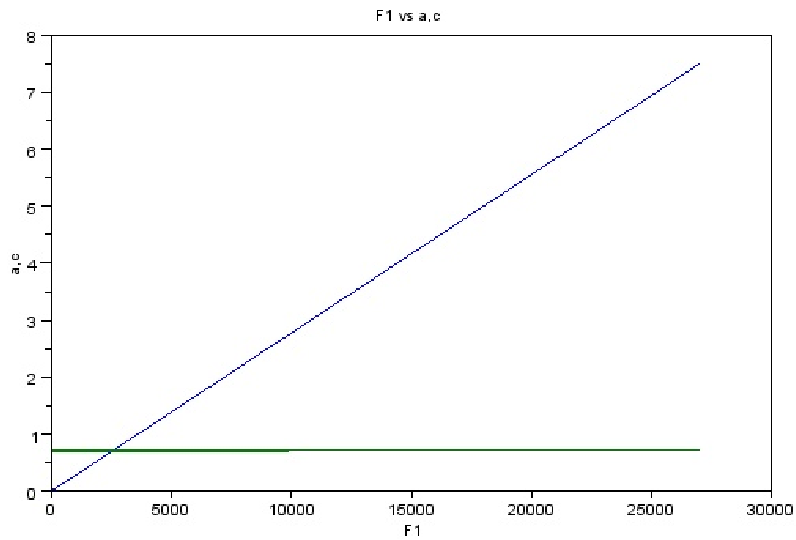

Figure 14.1: Flow with Elutriation and Change in Density of Solids

Scilab code Exa 14.3 Single Size Feed of Shrinking Particles

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-14, Example 3, Page 351
//Title: Single-Size Feed of Shrinking Particles
//
    ==========================================================

clear
```

```
clc

//INPUT
dp=1;//Particle size in mm
Fo=10;//Feed rate in kg/min
k=0.1;//Particle shrinkage rate in mm/min

//CALCULATION
R=k/2;//Particle shrinkage rate in terms of radius
W=(Fo*dp/2)/(4*R);//Bed weight from Eqn.(42)

//OUTPUT
printf('\nWeight of bed:%fkg',W);

//=========================================END OF PROGRAM
    ==============================================================
```

Scilab code Exa 14.4 Wide Size Distribution of Shrinking Particle

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-14, Example 4, Page 352
//Title: Wide Size Distribution of Shrinking
    Particle
//
    ==========================================================================

clear
clc

```

```
//INPUT
dpi=[1.05;0.95;0.85;0.75;0.65;0.55;0.45;0.35;0.25;0.15;0.05]; //Mean size in mm
Fo=[0;0.5;3.5;8.8;13.5;17.0;18.2;17.0;13.5;7.3;0]*10^-2; //Feed rate in kg/s
k=[0;0;0;0;0;0;0;0;2.0;12.5;62.5]*10^-5; //Elutriation constant in s^-1
R=-1.58*10^-5; //Rate of particle shrinkage in mm/s
deldpi=0.1; //Size intervals in mm

//CALCULATION
n=length(dpi);
m=2; //Starting with the largest value size interval that contains solids
W(m-1)=0;
while m<=n
   W(m)=(Fo(m)-R*W(m-1)/deldpi)/(k(m)-R/deldpi-3*R/dpi(m)); //From Eqn.(33)
   m=m+1;
end
Wt=sum(W); //Total sum

//OUTPUT
printf('\nTotal mass in the bed:%fkg',Wt);

//==========================================END OF PROGRAM==========================================
```

Scilab code Exa 14.5 Elutriation and Attrition of Catalyst

```
//Kunii D., Levenspiel O., 1991. Fluidization
```

```
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-14, Example 5, Page 353
//Title: Elutriation and Attrition of Catalyst
//
   ==========================================================

clear
clc

//INPUT
dpi=[0.17;0.15;0.13;0.11;0.09;0.07;0.05;0.03;0.01];
   //Mean size of particles in mm
a=[0;0.95;2.45;5.2;10.1;23.2;35.65;20.0;2.45]*10^-2;
   //Feed composition Fo(dpi)/Fo
y=[0;0;0;0;0;0;0.625;10.225;159.25]*10^-6;//
   Elutriation and cyclone efficiency k(dpi)(1-eta(
   dpi))
F=0.01;//Rate at which solids are withdrawn in kg/s
W=40000;//Weight of bed in kg
dp1=0.11//Initial size in mm
dp2=0.085;//Size after shrinking in mm
dpmin=0.01;//Minimum size in mm
deldpi=2*10^-2;//Size inerval in mm
t=20.8;//Time in days
si=1;

//CALCULATION
kdash=log((dp1-dpmin)/(dp2-dpmin))/(t*24*3600);//
   Rate of particle shrinkage from Eqn.(24)
n=length(dpi);
m=2;
Fo=0.05;//Initial value of Fo
F1(m-1)=0;
s=0;
c=0;
```

```
t=1E-6;
while m<=n
    R(m)=-kdash*(dpi(m)-dpmin);//Rate of size change
    x(m)=(a(m)*Fo-W*R(m-1)*F1(m-1)/deldpi)/(F+(W*y(m
       ))-(W*R(m)/deldpi)-3*W*R(m)/dpi(m));//Eqn
       .(34)
    F1(m)=x(m)*F;
    c=c+x(m);
    m=m+1;
    if abs(c-1)<t then break
    end
    Fo=Fo+0.0001;//Incrementing Fo
end

//OUTPUT
mprintf('\nFeed rate with deldpi=%fmm is %fg/hr',
   deldpi,Fo);
i=1;
mprintf('\nBed composition');
while i<=n
    printf('\n%f',x(i)*100);
    i=i+1;
end

//=================================================END OF PROGRAM
```

Chapter 15

Circulation Systems

Scilab code Exa 15.1 Circulation Rate when Deactivation Controls

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-15, Example 1, Page 369
//Title: Circulation Rate when Deactivation Controls
//==========================================================================================================

clear
clc

//INPUT
thalf=1;//Half life of catalyst in s
F=960;//Feed rate of oil in tons/day
W=50;//Weight of the bed in tons
a=0.5;//Activity after time equal to half life
abar=0.01;//Average activity of the catalyst

//CALCULATION
```

```
Ka=-log(a)/thalf;//Rate constant is s^-1, assuming I
    order kinetics from Eqn.(12)
Fs=Ka*W*abar/(1-abar);//Circulation rate of solids
   from Eqn.(16)
x=(Fs*60*60*24)/F;//Circulation rate per feed of oil

//OUTPUT
mprintf('\nSolid recirculation per feed of oil =
   %ftons of solid circulated/ton feed oil',x);

//================================END OF PROGRAM
   ==================================================
```

Scilab code Exa 15.2 Circulation Rate when Heat Duty Controls

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-15, Example 2, Page 370
//Title: Circulation Rate when Heat Duty Controls
//
   ==========================================================

clear
clc

//INPUT
deltaHr1=1260;//Enthalpy change during endothermic
   reaction in kJ/kg
deltaHr2=-33900;//Enthal[y change during exothermic
   reaction in kJ/kg
```

```
H1=703;//Enthalpy of feed oil in kJ/kg
T1=260;//Temperature of feed oil in degree celcius
H3=1419;//Enthalpy of cracked product in kJ/kg
T3=500;//Temperature of cracked product in degree
    celcius
Ta=20;//Temperature of entering air in degree
    celcius
Cpa=1.09;//Specific heat of entering air in kJ/kg K
Cpf=1.05;//Specific heat of flue gases in kJ/kg K
Cps=1.01;//Specific heat of solids in kJ/kg K
Cpv=3.01;//Specific heat of vaporized feed in kJ/kg
    K
T4=[520;540;560;580;600;620;640;660];//Temperature
    of flue gas in degree celcius
V=22.4;//Volume of 1 mole of Carbon dioxide gas in N
    -m^3
M=12;//Molecular weight of carbon in kg
rho=1.293;//Density of carbon dioxide gas in kg/N-m
    ^3
xa=0.21;//Mass fraction of oxygen in air
betac=0.07;//Mass fraction of carbon

//CALCULATION
n=length(T4);
i=1;

x2min=betac*(V*rho/(M*xa));//Minimum amount of air
    required for complete combustion
while i<=n
    x1(i)=(deltaHr1+0.93*H3-H1)/(Cps*(T4(i)-T3));//
        Fs/F1 by simplifying the overall energy
        balance
    x2(i)=[(0.07*(-deltaHr2)-(deltaHr1+0.93*H3-H1))
        /(Cpf*(T4(i)-Ta))]-0.07;//F2/F1 by
        simplifying the energy balance for
        regenerator
    if x2(i)>x2min then excess_air(i)=(x2(i)-x2min)/
        x2min; //Excess air used
```

```
    else excess_air(i)=0;
    end
    i=i+1;
end

//OUTPUT
printf('\nT4(degree celcius)');
printf('\tFs/F1');
printf('\t\tF2/F1');
printf('\t\tExcess air(percentage)');
i=1;
while i<=n
    mprintf('\n%f',T4(i));
    mprintf('\t\t%f',x1(i));
    mprintf('\t%f',x2(i));
    mprintf('\t%f',excess_air(i)*100);
    i=i+1;
end

//Disclaimer: The values of F2/F1 obtained by manual calculation has close correspondance to the ones obtained as the output, whereas it deviates largely from the values given in textbook.

//==========================================END OF PROGRAM
==============================================================
```

