

Scilab Textbook Companion for
Fluidization Engineering
by K. Daizo And O. Levenspiel¹

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

August 17, 2013

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

Book Description

Title: 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	5
3 Fluidization and Mapping of Regimes	9
4 The Dense Bed	17
5 Bubbles in Dense Beds	25
6 Bubbling Fluidized Beds	29
7 Entrainment and Elutriation from Fluidized Beds	41
8 High velocity Fluidization	51
9 Solid Movement Mixing Segregation and Staging	57
10 Gas Dispersion and Gas Interchange in Bubbling Beds	64
11 Particle to Gas Mass and Heat Transfer	72
12 Conversion of Gas in Catalytic Reactions	83
13 Heat Transfer between Fluidized Beds and Surfaces	95
14 The RTD and Size Distribution of Solids in Fluidized Beds	103
15 Circulation Systems	113
16 Design for Physical Operations	123

17 Design of Catalytic Reactors	135
18 The Design of Noncatalytic Gas Solid Reactors	148

List of Scilab Codes

Exa 3.1	Size Measure of Nonuniform Solids	9
Exa 3.2	Estimation of Minimum Fluidizing Velocity	10
Exa 3.3	Estimation of Terminal Velocity of Falling Particles . .	12
Exa 3.4	Prediction of Flow Regimes	13
Exa 4.1	Design of a Perforated Plate Distributor	17
Exa 4.2	Design of a Tuiere Distributor	20
Exa 4.3	Power Requirement for a Fluidized Coal Combustor .	22
Exa 5.1	Characteristics of a Singe Bubble	25
Exa 5.2	Initial Bubble Size at a Distributor	26
Exa 6.1	Bubble Size and Rise Velocity in Geldart A Beds . . .	29
Exa 6.2	Bubble Size and Rise Velocity in Geldart B Beds . . .	30
Exa 6.3	Scale down of a Commercial Chlorinator	33
Exa 6.4	Reactor Scale up for Geldart A Catalyst	35
Exa 6.5	Reactor Scale up for Geldart B Catalyst	39
Exa 7.1	Entrainment from fine particle beds with high freeboard	41
Exa 7.2	Entrainment from large particle beds with high freeboard	42
Exa 7.3	Entrainment from beds with a wide size distribution of solids	43
Exa 7.4	kstar from steady state experiments	45
Exa 7.5	Comparing predictions for kstar	46
Exa 7.6	Entrainment from a short vessel	49
Exa 8.1	Performance of a Fast Fluidized Vessel	51
Exa 9.1	Vertical Movement of Solids	57
Exa 9.2	Horizontal Drift Of Solids	59
Exa 9.3	Design of Baffle Plates	61
Exa 10.1	Estimate Interchange Coefficients in Bubbling Beds . .	64
Exa 10.2	Compare the Relative Importance of Kbc and Kce . .	66

Exa 10.3	Compare Interchange Rates for Adsorbed and Nonadsorbed Gases	69
Exa 11.1	Fitting Reported Mass Transfer Data with the Bubbling Bed Model	72
Exa 11.2	The Effect of m on Bubble Emulsion Interchange	74
Exa 11.3	Fitting Reported Heat Transfer Data with the Bubbling Bed Model	77
Exa 11.4	Heating a Particle in a Fluidized Bed	79
Exa 12.1	Fine Particle Geldart A Bubbling Bed Reactor	83
Exa 12.2	Commercial Sized Phthalic Anhydride Reactor	85
Exa 12.3	Bubbling Bed Reactor for Intermediate Sized Reactor	88
Exa 12.4	Reaction in the Slow Bubble Regime	90
Exa 12.5	Conversion in the Freeboard of a Reactor	92
Exa 13.1	h on a Horizontal Tube Bank	95
Exa 13.2	Effect of Gas Properties on h	97
Exa 13.3	Effect of Particle Size on h	99
Exa 13.4	Freeboard Heat Exchange	100
Exa 14.1	Flow with Elutriation	103
Exa 14.2	Flow with Elutriation and Change in Density of Solids	105
Exa 14.3	Single Size Feed of Shrinking Particles	108
Exa 14.4	Wide Size Distribution of Shrinking Particle	109
Exa 14.5	Elutriation and Attrition of Catalyst	110
Exa 15.1	Circulation Rate when Deactivation Controls	113
Exa 15.2	Circulation Rate when Heat Duty Controls	114
Exa 15.3	Aeration of Fine Particle Downcomer	116
Exa 15.4	Circulation in Side by Side Beds	118
Exa 15.5	Steam Seal of a Coarse Particle Downcomer	120
Exa 16.1	Single Stage Limestone Calciner	123
Exa 16.2	Multistage Limestone Calciner	125
Exa 16.3	Multistage Adsorber	128
Exa 16.4	Dryer Kinetics and Scale up	129
Exa 16.5	Solvent Recovery from Polymer Particles	131
Exa 17.1	Reactor Development Program	135
Exa 17.2	Design of a Commercial Acrylonitrile Reactor	140
Exa 17.3	Reactor Regenerator with Circulating Catalyst Catalytic Cracking	142
Exa 18.1	Kinetics of Zinc Blende Roasting	148
Exa 18.2	Kinetics of Carbon Burning	149