Scilab code Exa 15.3 Aeration of Fine Particle Downcomer

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

```

```
//Chapter-15, Example 3, Page 379
//Title: Aeration of Fine Particle Downcomer
//

clear
clc

//INPUT
Fs=100;//Solid flowrate in kg/s
ephsilon1=0.55;
ephsilon2=0.5;
p1=120;//Pressure at upper level in kPa
rhos=1000;//Density of solid in kg/m^3
rhog=1;//Density of gas in kg/m^3
gc=1;//Conversion factor
g=9.81;//Acceleration due to gravity in m/s^2
di=0.34;//Diameter of downcomer in m
pi=3.14;

//CALCULATION
x=(ephsilon1/ephsilon2)*((1-ephsilon2)/(1-ephsilon1));//To find pressure at lower level using Eqn.(30)
p2=x*p1;//Pressure at lower level using Eqn.(30)
deltap=p2-p1;
ephsilonbar=0.5*(ephsilon1+ephsilon2);
deltah=(deltap*10^3*gc)/(rhos*(1-ephsilonbar)*g);//Static head height from Eqn.(28)
At=0.25*pi*di^2;//Area of downcomer
Gs=Fs/At;//Flux of solids in downcomer
Gg=Gs*(ephsilon1/(1-ephsilon1))*(rhog/rhos)*(x-1);//Required gas aeration rate from Eqn.(31)
Fg=Gg*At;//Flow rate of gas required

//OUTPUT
mprintf('\nThe required flow rate of gas required
```

```
    for location of %fm below downcomer is %fkg/s',
    deltah,Fg);

//========================================================END OF PROGRAM
    ==================================================================
```

Scilab code Exa 15.4 Circulation in Side by Side Beds

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-15, Example 4, Page 380
//Title: Circulation in Side-by-Side Beds
//
    ==========================================================================

clear
clc

//INPUT
Fs=600;//Solid circulation rate in kg/s
dpbar=60;//Mean size of solids in micrometer
pA=120;//Pressure in vessel A in kPa
pB=180;//Pressure in vessel B in kPa
LfA=8;//Bed height in vessel A in m
LfB=8;//Bed height in vessel B i m
//Bulk densities in kg/m^3
rho12=100;
rho34=400;
rho45=550;
rho67=200;
```

```
rho78=200;
rho910=400;
rho1011=400;
rho1112=550;
rho13=100;
deltapdA=7;//Pressure drop across the distributor in regenerator in kPa
deltapdB=8;//Pressure drop across the distributor in reactor in kPa
deltap12=(9+4);//Friction loss and pressure difference required to accelerate the solids in transfer lines in kPa
deltap78=(15+3);//Friction loss and pressure difference required to accelerate the solids in transfer lines in kPa
deltap45=20;//Friction loss across the reactor's stripper downcomer in kPa
deltap1112=4;//Friction loss across the regenerator's downcomer in kPa
deltapvA=5;//Pressure drop assigned for the control valve in regenerator in kPa
deltapvB=15;//Pressure drop assigned for the control valve in reactor in kPa
deltah12=15;//Height of the riser in m
deltah86=30;//Height of the riser in m
deltah1011=7;//Height difference h10-h11 in m
g=9.81;//Acceleration due to gravity in m/s^2
gc=1;//Conversion factor
pi=3.14;

//CALCULATION
Gs=900;//From Fig.(8), to find dt
dt=sqrt((4/pi)*Fs/Gs);//Diameter of the downcomer
//Height of downcomer A from Eqn.(7)
deltahA=(1/(rho1112*g))*[(pB-pA)*gc*(10^3)+(deltap12+deltapdB+deltap1112+deltapvA)*gc*10^3-rho12*g*(-deltah12)-rho34*g*(-LfB)-rho1011*g*deltah1011];
//Height of downcomer B from Eqn.(8)
```

```
deltahB=(1/(rho45*g))*[-(pB-pA)*gc*10^3+(deltap45+
   deltapvB+deltap78+deltapdA)*gc*10^3+rho78*g*
   deltah86+rho910*g*LfA];

//OUTPUT
printf('\nHeight of downcomer for:');
mprintf('\n\tRegenerator:%fm',deltahA);
mprintf('\n\tReactor:%fm',deltahB);

//==========================================END OF PROGRAM
   ==========================================================
```

Scilab code Exa 15.5 Steam Seal of a Coarse Particle Downcomer

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-15, Example 5, Page 381
//Title: Steam Seal of a Coarse Particle Downcomer
//
   ==========================================================================


clear
clc

//INPUT
dp=10^-3;//Particle diameter in m
dt=0.8;//Diameter of reactor in m
us=0.15;//Descend velocityo of solids in m/s
L=15;//Length of downcomer
deltap1=300;//Pressure in lower vessel in kPa
```

```
deltap2=240;//Pressure in upper vessel in kPa
phis=0.8;//Sphericity of solids
ephsilonm=0.45;//Void fraction of bed
myu=4E-5;//Viscosity of gas in kg/m s
rhogl=2;//Density of gas in lower vessel in kg/m^3
rhogu=1.6;//Density of gas in upper vessel in kg/m^3
rhogbar=0.5*(rhogl+rhogu);//Average density in kg/m^3
gc=1;//Conversion factor

//CALCULATION
//(a)Without steam seal
deltapfr=(deltap1-deltap2)*10^3;//Frictional pressure drop between two levels in Pa
deluguess=50;//Guess value of deltau
function[fn]=solver_func(delu)//Function defined for solving the system
    fn=(deltapfr*gc/L)-(150*(1-ephsilonm)^2*myu*delu/(ephsilonm^2*(phis*dp)^2))-(1.75*(1-ephsilonm)*rhogbar*delu^2/(ephsilonm*phis*dp));
endfunction
[delu]=fsolve(deluguess,solver_func,1E-6);//Using inbuilt function fsolve for solving Eqn.(25) for deltau
uo=(delu-us)*ephsilonm;//Superficial gas velocity
Fg=rhogbar*uo*(pi/4)*dt^2;//Flow rate of gs up the tube

//(c)With steam seal
//For section 1 to 3
L1=10;
deluguess1=50;//Guess value of deltau
function[fn]=solver_func1(delu1)//Function defined for solving the system
    fn=(deltapfr*gc/L1)-(150*(1-ephsilonm)^2*myu*delu1/(ephsilonm^2*(phis*dp)^2))-(1.75*(1-ephsilonm)*rhogbar*delu1^2/(ephsilonm*phis*dp
```

```
    ));
endfunction
[delu1]=fsolve(deluguess1,solver_func1,1E-6);//Using inbuilt function fsolve for solving Eqn.(25) for deltau
uou=(delu1-us)*ephsilonm;//Upward superficial gas velocity
Fgu=rhogbar*uou*(pi/4)*dt^2;//Upward flow rate of gs up the tube
//For section 3 to 2
ugd=0.15;//Downward velocity of gas
uod=ugd*ephsilonm;//Downward superficial gas velocity
Fgd=rhogbar*uod*(pi/4)*dt^2;//Downward flow rate of gas up the tube
Fgt=Fgu+Fgd;//Total flow rate of gas

//OUTPUT
printf('\nWithout steam seal');
printf('\n\tFlow rate of gas up the tube:%fkg/s',Fg);
printf('\nWith steam seal');
printf('\n\tTotal flow rate of gas:%fkg/s',Fgt);

//========================================END OF PROGRAM
```

Chapter 16

Design for Physical Operations

Scilab code Exa 16.1 Single Stage Limestone Calciner

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-16, Example 1, Page 404
//Title: Single-Stage Limestone Calciner
//==========================================================================

clear
clc

//INPUT
T=1000;//Operating temperature of calciner in degree celcius
deltaHr=1795;//Heat of reaction in kJ/kg
M1=0.1;//Molecular weight of Calcium carbonate in kg/mol
M2=0.056;//Molecular weight of CaO in kg/mol
M3=0.044;//Molecular weight of Carbon dioxide  in kg
```

```
/mol
M4=0.029;//Molecular weight of Air in kg/mol
M5=0.029;//Molecular weight of Combustion gas in kg/mol
Cp1=1.13;//Specific heat of Calcium carbonate in kJ/kg K
Cp2=0.88;//Specific heat of CaO in kJ/kg K
Cp3=1.13;//Specific heat of Carbon dioxide in kJ/kg K
Cp4=1.00;//Specific heat of Air in kJ/kg K
Cp5=1.13;//Specific heat of Calcium carbonate in kJ/kg K
Tf=20;//Temperature of feed in degree celcius
ma=15;//Air required per kg of fuel in kg
Hc=41800;//Net combustion heat of fuel in kJ/kg
Tpi=20;//Initial temperature of solids in degree C
Tgi=1000;//Initial temperature of gas in degree C

//CALCULATION
mc=1;//Based on 1 kg of Calcium carbonate
B=(1/(Hc-(ma+mc)*Cp5*(T-Tpi)))*[M3*Cp3*(T-Tf)+M2*Cp2*(T-Tf)+deltaHr]//Fuel consumption(kg fuel/kg calcium carbonate)
B1=B*M3/M2;//Fuel consumption(kg fuel/kg Cao)
H=Hc*B1;//Heat required for calcination
eta=deltaHr/(B*Hc);//Thermal efficiency

//OUTPUT
mprintf('\nFuel consumption:%f kg fuel/kg Cao',B1);
mprintf('\nHeat requirement for calcination:%f kJ/kg Cao',H);
mprintf('\nThermal efficiency:%f percentage',eta*100);

//==================================END OF PROGRAM
```

Scilab code Exa 16.2 Multistage Limestone Calciner

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-16, Example 2, Page 405
//Title: Multistage Limestone Calciner
//

clear
clc

//INPUT
F=400;//Feed rate of Calcium carbonate in tons/day
T=1000;//Operating temperature of calciner in degree celcius
deltaHr=1795;//Heat of reaction in kJ/kg
M1=0.1;//Molecular weight of Calcium carbonate in kg/mol
M2=0.056;//Molecular weight of CaO in kg/mol
M3=0.044;//Molecular weight of Carbon dioxide  in kg/mol
M4=0.029;//Molecular weight of Air in kg/mol
M5=0.029;//Molecular weight of Combustion gas in kg/mol
Cp1=1.13;//Specific heat of Calcium carbonate in kJ/kg K
Cp2=0.88;//Specific heat of CaO in kJ/kg K
Cp3=1.13;//Specific heat of Carbon dioxide in kJ/kg K
Cp4=1.00;//Specific heat of Air in kJ/kg K
```

```
Cp5=1.17;//Specific heat of Combustion gas in kJ/kg K
Tf=20;//Temperature of feed in degree celcius
ma=15;//Air required per kg of fuel in kg
uo=0.8;//Superficial gas velocity in m/s
Hc=41800;//Net combustion heat of fuel in kJ/kg
Tpi=20;//Initial temperature of solids in degree C
Tgi=1000;//Initial temperature of gas in degree C
rhoa=1.293;//Density of air in kg/m^3
pi=3.14;

//CALCULATION
mc=1;//Based on 1 kg of Calcium carbonate
Bguess=2;//Guess value of B
function[fn]=solver_func(B)//Function defined for solving the system
    phi=((ma+mc)*Cp5*B+(M3*Cp3))/Cp1;
    T3=(Tpi+(phi+phi^2+phi^3)*Tgi)/(1+phi+phi^2+phi^3);
    phiplus=30.6*B
    Tr=(T+Tpi*phiplus)/(1+phiplus);
    fn=Hc*B+Cp3*(T3-Tpi)+ma*B*Cp4*(Tr-20)-(ma+mc)*Cp5*(T-Tpi)-M3*Cp3*(T-Tpi)-M2*Cp2*(T-Tpi)-deltaHr;
    //fn=(1/20800)*(2470-T3-13.34*(Tr-20));
endfunction
[B]=fsolve(Bguess,solver_func,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
phi=((ma+mc)*Cp5*B+(M3*Cp3))/Cp1;
//Temperature of various stages
T1=(Tpi+(phi)*Tgi)/(1+phi);
T2=(Tpi+(phi+phi^2)*Tgi)/(1+phi+phi^2);
T3=(Tpi+(phi+phi^2+phi^3)*Tgi)/(1+phi+phi^2+phi^3);
phiplus=30.6*B
Tr=(T+Tpi*phiplus)/(1+phiplus);
eta=deltaHr/(B*Hc);//Thermal efficiency
H=B*Hc/M2;//Heat requirement
//For lower heat recovery section
```

```
Ql=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(Tr+273)));//Volumetric flow rate of gas in the lower heat recovery section
dtl=sqrt(4/pi*Ql/uo);//Diameter of lower bed
//For calcination section
Qc=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T+273)));//Volumetric flow rate of gas in the calcination section
dtc=sqrt(4/pi*Qc/uo);//Diameter of calcination section
//For I stage
Q1=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T1+273)));//Volumetric flow rate of gas in the I stage
dt1=sqrt(4/pi*Q1/uo);//Diameter of I stage
//For II stage
Q2=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T2+273)));//Volumetric flow rate of gas in the II stage
dt2=sqrt(4/pi*Q2/uo);//Diameter of II stage
//For III stage
Q3=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T3+273)));//Volumetric flow rate of gas in the III stage
dt3=sqrt(4/pi*Q3/uo);//Diameter of III stage

//OUTPUT
printf('\nDiameter of lower bed:%fm',dtl);
printf('\nDiameter of calcination section:%fm',dtc);
printf('\nBed no.\t\t1\t2\t\t3');
printf('\nDiameter(m)%f\t%f\t%f',dt1,dt2,dt3);

//The value of diameter of each section is largely deviating from the values in the textbook. This is because the fuel consumption B have not been included in the energy balance equation. And the value of molecular weight is wrong by one decimal point.

//==================================================END OF PROGRAM
```

Scilab code Exa 16.3 Multistage Adsorber

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-16, Example 3, Page 413
//Title: Multistage Adsorber
//

clear
clc

//INPUT
T=20;//Temeprature in degree C
M=0.018;//Molecular weight of water in kg/mol
Q=10;//Flow rate of dry air in m^3/s
R=82.06E-6;//Universal gas constant
pi=0.0001;//Initial moisture content in atm
pj=0.01;//Final moisture content in atm

//CALCULATION
a=Q*(273+T)/273;//Term At*uo
b=a*M/(R*(T+273));//Term C*At*uo
//The value of slope can be found only by graphical
   mehtod. Hence it has been taken directly from the
    book(Page no.414,Fig.E3)
m=10.2;
Fo=b/m;//Flow rate of solids
Q3=(b/Fo)*(pj-pi);//Moisture content of leaving
```

```
    solids

//OUTPUT
printf('\nMoisture content of leaving solids:%f kg
    H2O/kg dry solids',Q3);

//==========================================END OF PROGRAM
    ================================================================
```

Scilab code Exa 16.4 Dryer Kinetics and Scale up

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-16, Example 4, Page 422
//Title: Dryer Kinetics and Scale-up
//
    ==========================================================================

clear
clc

//INPUT
Qfi=0.20;//Initial moisture fraction
Qfbar=0.04;//Average final moisture fraction
rhos=2000;//Density of solid in kg/m^3
Cps=0.84;//Specific heat of solids in kJ/kg K
Fo=7.6E-4;//Flow rate of solids in kg/m^3
Tsi=20;//Inital temperature of solids in degree C
rhog=1;//Density of gas in kg/m^3
Cpg=1;//Specific heat of gas in kJ/kg K
```

```
uo=0.3;//Superficial gas velocity in m/s
Tgi=200;//Initial temperature of gas in degee C
L=2370;//Enthalpy of liquid in kJ/kg
Cpl=4.2;//Specific heat of liquid in kJ/kg K
dt=0.1;//Diameter of reactor in m
Lm=0.1;//Length of fixed bed in m
ephsilonm=0.45;//Void fraction of fixed bed
pi=3.14;
Fo1=1;//Feed rate for commercial-scale reactor in kg/s

//CALCULATION
//(a)Bed temperature
Teguess=50;//Guess value of Te
function[fn]=solver_func(Te)//Function defined for solving the system
    fn=(pi/4)*dt^2*uo*rhog*Cpg*(Tgi-Te)-Fo*(Qfi-Qfbar)*[L+Cpl*(Te-Tsi)]-Fo*Cps*(Te-Tsi);
endfunction
[Te]=fsolve(Teguess,solver_func,1E-6);//Using inbuilt function fsolve for solving Eqn.(53) for Te

//(b)Drying time for a particle
xguess=2;//Guess value of x, ie term tou/tbar
function[fn]=solver_func1(x)//Function defined for solving the system
    fn=1-(Qfbar/Qfi)-(1-exp(-x))/x;
endfunction
[x]=fsolve(xguess,solver_func1,1E-6);//Using inbuilt function fsolve for solving Eqn.(61) for x
W=(pi/4)*dt^2*Lm*(1-ephsilonm)*rhos;//Weight of soilds in bed
tbar=W/Fo;//Mean residence time of solids from Eqn.(59)
tou=tbar*x;//Time for complete drying of a particle