Exa 18.3	Roasting Kinetics from Flowing Solids Data	151
Exa 18.4	Scale up of a Reactor with Flowing Solids	153
Exa 18.5	Design of a Roaster for Finely Ground Ore	156
Exa 18.6	Design of a Roaster for Coarse Ore	161

List of Figures

10.1 Estimate Interchange Coefficients in Bubbling Beds	67
11.1 Fitting Reported Mass Transfer Data with the Bubbling Bed Model	75
11.2 Fitting Reported Heat Transfer Data with the Bubbling Bed Model	80
14.1 Flow with Elutriation and Change in Density of Solids	108

Chapter 3

Fluidization and Mapping of Regimes

Scilab code Exa 3.1 Size Measure of Nonuniform Solids

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-3, Example 1, Page 68
4 //Title: Size Measure of Nonuniform Solids
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 weight = [0;60;150;270;330;360]; // Weight in grams
   for the oversized particles
11 psize = [50;75;100;125;150;175]; //PSD in micrometers
12
13 //CALCULATION
14 len = length(psize); // To obtain the size of input
```

```

    array
15 // Computation of sauter mean diameter for the given
    PSD
16 i = 1;
17 while i<len
18     dpi(i)=(psize(i,:)+ psize(i+1,:))/2;
19     weightf(i)=(weight(i+1)-weight(i))/weight(6)
        ;
20     dp(i)=weightf(i)/dpi(i);
21     i=i+1;
22 end
23 dpbar=1/sum(dp); // Calculation of average particle
    daimeter Eq.(15)
24
25 //OUTPUT
26 mprintf('\n The Sauter mean diameter of the material
    with the given particle size distribution = %f
    micrometer ',dpbar);
27
28 //=====END OF PROGRAM

```

Scilab code Exa 3.2 Estimation of Minimum Fluidizing Velocity

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-3, Example 2, Page 76
4 //Title: Estimation of Minimum fluidizing velocity
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 ephsilon=0.55; //Void fraction of bed
11 rhog=0.0012; //Density of gas in g/cc
12 myu=.00018; //Viscosity of gas in g/cm s
13 dpbar=0.016; //Mean diameter of solids in centimeter
14 phis=0.67; //Sphericity of solids
15 rhos=2.6; //Density of solids in g/cc
16 g=980; //Acceleration due to gravity in square cm/s^2
17
18 //CALCULATION
19 //Computation of umf using the simplified equation
    for small particles
20 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)
21 Re=(dpbar*umf*rhog)/myu; //To calculate Reynolds
    number for particle
22
23 //Computation of umf if neither void fraction of bed
    nor sphericity is known
24 c1=28.7; c2=0.0494; //Value of constants from Table
    4, page 70
25 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)
26 err=((umf-umf1)/umf)*100; //Calculation of error from
    experimental value
27
28 //OUTPUT
29 if Re<20 then
30     mprintf('\nThe particle Reynolds no = %f',Re)
31     printf('\nThe simplified equation used for
        calculating minimum fluidizing velocity is

```

```

        valid. ');
32 end
33 mprintf('\nThe minimum fluidizing velocity by
    simplified equation for small particles = %fcm/s',
    ,umf);
34 mprintf('\nThe minimum fluidizing velocity by
    equation for coarse partilces = %fcm/s',umf1);
35 mprintf('\nThis value is %f percent below the
    experimentally reported value.',err);
36
37 //=====END OF PROGRAM

```

Scilab code Exa 3.3 Estimation of Terminal Velocity of Falling Particles

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-3, Example 3, Page 82
4 //Title: Estimation of terminal velocity of falling
    particles
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 rhog=1.2e-3;//Density of air in g/cc
11 myu=1.8e-4//Viscosity of air in g/cm s
12 dpbar=0.016//Mean diameter of solids in centimeter
13 phis=0.67;//Sphericity of solids

```

```

14 rhos=2.6; //Density of solids in g/cc
15 g=980 //Acceleration due to gravity in square cm/s^2
16
17 //CALCULATION
18 dpstar=dpbar*((rhog*(rhos-rhog)*g)/myu^2)^(1/3); //
    Calculation of dimensionless particle size Eq
    .(31)
19 utstar=((18/(dpstar^2))+(2.335-(1.744*phis))/(dpstar
    ^0.5))^-1; //Calculation of dimensionless gas
    velocity Eq.(33)
20 ut=utstar*((myu*(rhos-rhog)*g)/rhog^2)^(1/3); //
    Calculation of terminal velocity of falling
    particles Eq.(32)
21
22
23 //OUTPUT
24 mprintf('\nThe dimensionless particle size = %f',
    dpstar);
25 mprintf('\nThe dimensionless gas velocity = %f',
    utstar);
26 mprintf('\nThe terminal velocity of falling
    particles = %fcm/s', ut);
27
28 //=====END OF PROGRAM

```

Scilab code Exa 3.4 Prediction of Flow Regimes

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-3, Example 4, Page 91

```

```

4 //Title: Prediction of flow regime
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 rhos=1.5; //Density of Solid in g/cc
11 uo1=40; uo2=80; //Superficial gas velocity in cm/s
12 dp1=0.006; dp2=0.045; //Particle size in centimeter
13 rhog1=1.5E-3; rhog2=1E-3; //Density of gas in g/cc
14 myu1=2E-4; myu2=2.5E-4; //Viscosity of air in g/cm s
15 g=980; //Acceleration due to gravity in square cm/s^2
16
17 //CALCULATION
18 //for smaller particles
19 dpstar1=dp1*((rhog1*(rhos-rhog1)*g)/myu1^2)^(1/3); //
    Calculation of dimensionless particle diameter Eq
    .(31)
20 uostar1=uo1*((rhog1^2)/((myu1)*(rhos-rhog1)*g))
    ^ (1/3);
21 uostar2=uo2*((rhog1^2)/((myu1)*(rhos-rhog1)*g))
    ^ (1/3); //Calculation of dimensionless superficial
    gas velocity Eq.(32)
22
23 //for larger particles
24 dpstar2=dp2*((rhog2*(rhos-rhog2)*g)/myu2^2)^(1/3); //
    Calculation of dimensionless particle diameter Eq
    .(31)
25 uostar3=uo1*((rhog2^2)/((myu2)*(rhos-rhog2)*g))
    ^ (1/3);
26 uostar4=uo2*((rhog2^2)/((myu2)*(rhos-rhog2)*g))
    ^ (1/3); //Calculation of dimensionless superficial
    gas velocity Eq.(32)
27
28
29 //OUTPUT