//(c)Commercial-scale dryer
```

```
W1=Fo1*tbar;
Atguess=5;//Guess value of area
function[fn]=solver_func3(At)//Function defined for solving the system
    fn=At*uo*rhog*Cpg*(Tgi-Te)-Fo1*(Qfi-Qfbar)*[L+Cpl*(Te-Tsi)]-Fo1*Cps*(Te-Tsi);
endfunction
[At]=fsolve(Atguess,solver_func3,1E-6);//Using inbuilt function fsolve for solving Eqn.(53) for At
dt1=sqrt(4/pi*At);//Diameter of commercial-scale dryer
Q1=At*uo*rhog;//Flow rate necessary for the operation

//OUTPUT
printf('\nBed temperature:%f degree C',Te);
printf('\nTime for complete drying of particle:%fs',tou);
printf('\nFlow rate of gas necessary for Commercial-scale dryer:%fkg/s',Q1);

//========================================END OF PROGRAM========================================
```

Scilab code Exa 16.5 Solvent Recovery from Polymer Particles

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-16, Example 5, Page 425
//Title: Solvent Recovery from Polymer Particles
```

```
//
    ========================================================================

clear
clc

//INPUT
rhos=1600;//Density of solid in kg/m^3
Cps=1.25;//Specific heat of solids in kJ/kg K
Fo=0.5;//Flow rate of solids in kg/s
Tsi=20;//Inital temperature of solids in degree C
Qwi=1;//Initial moisture fraction in water
Qwf=0.2;//Final moisture fraction in water
Qhi=1.1;//Initial moisture fraction in heptane
Qhf=0.1;//Final moisture fraction in heptane
Tgi=240;//Initial temperature of gas in degee C
Te=110;//Bed temperature in degree C
ephsilonm=0.45;//Void fraction of fixed bed
ephsilonf=0.75;//Void fraction of fluidized bed
uo=0.6;//Superficial gas velocity in m/s
di=0.08;//Diameter of tubes in m
li=0.2;//Pitch for square arrangement
hw=400;//Heat transfer coefficient in W/m^2 K
Tc=238;//Temperature at which steam condenses in
    degree C
//Specific heats in kJ/kg K
Cwl=4.18;//Water liquid
Cwv=1.92;//Water vapor
Chl=2.05;//Heptane liquid
Chv=1.67;//Heptane vapor
//Latent heat of vaporization in kJ/kg
Lw=2260;//Water
Lh=326;//Heptane
//Density of vapor in kg/m^3 at operating conditions
rhow=0.56;//Water
rhoh=3.1;//Heptane
Lf=1.5;//Length of fixed bed in m
```

```
t=140;//Half-life of heptane in s
L=1.5;//Length of tubes in heat exchanger
pi=3.14;

//CALCULATION
//(a) Dryer without Internals
xw=(Qwi-Qwf)/(Qhi-Qhf);//Water-heptane weight ratio
xv=((Qwi-Qwf)/18)/((Qhi-Qhf)/100);//Water-heptane
   volume ratio
T=(Qwi-Qwf)/18+(Qhi-Qhf)/100;//Total volume
rhogbar=((Qwi-Qwf)/18)/T*rhow+((Qhi-Qhf)/100)/T*rhoh
   ;//Mean density of the vapor mixture
Cpgbar=(((Qwi-Qwf)/18)/T)*rhow*Cwv+(((Qhi-Qhf)/100)/
   T)*rhoh*Cwv;//Mean specific heat of vapor mixture
//Volumetric flow of recycle gas to the dryer in m
   ^3/s from Eqn.(53)
x=(Cpgbar*(Tgi-Te))^-1*[Fo*(Qwi-Qwf)*[Lw+Cwl*(Te-Tsi
   )]+Fo*(Qhi-Qhf)*[Lh+Chl*(Te-Tsi)]+Fo*(Cps*(Te-Tsi
   ))];
r=Fo*[(Qwi-Qwf)/rhow+(Qhi-Qhf)/rhoh};//Rate of
   formation of vapor in bed
uo1=uo*(x/(x+r));//Superficial velocity just above
   the distributor
At=x/uo1;//Cross-sectional area of bed
dt=sqrt(4/pi*At);//Diameter of bed
B=-log(Qwf/Qwi)/t;//Bed height from Eqn.(63)
tbar=((Qhi/Qhf)-1)/B;//Mean residence time of solids
W=Fo*tbar;//Weight of bed
Lm=W/(At*(1-ephsilonm)*rhos);//Static bed height
Lf=(Lm*(1-ephsilonm))/(1-ephsilonf);//Height of
   fluidized bed

//(b) Dryer with internal heaters
f=1/8;//Flow rate is 1/8th the flow rate of
   recirculation gas as in part (a)
x1=f*x;//Volumetric flow of recycle gas to the dryer
    in m^3/s from Eqn.(53)
uo2=uo*(x1/(x1+r));//Superficial velocity just above
```

```
the distributor
Abed=x1/uo2;//Cross-sectional area of bed
q=[Fo*(Qwi-Qwf)*[Lw+Cwl*(Te-Tsi)]+Fo*(Qhi-Qhf)*[Lh+Chl*(Te-Tsi)]+Fo*(Cps*(Te-Tsi))]-Abed*uo2*Cpgbar*(Tgi-Te);//Heat to be added from energy balance of Eqn.(53)
Aw=q*10^3/(hw*(Tc-Te));//Total surface area of heat exchanger tubes
Lt=Aw/(pi*di);//Total length of tubes
Nt=Lt/L;//Total number of tubes
Atubes=Nt*(pi/4*di^2);//Total cross-sectional area of tubes
Atotal=Abed+Atubes;//Total cross-sectional area of tube filled dryer
d=sqrt(Atotal*pi/4);//Diameter of vessel
li=sqrt(Atotal/Nt);//Pitch for square array of tubes

//OUTPUT
printf('\n\t\t\tBed diameter(m)\tRecycle vapor flow(m^3/s)');
printf('\nWithout internal heater\t%f\t%f',dt,x);
printf('\nWith heating tubes\t%f\t%f',d,x1);

//==========================================END OF PROGRAM==========================================
```

Chapter 17

Design of Catalytic Reactors

Scilab code Exa 17.1 Reactor Development Program

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-17, Example 1, Page 434
//Title: Reactor Development Program
//
   ==========================================================

clear
clc

//INPUT
dt=[0.081;0.205;3.6];//Reactor diameter for the
   three reactors in m
dte=[0.04;0.12;0.70];//Equivalent diameters for the
   three reactors in m
db=[0.05;0.057;0.07];//Estimated bubble size in the
   three reactors in m
Kr1=1.3889;//Kinetic constant for Reaction 1 in s^-1
```

```
Kr2=0.6111;//Kinetic constant for Reaction 2 in s^-1
Kr3=0.022;//Kinetic constant for Reaction 3 in s^-1
dp=60;//Particle size in micrometer
ephsilonm=0.50;//Void fraction of fixed bed
ephsilonmf=0.55;//Void fraction at minimum fluidized condition
umf=0.006;////Velocity at minimum fluidization condition in m/s
D=2E-5;//Diffusion coefficient of gas in m^2/s
gammab=0.005;//Ratio of volume of dispersed solids to that of bubble phase
uo=0.2;//Superficial gas velocity in m/s
XA=0.9;//Conversion
g=9.81;//Acceleration due to gravity in square m/s^2

//CALCULATION
Kr12=Kr1+Kr2;
n=length(dt);
i=1;
while i<=n
    //Preliminary Calcualtions
    ubr(i)=0.711*(g*db(i))^0.5;//Rise velocity of bubble from Eqn.(6.7)
    ub(i)=1.55*{(uo-umf)+14.1*(db(i)+0.005)}*dte(i)^0.32+ubr(i);//Bubble velocity for Geldart A particles from Equation from Eqn.(6.11)
    delta(i)=uo/ub(i);//Fraction of bed in bubbles from Eqn.(6.29)
    ephsilonf(i)=1-(1-delta(i))*(1-ephsilonmf);//Void fraction of fixed bed from Eqn.(6.20)
    fw=0.6;//Wake volume to bubble volume from Fig.(5.8)
    gammac(i)=(1-ephsilonmf)*((3/(ubr(i)*ephsilonmf/umf-1))+fw);//Volume of solids in cloud to that of the bubble from Eqn.(6.36)
    gammae(i)=((1-ephsilonmf)*((1-delta(i))/delta(i)))-gammab-gammac(i);//Volume of solids in emulsion to that of the bubble from Eqn
```

```
.(6.35)
Kbc(i)=4.5*(umf/db(i))+5.85*((D^0.5*g^0.25)/db(i)^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
Kce(i)=6.77*((D*ephsilonmf*0.711*(g*db(i))^0.5)/db(i)^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)
//Effective rate constant from Eqn.(12.32)
Kf12(i)=(gammab*Kr12+1/((1/Kbc(i))+(1/(gammac(i)*Kr12+1/((1/Kce(i))+(1/(gammae(i)*Kr12)))))))*(delta(i)/(1-ephsilonf(i)));
//Rate of reaction 2 for fluidized bed from Eqn.(12.14)
Kf3(i)=(gammab*Kr3+1/((1/Kbc(i))+(1/(gammac(i)*Kr3+1/((1/Kce(i))+(1/(gammae(i)*Kr3)))))))*(delta(i)/(1-ephsilonf(i)));
//Rate of raection with respect to A from Eqn.(12.35)
KfA(i)=[[Kbc(i)*Kce(i)/gammac(i)^2+(Kr12+Kce(i)/gammac(i)+Kce(i)/gammae(i))*(Kr3+Kce(i)/gammac(i)+Kce(i)/gammae(i))]*delta(i)*Kbc(i)*Kr12*Kr3/(1-ephsilonf(i))]    /[[(Kr12+Kbc(i)/gammac(i))*(Kr12+Kce(i)/gammae(i))+Kr12*Kce(i)/gammac(i)]*[(Kr3+Kbc(i)/gammac(i))*(Kr3+Kce(i)/gammae(i))+Kr3*Kce(i)/gammac(i)]];
KfAR(i)=((Kr1/Kr12)*Kf12(i))-KfA(i);//Rate of reaction from Eqn.(12.34)
KfAR1(i)=((Kr1/Kr12)*Kf12(i));//Since KfA is small

//(b)Relate Selectivity with conversion in three reactors
x=-log(1-XA);//The term Kf12*tou in Eqn.(12.26)
tou(i)=x/Kf12(i);//Residence time from Eqn.(12.26)
y(i)=(KfAR1(i)/(Kf3(i)-Kf12(i)))*(exp(-x)-exp(-tou(i)*Kf3(i)));//CR/CAi from Eqn.(12.27)
SR(i)=y(i)/XA;//Selectivity of R
```

```

    //(c)Relate exit composition to space time
    tou1=5;//Space time in s
    XA1(i)=1-exp(-Kf12(i)*tou1);//Conversion from Eqn.(12.26)
    y1(i)=((KfAR1(i)/(Kf12(i)-Kf3(i)))*[exp(-Kf3(i)*tou1)-exp(-Kf12(i)*tou1)]);//CR/CAi R from Eqn.(12.27)

    //(d)Calculate height of bed needed to maximize production
    y2(i)=(KfAR1(i)/Kf12(i))*(Kf12(i)/Kf3(i))^(Kf3(i)/(Kf3(i)-Kf12(i)));//CRmax/CAi R from Eqn.(12.37)
    tou2(i)=log(Kf3(i)/Kf12(i))/(Kf3(i)-Kf12(i));//Space time from Eqn.(38)
    Lf(i)=(uo/(1-ephsilonf(i)))*tou2(i);//Length of bed at fully fluidized condition from Eqn.(12.5)
    Lm(i)=Lf(i)*(1-ephsilonf(i))/(1-ephsilonm);//Length of bed when settled
    XA2(i)=1-exp(-Kf12(i)*tou2(i));//Conversion from Eqn.(12.26)
    i=i+1;
end

//OUTPUT
printf('\nLet Laboratory, Pilot plant, Semicommercial unit be Reactor 1,2 & 3 respectively');
printf('\n(a)Relation between effective rate constant(Kf12) to the gas flow rate(uo)');
printf('\n\tReactor No.\tKf12(s^-1)\tuo(m/s)');
i=1;
while i<=n
     mprintf('\n\t%1.0f',i);
     mprintf('\t\t%f',Kf12(i));
     mprintf('\t%f',uo);
```

```
    i=i+1;
end
printf('\n(b)Relation between selectivity with conversion');
printf('\n\tReactor No.\tKf12(s^-1)\tSR(mol R formed/mol A reacted)');
i=1;
while i<=n
    mprintf('\n\t%1.0f',i);
    mprintf('\t\t%f',Kf12(i));
    mprintf('\t%f',SR(i));
    i=i+1;
end
printf('\n(c)Relation between exit compostion and space time');
printf('\n\tReactor No.\tXA\t\tCR/CAi');
i=1;
while i<=n
    mprintf('\n\t%1.0f',i);
    mprintf('\t\t%f',XA1(i));
    mprintf('\t%f',y1(i));
    i=i+1;
end
printf('\n(d)Height of bed needed to maximize the production of acrylonitrile');
printf('\n\tReactor No.\tLm(m)\t\tXA');
i=1;
while i<=n
    mprintf('\n\t%1.0f',i);
    mprintf('\t\t%f',Lm(i));
    mprintf('\t%f',XA2(i));
    i=i+1;
end

//=====================================END OF PROGRAM
```

Scilab code Exa 17.2 Design of a Commercial Acrylonitrile Reactor

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-17, Example 2, Page 438
//Title: Design of a Commercial Acrylonitrile
   Reactor
//
   ==========================================================================


clear
clc

//INPUT
deltaHr=5.15E8;//Heat of reaction in J/k mol
W=5E4;//Weight of acrylonitirle produced per 334-day
    year in tonnes
db=0.07;//Estimated bubble size in m
dte=0.7;//Equivalent diameter in m
Kf12=0.35;//Effective rate constant in s^-1 from
   Example 1
dp=60;//Particle size in micrometer
ephsilonm=0.50;//Void fraction of fixed bed
ephsilonmf=0.55;//Void fraction at minimum fluidized
    condition
T=460;//Temperature in reactor in degree C
Pr=2.5;//Pressure inside reactor in bar
//Feed gas composition
x1=1;//Propylene
x2=1.1;//Ammonia
x3=11;//Air
```

```
do1=0.08;//OD of heat exchanger tubes in m\
L=7;//Length of tubes in m
ho=300;//Outside heat transfer coefficient in W/m^2 K
hi=1800;//Inside heat transfer coefficient in W/m^2 K
Tc=253.4;//Temperature of coolant in degree C
pi=3.14;

//CALCULATION
//Preliminary calculation
uo=0.46;//Superficial gas velocity from Fig.E1(a) for the value of Kf12 & db
tou=8;//Space time from Fig.E2(b) for highest concentraion of product R
Lm=uo*tou/(1-ephsilonm);
y=0.58;//CR/CAi from Fig.E1(c) for the value of tou & Kf12
XA=0.95//From Fig.E1(c) for the value of tou & Kf12
SR=y/XA;//Selectivity of R

//Cross-sectional area of the reactor
P=W*10^3/(334*24*3600);//Production rate of acrylonitrile
F=(P/0.053)/(SR*XA/0.042);//Feed rate of propylene
V=((F*22.4*(T+273)*(x1+x2+x3))/(42*273*Pr));
At=V/uo;//Cross-sectional area of reactor needed for the fluidized bed

//Heat exchanger calculation
q=F*XA*deltaHr/42;//Rate of heat liberation in the reactor
U=(ho^-1+hi^-1)^-1;//Overall heat transfer coefficient
deltaT=T-Tc;//Driving force for heat transfer
Aw=q/(U*deltaT);//Heat exchanger area required to remove q
Nt=Aw/(pi*do1*L);
```

```
li1=(At/Nt)^0.5;//Pitch for square pitch arrangement
dte1=4*[li1^2-(pi/4)*do1^2]/(pi*do1);
if dte1>dte then li=(pi/4*dte*do1+pi/4*do1^2)^0.5;//Pitch if we add dummy tubes
end
f=li^2-pi/4*do1^2;//Fraction of bed cross section taken up by tubes
dt1=sqrt(4/pi*At/(1-f));//Reactor diameter including all its tubes

//OUTPUT
printf('\nSuperficial gas velocity=%fm/s',uo);
printf('\nNo. of %1.0fm tubes required=%1.0f',L,Nt);
printf('\nReactor diameter=%fm',dt1);

//========================================================END OF PROGRAM=========================================================================
```

Scilab code Exa 17.3 Reactor Regenerator with Circulating Catalyst Catalytic Cracking