```

```

30 printf('\nFor particle of size %f centimeter',dp1);
31 mprintf('\nThe dimensionless particle diameter = %f',
    ,dpstar1);
32 mprintf('\nThe dimensionless superficial gas
    velocity = %fcm/s(for superficial gas velocity of
    %fcm/s)',uostar1,uo1);
33 mprintf('\nThe dimensionless superficial gas
    velocity = %fcm/s(for superficial gas velocity of
    %fcm/s)',uostar2,uo2);
34 mprintf('\n\nFrom Fig.16(page 89)comparing u*=%f vs
    dp*=%f',uostar1,dpstar1);
35 mprintf('\nFor Superficial gas velocity =%f \nMode
    of Fluidization:Onset of turbulent fluidization
    in an ordinary bubbling bed',uo1);
36 mprintf('\nFrom Fig.16(page 89)comparing u* =%f vs
    dp* =%f',uostar2,dpstar1);
37 mprintf('\nFor Superficial gas velocity =%f \nMode
    of Fluidization:Fast fluidization(requires a
    circulating solid system)',uo2);
38 printf('\n\nFor particle of size %f centimeter',dp2)
39 mprintf('\nThe dimensionless particle diameter = %f',
    ,dpstar2);
40 mprintf('\nThe dimensionless superficial gas
    velocity = %fcm/s(for superficial gas velocity of
    %fcm/s)',uostar3,uo1);
41 mprintf('\nThe dimensionless superficial gas
    velocity = %fcm/s(for superficial gas velocity of
    %fcm/s)',uostar4,uo2);
42 mprintf('\n\nFrom Fig.16(page 89)comparing u*=%f vs
    dp*=%f',uostar3,dpstar2);
43 mprintf('\nFor Superficial gas velocity =%f \nMode
    of Fluidization:Bubbling Fluidization',uo1);
44 mprintf('\nFrom Fig.16(page 89)comparing u* =%f vs
    dp* =%f',uostar4,dpstar2);
45 mprintf('\nFor Superficial gas velocity =%f \nMode
    of Fluidization:Bubbling Fluidization',uo2);
46
47 //=====END OF PROGRAM

```




Chapter 4

The Dense Bed

Scilab code Exa 4.1 Design of a Perforated Plate Distributor

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-4, Example 1, Page 106
4 //Title: Design of a Perforated Plate Distributor
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 dt=4;//Vessel diameter in m
11 Lmf=2;//Length of the bed in m
12 ephsilonmf=0.48;//Void fraction of bed
13 rhos=1500;//Density of solid in kg/m^3
14 rhog=3.6;//Density of gas in kg/m^3
15 myu=2E-5;//Viscosity of gas in kg/m s
16 po=3;//Pressure of inlet gas in bar
17 uo=0.4;//Superficial velocity of gas in m/s
```

```

18 uorm=40; //Maximum allowable jet velocity from holes
    in m/s
19 g=9.80; //Acceleration due to gravity in m/s^2
20 gc=1;
21 pi=3.1428;
22
23 //CALCULATION
24 //Computation of minimum allowable pressure drop
    through the distributor
25 deltapb={(1-epsilon*mf)*(rho_s-rho_g)*g*Lmf}/gc; //
    Calculation of pressure drop in bed using Eqn
    .(3.17)
26 deltapd=0.3*deltapb; //Calculation of pressure drop
    in distributor using Eqn.(3)
27
28 //Computation of orifice coefficient
29 Ret=(dt*uo*rho_g)/myu;
30 if Ret>=3000 then Cd=0.60;
31 elseif Ret>=2000 then Cd=0.61;
32 elseif Ret>=1000 then Cd=0.64;
33 elseif Ret>=500 then Cd=0.68;
34 elseif Ret>=300 then Cd=0.70;
35 elseif Ret>=100 then Cd=0.68;
36 end
37
38 //Computation of gas velocity through orifice
39 uor=Cd*((2*deltapd)/rho_g)^0.5; //Calculation of gas
    velocity through orifice by using Eqn.(12)
40 f=(uo/uor)*100; //Calculation of fraction of open
    area in the perforated plate
41
42
43 //Computation of number of orifices per unit area of
    distributor
44 dor=[0.001;0.002;0.004]; //Different orifice
    diameters in m
45 n=length(dor);
46 i=1;

```

```

47 while i<=n
48     Nor(i)=(uo*4)/(pi*uor*(dor(i))^2); // Calculation
        of number of orifices by using Eqn.(13)
49     i=i+1;
50 end
51
52 //OUTPUT
53 mprintf('\nThe pressure drop in bed:%fPa',deltapb);
54 mprintf('\nThe minimum allowable pressure drop in
        distributor:%fPa',deltapd);
55 if uor<uorm then mprintf('\nThe gas veleocity of %fm
        /s is satisfactory',uor);
56     else mprintf('\nThe gas veleocity of %fm/s is
        not satisfactory',uor);
57 end
58 if f<10 then mprintf('\nThe fraction of open area of
        %f percent is allowable',f);
59     else mprintf('\nThe fraction of open area of %f
        percent is not allowable',f);
60 end
61 printf('\nDiameter of orifice(m)');
62 printf('\tNumber of orifices per unit area(per sq.m)
        ');
63 j=1;
64 while j<=n
65     mprintf('\n%f',dor(j));
66     mprintf('\t\t%f',Nor(j));
67     j=j+1;
68 end
69 printf('\nThis number can be rounded off. ');
70 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));
71
72 //=====END OF PROGRAM

```

Scilab code Exa 4.2 Design of a Tuyere Distributor

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-4, Example 2, Page 108
4 //Title: Design of a Tuyere Distributor
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 lor=0.1; //Minimum allowable tuyere spacing in m
11 uorm=30; //Maximum allowable jet velocity from the
   tuyere in m/s
12 uo=0.4; //Superficial velocity of gas in m/s
13 uor=30.2; //Gas velocity through orifice, from Exa 1,
   in m/s
14 Cd=0.6; //Discharge coefficient from Exa 1
15 rhog=3.6 //Density of gas in kg/m^3
16 pi=3.1428;
17
18 //CALCULATION
19 Nor=1/(lor^2); //Calculation of number of orifices
   per unit area by assuming minimum spacing for
   tuyeres
20 dor={ (4/pi)*(uo/uor)*(1/Nor) }^0.5; //Calculation of
   diameter of inlet orifice by using Eqn.(13)
21
22 //Computation of diameter of hole for different
```

```

        number of holes per tuyere
23 q=(lor^2)*uo; //Volumetric flow rate in m^3/s
24 Nh=[8;6;4]; //Different number of holes per tuyere
25 n=length(Nh);
26 i=1;
27 while i<=n
28     dh(i)=((((q/Nh(i))*(4/pi))/uorm)^0.5); //
        Calculation of diameter of holes
29     i=i+1;
30 end
31 deltaph=(rhog/2)*((uor/Cd)^2);
32
33 //OUTPUT
34 printf('\nNumber of holes(number of holes/tuyeres)')
    ;
35 printf('\tDiameter of hole(m)');
36 j=1;
37 while j<=n
38     mprintf('\n%f',Nh(j));
39     mprintf('\t\t\t\t\t%f',dh(j));
40     j=j+1;
41 end
42 printf('\nThe design chosen is as follows');
43 printf('\n\tTuyeres are as shown in Fig.2(b),page 97
    ');
44 mprintf('\n\tNumber of holes = %f(Since rectangular
    pitch is chosen for tuyeres)',Nh(2));
45 mprintf('\n\tDiameter of hole = %fm',dh(2));
46 mprintf('\n\tDiameter of incoming high-pressure-drop
    orifice = %fm ID',dor);
47 printf('\nChecking the pressure drop in tuyeres');
48 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);
49
50 //=====END OF PROGRAM

```

Scilab code Exa 4.3 Power Requirement for a Fluidized Coal Combustor

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-4, Example 3, Page 110
4 //Title: Power Requirement for a Fluidized Coal
   Combustor(FBC)
5 //
6 clear all
7 clc
8
9 //INPUT
10 deltapd=[3;10]//Distributor pressure drop in kPa
11 deltapd2=10;//Distributor pressure drop in kPa
12 po=101;//Entering air pressure in kPa
13 To=20;//Entering air temperature in degree C
14 y=1.4;//Fugacity of air
15 deltapb=10;//Pressure drop in bed in kPa
16 p3=103;//Pressure at the bed exit in kPa
17 F=8;//Feed rate of coal in tons/hr
18 H=25;//Gross heatig value of coal in MJ/kg
19 Fa=10;//Air required at standard condition in mm^3/
   kg
20 etac=0.75;//Efficiency of compressor
21 etap=36;//Efficiency of plant in %
22
23 //CALCULATION
24 //Calculation of volumetric flow rate of air
```