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-17, Example 3, Page 444
//Title: Reactor-Regenerator with Circulating Catalyst: Catalytic Cracking
//==========================================================================================================

clear
```

```
clc

//INPUT
db=0.08;//Estimated bubble size in m
dte=2;//Equivalent diameter in m
F1=55.6;//Feed rate of oil in kg/s
XA=0.63;//Conversion
uo=0.6;//Superficial gas velocity in m/s
T1=500;//Temperature of reactor in degree C
T2=580;//Temperature of regenerator in degree C
Fs=F1*23.3;//Solid circulation rate from Ex.(15.2)
rhos=1200;//Density of catalyst in kg/m^3
dpbar=60;//Average particle size in micrometer
ephsilonm=0.50;//Void fraction of fixed bed
ephsilonmf=0.55;//Void fraction at minimum fluidized
    condition
umf=0.006;////Velocity at minimum fluidization
   condition in m/s
dt=8;//Diameter of reactor in m
D=2E-5;//Diffusion coefficient of gas in m^2/s
Kr=8.6;//Rate constant for reaction at 500 degree C
   in s^-1
Ka1=0.06;//Rate constant for deactivatiion at 500
   degree C in s^-1
Ka2=0.012;//Rate constant for regeneration at 580
   degree C in s^-1
gammab=0.005;//Ratio of volume of dispersed solids
   to that of bubble phase
g=9.81;//Acceleration due to gravity in square m/s^2
pi=3.14;

//CALCULATION
//Parameters for the fluidized reactor
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from
   Eqn.(6.7)
ub=1.55*{(uo-umf)+14.1*(db+0.005)}*dte^0.32+ubr;//
   Bubble velocity for Geldart A particles from
   Equation from Eqn.(6.11)
```

```
delta=uo/ub;//Fraction of bed in bubbles from Eqn.(6.29)
ephsilonf=1-(1-delta)*(1-ephsilonmf);//Void fraction of fixed bed from Eqn.(6.20)
fw=0.6;//Wake volume to bubble volume from Fig.(5.8)
gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw);//Volume of solids in cloud to that of the bubble from Eqn.(6.36)
gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-gammac;//Volume of solids in emulsion to that of the bubble from Eqn.(6.35)
Kbc=4.5*(umf/db)+5.85*((D^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
Kce=6.77*((D*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)

//Bed height versus catalyst activity in reactor
a1bar=0.07;//Guess value for average activity in reactor
x=Kr*a1bar;//Value of Kra1 to be used in the following equation
Kf=(gammab*x+1/((1/Kbc)+(1/(gammac*x+1/((1/Kce)+(1/(gammae*x)))))))*(delta/(1-ephsilonf));//Effective rate constant from Eqn.(12.14)
tou=-log(1-XA)/Kf;//Space time from Eqn.(12.16)
Lm=tou*uo/(1-ephsilonm);//Length of fixed bed for guess value of a1bar
a1bar1=[0.0233;0.0465;0.0698;0.0930;0.116;0.140];//Various activity values to find Lm
n=length(a1bar1);
i=1;
while i<=n
    x1(i)=Kr*a1bar1(i);
    Kf1(i)=(gammab*x1(i)+1/((1/Kbc)+(1/(gammac*x1(i)+1/((1/Kce)+(1/(gammae*x1(i))))))))*(delta/(1-ephsilonf));//Effective rate constant
```

```
from Eqn.(12.14)
    tou1(i)=-log(1-XA)/Kf1(i);//Space time from Eqn.(12.16)
    Lm1(i)=tou1(i)*uo/(1-ephsilonm);//Length of fixed bed for guess value of a1bar...Condition (i)
    i=i+1;
end

//Find the optimum size ratio for various a1bar
Lm=[5;6;7;8;10;12];
m=length(Lm);
i=1;
while i<=m
    W1(i)=(pi/4)*dt^2*rhos*(1-ephsilonm)*Lm(i);//Bed weight
    t1bar(i)=W1(i)/Fs;//Mean residence time of solids in reactor
    t2bar(i)=t1bar(i)*(Ka1/Ka2)^0.5;//Mean residence time of soilds at optimum from Eqn.(16)
    a1bar2(i)=(Ka2*t2bar(i))/(Ka1*t1bar(i)+Ka1*t1bar(i)*Ka2*t2bar(i)+Ka2*t2bar(i));//From Eqn.(15)...Condition (ii)
    i=i+1;
end

//Final design values
Lm4=7.3;//For satisfying condition (i) & (ii)
a1bar3=0.0744;//By interpolation
x2=a1bar3*Kr;
W11=(pi/4)*dt^2*rhos*(1-ephsilonm)*Lm4;//Bed weight for reactor
t1bar1=W11/Fs;//Mean residence time of solids in reactor
a2bar=(1+Ka1*t1bar1)*a1bar3;//Average activity in regenrator from Eqn.(10)
t2bar1=t1bar1*(Ka1/Ka2)^0.5;//Mean residence time of solids in regenerator from Eqn.(16)
```

```
W2=W11*(t2bar1/t1bar1);//Bed weight for regenerator
dt2=dt*(W2/W11)^0.5;//Diameter of regenerator assuming same static bed height for reactor and regerator

//OUTPUT
printf('\nBed height versus catalyst activity in reactor');
printf('\n\tAverage activity');
printf('\tLength of fixed bed(m)');
i=1;
while i<=n
    mprintf('\n\t%f',a1bar1(i));
    mprintf('\t\t%f',Lm1(i));
    i=i+1;
end
printf('\nOptimum size ratio for various activity in reactor');
printf('\n\tLength of fixed bed(m)');
printf('\tAverage activity');
i=1;
while i<=m
    mprintf('\n\t%f',Lm(i));
    mprintf('\t\t%f',a1bar2(i));
    i=i+1;
end
printf('\nFinal design values');
printf('\n\tDiameter of reactor(m):%f',dt);
printf('\n\tBed weight for reactor(tons):%f',W11/10^3);
printf('\n\tBed weight for regenerator(tons):%f',W2/10^3);
printf('\n\tDiameter of regenerator(m):%f',dt2);
printf('\n\tSolid circulation rate(tons/hr):%f',Fs*3.6);

//========================================END OF PROGRAM
```

Chapter 18

The Design of Noncatalytic Gas Solid Reactors

Scilab code Exa 18.1 Kinetics of Zinc Blende Roasting

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-18, Example 1, Page 456
//Title: Kinetics of Zinc Blende Roasting
//
    ==========================================================


clear
clc

//INPUT
xA=0.08;//Fraction of oxygen in stream
dp=[2;0.1];//Particle diameter in mm
rhos=4130;//Density of catalyst in kg/m^3
Ds=8E-6;//Diffusion coefficient of solid in m^2/s
kc=0.02;//Reaction rate constant in m/s
```

```
P=10^5;//Pressure in bar\
R=8.314;//Universal gas constant
T=900;//Temperature in degree C
mB=0.09745;//Molecular weight of ZnS in kg/mol

//CALCULATION
b=2/3;//Stoichiometric coefficient of ZnS in the reaction equation
CA=xA*P/(R*(T+273));//Concentration of Oxygen
rhob=rhos/mB;//Molar density of pure solid
n=length(dp);
i=1;
while i<=n
    kbar(i)=(kc^-1+(dp(i)*10^-3/(12*Ds)))^-1;//Average reaction rate constant from Eqn.(11)
    tou(i)=rhob*dp(i)*10^-3/(2*b*kbar(i)*CA);//Time for complete reaction in seconds from Eqn.(9)
    i=i+1;
end

//OUTPUT
printf('\nParticle Size(mm)\tAverage rate constant(m/s)\tTime for complete reaction(min)');
i=1;
while i<=n
    mprintf('\n%f\t\t%f\t\t\t%f',dp(i),kbar(i),tou(i)/60);
    i=i+1;
end

//===================================END OF PROGRAM
```

Scilab code Exa 18.2 Kinetics of Carbon Burning

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-18, Example 2, Page 457
//Title: Kinetics of Carbon Burning
//
    ==========================================================================


clear
clc

//INPUT
xA=0.08;//Fraction of oxygen in stream
dp=1;//Particle diameter in mm
rhos=2200;//Density of catalyst in kg/m^3
kc=0.2;//Reaction rate constant in m/s
mC=0.012;//Molecular weight of carbon in kg/mol
P=10^5;//Pressure in bar\
R=8.314;//Universal gas constant
T=900;//Temperature in degree C

//CALCULATION
b=1;//Stoichiometric coefficient of C in the
    reaction equation
CA=xA*P/(R*(T+273));//Concentration of Oxygen
rhob=rhos/mC;//Molar density of pure solid reactant
tou=rhob*10^-3/(2*b*kc*CA);//Time required for
    complete reaction in seconds

//OUTPUT
mprintf('\nThe time required for complete combustion
    :%fmins',tou/60);

//========================================END OF PROGRAM
```

Scilab code Exa 18.3 Roasting Kinetics from Flowing Solids Data

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-18, Example 3, Page 462
//Title: Roasting Kinetics from Flowing Solids Data
//
   ===========================================================


clear
clc

//INPUT
dp=110;//Particle size in micrometer
T=900;//Temperature of roaster in degree C
tbar1=[3;10;30;50];//Reported average time in min
XBbarr=[0.840;0.940;0.985;0.990];//Reported value of
    average conversion
tbar=3;
XBbar=0.840;//Average conversion for tbar = 3 mins

//CALCULATION
//Uniform-Reaction Model
x=(1/tbar)*(1/(1-XBbar)-1);//Term KrCA of Eqn.(20)
n=length(tbar1);
i=1;
while i<=n
    XBbar1(i)=1-1/(1+x*tbar1(i));//Average
```

```
conversion using calculated value of KrCA from Eqn.(20)
    i=i+1;
end

//Shrinking-Core, Rection Control
touguess=2;//Guess value of tou
function[fn]=solver_func(tou)//Function defined for solving the system
    fn=(1-XBbar)-(0.25*tou/tbar)+(0.05*(tou/tbar)^2)-((1/120)*(tou/tbar)^3);
endfunction
[tou]=fsolve(touguess,solver_func,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
i=1;
while i<=n
    XBbar2(i)=1-(0.25*tou/tbar1(i))+(0.05*(tou/tbar1(i))^2)-((1/120)*(tou/tbar1(i))^3);//Average conversion using calculated value of tou from Eqn.(23)
    i=i+1;
end

//Shrinking-Core, Diffusion Control
touguess1=2;//Guess value of tou
function[fn]=solver_func1(tou)//Function defined for solving the system
    fn=(1-XBbar)-(1/5*tou/tbar)+(19/420*(tou/tbar)^2)-(41/4620*(tou/tbar)^3)+(0.00149*(tou/tbar)^4);
endfunction
[tou1]=fsolve(touguess1,solver_func1,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
i=1;
while i<=n
    //Average conversion using calculated value of
```

```
        tou from Eqn.(23)
    XBbar3(i)=1-(1/5*tou1/tbar1(i))+(19/420*(tou1/
        tbar1(i))^2)-(41/4620*(tou1/tbar1(i))^3)
        +(0.00149*(tou1/tbar)^4);
    i=i+1;
end

//OUTPUT
printf('\n\t\t\tXBbar calculated for Models');
printf('\nReported Data');
printf('\ntbar(min)\tXBbar,obs\tUniform Reaction\
    tShrinking-Core, Rection Control\t\tShrinking-
    Core, Diffusion Control');
i=1;
while i<=n
    mprintf('\n%f\t%f\t%f\t\t%f\t\t\t\t%f',tbar1(i),
        XBbarr(i),XBbar1(i),XBbar2(i),XBbar3(i));
    i=i+1;
end

//=========================================END OF PROGRAM
    ==============================================================
```

Scilab code Exa 18.4 Scale up of a Reactor with Flowing Solids

```
//Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491

//Chapter-18, Example 4, Page 462
//Title: Scale-up of a Reactor with Flowing Solids
//
    ================================================================
```

```

clear
clc

//INPUT
W=1;//Bed weight in kg
F1=0.01;//Solid feed rate in kg/min
dp=[200;600];//Particle size in micrometer
XBbar=[0.85;0.64];//Average conversion for corresponding particle sizes
rhos=2500;//Density of solid in kg/m^3
ephsilonm=0.4;//Void fracton of fixed bed
F11=4;//Feed rate of solids in tons/hr
XBbar1=0.98;
dp1=600;
pi=3.14;

//CALCULATION
//Shrinking-Core, Rection Control
n=length(dp);
i=1;
touguess=2;//Guess value of tou
while i<=n
    function[fn]=solver_func2(tou)//Function defined for solving the system
        fn=(1-XBbar(i))-(0.25*tou/107)+(0.05*(tou/107)^2)-((1/120)*(tou/107)^3);
    endfunction
    [tou(i)]=fsolve(touguess,solver_func2,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
    i=i+1;
end
tou1=tou(2);

//For a single stage fluidized roaster
tbar1=0.25*(tou1/(1-XBbar1))/60;//Mean residence
```

```
time of solids in reactor in hr from Eqn.(24)
W1=F11*tbar1;
dtguess=2;//Guess value of tou
function[fn]=solver_func3(dt)//Function defined for solving the system
    fn=W1*10^3-(pi/4)*dt^2*0.5*dt*rhos*(1-ephsilonm);//Since Lm=0.5dt
endfunction
[dt]=fsolve(dtguess,solver_func3,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
Lm=dt/2;//Length of bed required

//For a two-stage fluidized roaster
tbar2=tou1*sqrt(1/(20*(1-XBbar1)))/60;//Mean residence time of solids in reactor in hr from Eqn.(30)
W2=F11*tbar2;
dtguess1=2;//Guess value of tou
function[fn]=solver_func4(dt)//Function defined for solving the system
    fn=W2*10^3-(pi/4)*dt^2*0.5*dt*rhos*(1-ephsilonm);//Since Lm=0.5dt
endfunction
[dt1]=fsolve(dtguess,solver_func4,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
Lm1=dt1/2;//Length of bed required

//OUTPUT
printf('\nSingle stage fluidized roaster');
printf('\n\tWeight of bed needed:%ftons',W1);
printf('\n\tDiameter of reactor:%fm',dt);
printf('\n\tLength of bed:%fm',Lm);
printf('\nTwo-stage fluidized roaster');
printf('\n\tWeight of bed needed:%ftons',W2);
printf('\n\tDiameter of reactor:%fm',dt1);
printf('\n\tLength of bed:%fm',Lm1);
```

```
printf('\nThese results show that this operation can be accomplished in a single bed of %ftons or in two beds of %f tons each.',W1,W2);

//========================================END OF PROGRAM
```