```

25 vo=((F*1000)*Fa*((To+273)/273))/3600;
26
27 //Case(a) Distributor Pressure drop = 3kPa and Case(
    b) Distributor Pressure drop = 10kPa
28 n=length(deltapd);
29 i=1;
30 while i<=n
31     p2(i)=p3+deltapb;//Calculation of pressure at
        the entrance of the bed
32     p1(i)=p2(i)+deltapd(i);//Calculation of pressure
        before entering the bed
33     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)
34     i=i+1;
35 end
36
37 //Case(c) 50% of the required bypassed to burn the
    volatile gases. Distributor Pressure drop = 3kPa
38 //No change in pressure drop from case(a)
39 v1=vo/2;//New volumetric flow rate of air
40 ws1=ws(1)/2;//Power required for blower for primary
    air
41 ws2=(y/(y-1))*po*v1*((p3/po)^((y-1)/y)-1)*(1/etac);
    //Power required for blower for bypassed air
42 wst=ws1+ws2;//Total power required for the two
    blowers
43 p=((ws(1)-wst)/ws(1))*100;//Saving in power when
    compared to case(a)
44
45 //OUTPUT
46 printf('\nCase(a) ');
47 mprintf('\n\tVolumetric flow rate of air = %f m^3/hr
    ',vo);
48 mprintf('\n\tPower required for compressor = %f kW',
    ws(1));
49 printf('\nCase(b) ');
50 mprintf('\n\tVolumetric flow rate of air = %f m^3/hr

```



```

    ',vo);
51 mprintf('\n\tPower required for compressor = %f kW',
    ws(2));
52 printf('\nCase(c)');
53 mprintf('\n\tVolumetric flow rate of air = %f m^3/hr
    ',v1);
54 mprintf('\n\tPower required for compressor for
    primary air = %f kW',ws1);
55 mprintf('\n\tPower required for blower for bypassed
    air = %f kW',ws2);
56 mprintf('\n\tTotal power required for the two
    blowers = %f kW',wst);
57 mprintf('\n\tPower saved compared to case(a) = %f
    percent',p);
58
59 //=====END OF PROGRAM

```

Chapter 5

Bubbles in Dense Beds

Scilab code Exa 5.1 Characteristics of a Single Bubble

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-5, Example 1, Page 126
4 //Title: Characteristics of a Single Bubble
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 dt=60; //ID of tube in cm
11 dp=300; //Size of particles of bed in micrometers
12 umf=3; //Velocity at minimum fluidization condition
   in cm/s
13 ephsilonmf=0.5; //Void fraction of bed at minimum
   fluidization condition
14 db=5; //Diameter of bubble in cm
15 g=980; //Acceleration due to gravity in cm/s^2
```

```

16
17 //CALCULATION
18 //Computation of rise velocity of bubble
19 if (db/dt)<0.125    then ubr=(0.711*((g*db)^0.5));//
    Rise velocity by Eqn.(3)
20 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)
21 end
22
23 //Computation of cloud thickness
24 Rb=db/2;//Radius of bubble
25 uf=umf/epsilonmf;//Velocity of emulsion gas
26 Rc=Rb*((ubr+(2*uf))/(ubr-uf))^(1/3);//Radius of
    cloud by Eqn.(6)
27
28 //OUTPUT
29 mprintf('\nThe rise velocity of the bubble=%fcm/s',
    ubr);
30 mprintf('\nThe cloud thickness=%fcm',Rc-Rb);
31 mprintf('\nFrom Fig.8(page 124)comparing fw vs dp,
    for dp = %f micrometer, wake fraction = 0.24 ',dp)
    ;
32
33 //=====END OF PROGRAM