Scilab code Exa 18.5 Design of a Roaster for Finely Ground Ore

```
//Kunii D., Levenspiel O., 1991. Fluidization Engineering(II Edition). Butterworth-Heinemann, MA, pp 491

//Chapter-18, Example 5, Page 468
//Title: Design of a Roaster for Finely Ground Ore
//==========================================================================

clear
clc

//INPUT
T=900;//Temperature in roaster in degree C
P=101325;//Pressure in Pa
R=8.314;//Universal gas constant
dpbar=150;//Average particle size in micrometer
rhosbar=4130;//Average particle density in kg/m^3
kc=0.015//Rate constant in m/s for reaction which follows shrinking core model
Ds=8E-6;//Diffusion coefficient of solid in m^2/s
uo=0.6;//Superficial gas velocity in m/s
D=2.3E-4;//Diffusion coefficient of gas in m^2/s
```

```
Lm=1;//Length of fixed bed in m
dte=0.4;//Equivalent diameter of bed
umf=0.025;//Velocity at minimum fluidization
    condition in m/s
ephsilonm=0.45;//Void fraction of fixed bed
ephsilonmf=0.50;//Void fraction at minimum fluidized
     condition
db=0.2;//Estimated bubble size in m
gammab=0.005;//Ratio of volume of dispersed solids
    to that of bubble phase
Fo=2;//Feed rate of solids in kg/s
XA=0.6677;//Conversion of Oxygen
xA=0.21;//Mole fraction of oxygen in feed
mB=0.09744;//Molecular weight of ZnS
F=0.85;//Fraction of open area
g=9.81;//Acceleration due to gravity in square m/s^2
pi=3.14;

//CALCULATION
//(a)Extreme Calculation
a=3/2;//Stoichiometric coefficient of Oxygen in the
    reaction equation
At=(Fo/mB)*(a)/(uo*(273/(T+273))*(XA*xA)/0.0224);
dt=sqrt(At/F*4/pi);

//(b)The Three-Step Procedure
//Step 1. Conversion of gas
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from
    Eqn.(6.7)
ub=1.6*{(uo-umf)+1.13*db^0.5}*dte^1.35+ubr;//Bubble
    rise velocity for Geldart B particle
delta=uo/ub;//Fraction of bed in bubbles from Eqn
    .(6.29)
ephsilonf=1-(1-delta)*(1-ephsilonmf);//Void fraction
     of fixed bed from Eqn.(6.20)
fw=0.15;//Wake volume to bubble volume from Fig
    .(5.8)
gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw
```

```
);//Volume of solids in cloud to that of the bubble from Eqn.(6.36)
gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-gammac;//Volume of solids in emulsion to that of the bubble from Eqn.(6.35)
Kbc=4.5*(umf/db)+5.85*((D^0.5*g^0.25)/db^(5/4));//Gas interchange coefficient between bubble and cloud from Eqn.(10.27)
Kce=6.77*((D*ephsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;//Gas interchange coefficient between emulsion and cloud from Eqn.(10.34)
x=delta*Lm*(1-ephsilonm)/((1-ephsilonf)*uo);//Term Lf/ub of Eqn.(12.16) from Eqn.(6.19)
CAi=xA*P/(R*(T+273));//Initial concentration of oxygen

//Step 2.Conversion of solids
rhob=rhosbar/mB;//Density of ZnS
kbar=(kc^-1+(dpbar*10^-6/(12*Ds))^-1)^-1;//Modified rate constant from Eqn.(11)
tbar=At*Lm*(1-ephsilonm)*rhosbar/Fo;//Mean residence time of solids
Krguess=2;//Guess value of Kr
function[fn]=solver_func(Kr)//Function defined for solving the system
    Kf=gammab*Kr+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)+(1/(gammae*Kr))))));//Reaction rate for fluidized bed from Eqn.(14)
    XA=1-exp(-x*Kf);//Conversion of oxygen from Eqn.(42)
    CAbar=(CAi*XA*uo)/(Kr*Lm*(1-ephsilonm));//Average concentration of oxygen from Eqn.(43)
    tou=rhob*dpbar*10^-6*a/(2*kbar*CAbar);//Time for complete reaction from Eqn.(9)
    y=tbar/tou;//Term tbar/tou
    XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average conversion of ZnS from Eqn.(22)
    //Step 3. Material balance of both streams
```

```
    fn=(Fo/mB)*XBbar-(At*uo*CAi*XA/a);//From Eqn.(44b)
endfunction
[Kr]=fsolve(Krguess,solver_func,1E-6);//Using inbuilt function fsolve for solving for Kr
Kf=gammab*Kr+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)+(1/(gammae*Kr))))));//Reaction rate for fluidized bed from Eqn.(14)
XA=1-exp(-x*Kf);//Conversion of oxygen from Eqn.(42)
CAbar=(CAi*XA*uo)/(Kr*Lm*(1-ephsilonmf));//Average concentration of oxygen from Eqn.(43)
tou=rhob*dpbar*10^-6*a/(2*kbar*CAbar);//Time for complete reaction from Eqn.(9)
y=tbar/tou;//Term tbar/tou
XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average conversion of ZnS from Eqn.(22)


//(c) For other feed rates of solids
F1=[2;2.5;3;3.5];//Various feed rates of solids in kg/s
n=length(F1)
i=1;
Krguess1=2;//Guess value of Kr
while i<=n
    tbar1(i)=At*Lm*(1-ephsilonm)*rhosbar/F1(i);//Mean residence time of solids
    function[fn]=solver_func1(Kr)//Function defined for solving the system
        Kf1=gammab*Kr+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)+(1/(gammae*Kr))))));//Reaction rate for fluidized bed from Eqn.(14)
        XA1=1-exp(-x*Kf1);//Conversion of oxygen from Eqn.(42)
        CAbar1=(CAi*XA1*uo)/(Kr*Lm*(1-ephsilonm));//Average concentration of oxygen from Eqn.(43)
```

```
        tou1=rhob*dpbar*10^-6*a/(2*kbar*CAbar1);//Time for complete reaction from Eqn.(9)
        y1(i)=tbar1(i)/tou1;//Term tbar/tou
        XBbar1(i)=3*y1(i)-6*y1(i)^2+6*y1(i)^3*(1-exp(-1/y1(i)));//Average conversion of ZnS from Eqn.(22)
        //Step 3. Material balance of both streams
        fn=(F1(i)/mB)*XBbar1(i)-(At*uo*CAi*XA1/a);//From Eqn.(44b)
    endfunction
    [Kr1(i)]=fsolve(Krguess1,solver_func1,1E-6);//Using inbuilt function fsolve for solving Eqn.(23) for tou
    Kf1(i)=gammab*Kr1(i)+1/((1/Kbc)+(1/(gammac*Kr1(i)+1/((1/Kce)+(1/(gammae*Kr1(i)))))));//Reaction rate for fluidized bed from Eqn.(14)
    XA1(i)=1-exp(-x*Kf1(i));//Conversion of oxygen from Eqn.(42)
    CAbar1(i)=(CAi*XA1(i)*uo)/(Kr1(i)*Lm*(1-ephsilonmf));//Average concentration of oxygen from Eqn.(43)
    tou1(i)=rhob*dpbar*10^-6*a/(2*kbar*CAbar1(i));//Time for complete reaction from Eqn.(9)
    y1(i)=tbar1(i)/tou1(i);//Term tbar/tou
    XBbar1(i)=3*y1(i)-6*y1(i)^2+6*y1(i)^3*(1-exp(-1/y1(i)));//Average conversion of ZnS from Eqn.(22)
    i=i+1;
end

//OUTPUT
printf('\nExtreme Calculation');
printf('\n\tDiameter of tube with all its internals:%fm',dt);
printf('\nThree step procedure');
printf('\n\tConversion of ZnS:%f',XBbar);
printf('\nFor other feed rates of solids');
printf('\n\tFeed(kg/s)\ttbar(s)\t\tXBbar/XA\tKrbar(s
```

```
^-1)\tCAbar/CAi\ttou(s)\t\tXA\t\tXB');
i=1;
while i<=n
    mprintf('\n\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f',F1(
       i),tbar1(i),XBbar1(i)/XA1(i),Kr1(i),CAbar1(i)
       /CAi,tou1(i),XA1(i),XBbar1(i));
    i=i+1;
end

//=====================================END OF PROGRAM
   ========================================================
```

Scilab code Exa 18.6 Design of a Roaster for Coarse Ore

```
//Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491

//Chapter-18, Example 5, Page 471
//Title: Design of a Roaster for Coarse Ore
//
   ==================================================================


clear
clc

//INPUT
T=900;//Temperature in roaster in degree C
P=101325;//Pressure in Pa
R=8.314;//Universal gas constant
dp=750;//Particle size in micrometer5
Fo=2.5;//Feed rate of solids in kg/s
```

```
uo=0.6;//Superficial gas velocity in m/s
W=80140;//Weight of bed in kg
ephsilonmf=0.50;//Void fraction at minimum fluidized condition
umf=0.5;//Velocity at minimum fluidization condition in m/s
db=0.2;//Estimated bubble size in m
g=9.81;//Acceleration due to gravity in square m/s^2
Lm=1;//Length of fixed bed in m
ephsilonm=0.45;//Void fraction of fixed bed
xA=0.21;//Mole fraction of oxygen in feed
kc=0.015//Rate constant in m/s for reaction which follows shrinking core model
Ds=8E-6;//Diffusion coefficient of solid in m^2/s
rhosbar=4130;//Average particle density in kg/m^3
mB=0.09744;//Molecular weight of ZnS
a=3/2;//Stoichiometric coefficient of Oxygen in the reaction equation

//CALCULATION
//Selection of models to represent reactor
ubr=0.711*(g*db)^0.5;//Rise velocity of bubble from Eqn.(6.7)
f=ubr/(umf/ephsilonmf);

//Step 1.
ub=uo-umf+ubr;//Rise velocity of bubbles from Eqn.(6.8)
delta=(uo-umf)/(ub+2*umf);//Fraction of the bed in bubbles from Eqn.(6.26)
Krguess=2;//Guess value of Kr
x=Lm*(1-ephsilonm)*umf*(1-delta)/uo^2;
CAi=xA*P/(R*(T+273));//Initial concentration of oxygen

//Step 2.
kbar=(kc^-1+(dp*10^-6/(12*Ds))^-1)^-1;//Modified rate constant from Eqn.(11)
```

```
tbar=W/Fo;//Mean residence time of solids from Eqn.(14.2)
rhob=rhosbar/mB;//Density of ZnS
function[fn]=solver_func1(Kr)//Function defined for solving the system
    XA=1-exp(-x*Kr);//Conversion from Eqn.(42)
    CAbar=(CAi*XA*uo^2)/(Kr*Lm*(1-ephsilonm)*umf*(1-delta));//Average concentration of oxygen from Eqn.(43)
    tou=rhob*dp*10^-6*a/(2*kbar*CAbar);//Time for complete reaction from Eqn.(9)
    y=tbar/tou;//Term tbar/tou
    XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average conversion of ZnS from Eqn.(22)
    //Step 3.
    fn=XBbar-1.2*XA;//From Table E5, for Fo=2.5kg/s
endfunction
[Kr]=fsolve(Krguess,solver_func1,1E-6);//Using inbuilt function fsolve for solving for Kr
XA=1-exp(-x*Kr);//Conversion from Eqn.(42)
CAbar=(CAi*XA*uo^2)/(Kr*Lm*(1-ephsilonm)*umf*(1-delta))//Average concentration of oxygen from Eqn.(43)
tou=rhob*dp*10^-6*a/(2*kbar*CAbar);//Time for complete reaction from Eqn.(9)
y=tbar/tou;//Term tbar/tou
XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average conversion of ZnS from Eqn.(22)

//OUTPUT
printf('\nSelection of models to represent reactor');
printf('\n\tSince ratio ubr/(umf/ephsilonmf)= %f <1, the reactor is operating in slow bubble regime',f);
printf('\n\tSince particle size =%f micrometer, they react according to shrinking-core model',dp);
printf('\n\tConversion obtained for %f micrometer
```

```
particle:%f',dp,XBbar);

//==================================================END OF PROGRAM
=================================================================
```