```

Scilab code Exa 5.2 Initial Bubble Size at a Distributor

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2

```

```

3 //Chapter -5, Example 2, Page 132
4 //Title: Initial Bubble Size at a Distributor
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 uo=15; // Superficial gas velocity in cm/s
11 umf=1; // Velocity at minimum fluidization condition
    in cm/s
12 lor=2; // Pitch of perforated plate in cm
13 g=980; // Acceleration due to gravity in cm/s^2
14 //CALCULATION
15 //Case(a) For porous plate
16 dbo1=(2.78/g)*(uo-umf)^2; // Initial bubble size using
    Eqn.(19)
17
18 //Case(b) For Perforated plate
19 Nor=(2/sqrt(3))*(1/lor)^2; // Number of orifices in cm
    ^-2
20 dbo2=(1.30/(g^0.2))*((uo-umf)/Nor)^0.4; // Initial
    bubble size using Eqn.(15) assuming initial bubble
    size is smaller than hole spacing
21
22 //OUTPUT
23 printf('\nCase(a) For porous plate ');
24 printf('\n\tInitial bubble size=%fcm',dbo1);
25 printf('\nCase(b) For Perforated plate ');
26 printf('\n\tInitial bubble size=%fcm',dbo2);
27 printf('\n\tSince %f<%f, the equation used is
    correct.',dbo2,lor);
28
29 //=====END OF PROGRAM

```

Chapter 6

Bubbling Fluidized Beds

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

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-6, Example 1, Page 150
4 //Title: Bubble Size and Rise Velocity in Geldart A
   Beds
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 z=0.5; //Height of bed in m
11 dt=0.5; //ID of tube in m
12 rhos=1.6; //Density of catalyst in g/cm^3
13 dpbar=60; //Average catalyst diameter in micrometer
14 umf=0.002; //Velocity at minimum fluidization
   condition in m/s
15 uo=0.2; //Superficial velocity in m/s
```

```

16 dor=2; //Diameter of orifice in mm
17 lor=20; //Pitch of perforated plate in mm
18 g=9.80; //g=980; //Acceleration due to gravity in m/s
    ^2
19
20 //CALCULATION
21 //Method 1. Procedure using Eqn.(10) & Eqn.(11)
22 db=(0.035+0.040)/2; //Bubble size at z=0.5m from Fig
    .7(a) & Fig.7(b)
23 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)
24
25 //Method 2. Werther's procedure
26 si=0.8; //From Fig.6 for Geldart A solids
27 ub2=si*(uo-umf)+(3.2*(dt^(1/3)))*(0.711*(g*db)^0.5);
    //Bubble velocity using Eqn.(9)
28
29 //OUTPUT
30 printf('\nMethod 1. Procedure using Eqn.(10) & Eqn
    .(11) ');
31 mprintf('\n\tDiameter of the bubble=%fm',db);
32 mprintf('\n\tRise velocity of the bubble=%fm/s',ub1)
    ;
33 printf('\nMethod 2. Werthers procedure ');
34 mprintf('\n\tDiameter of the bubble=%fm',db);
35 mprintf('\n\tRise velocity of the bubble=%fm/s',ub2)
    ;
36
37 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-6, Example 2, Page 151
4 //Title: Bubble Size and Rise Velocity in Geldart B
   Beds
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 z=0.5;//Height of bed in m
11 dt=0.5;//ID of tube in m
12 rhos=2.6;//Density of catalyst in g/cm^3
13 dpbar=100;//Averge catalyst diameter in micrometer
14 umf=0.01;//Velocity at minimum fluidization
   condition in m/s
15 uo=0.45;//Superficial velocity in m/s
16 dor=2;//Diameter of orifice in mm
17 lor=30;//Pitch of perforated plate in mm
18 g=9.80;//Acceleration due to gravity in m/s^2
19 pi=3.142857;
20
21 //CALCULATION
22 //Part(a).Bubble Size
23 Nor=(2/sqrt(3))*(1/lor^2);
24 dbo=5.5;
25
26 //Method 1.Werther's procedure for finding bubble
   size
27 z1=[0;5;10;20;30;50;70];
28 n=length(z1);
29 i=1;
30 while i<=n
31     db(i)=0.853*((1+0.272*(uo-umf)*100)^(1/3))

```



```

        *(1+0.0684*z1(i))^1.21;
32     i=i+1;
33 end
34 db1=0.163;//Since bubble size starts at dbo=5.5cm at
        z=0, we shift the curve accordingly to z=0.5m
35
36 //Method 2.Mori and Wen's procedure for finding
        bubble size
37 dbm=0.65*((pi/4)*((dt*100)^2)*(uo-umf)*100)^0.4;
38 db2=dbm-(dbm-dbo)*exp(-0.3^(z/dt));
39
40 //Part(b).Bubble Velocity
41 //Method 1.Procedure using Eqn.(12)
42 ub1=1.6*((uo-umf)+1.13*db1^0.5)*(dt^1.35)+(0.711*(g*
        db1)^0.5);
43
44 //Method 2.Werther's Procedure
45 si=0.65;
46 ub2=si*(uo-umf)+2*(dt^0.5)*(0.711*(g*db1)^0.5);
47
48 //Using Eqn.(7) & Eqn.(8)
49 ubr1=0.711*(g*db1)^0.5;
50 ubr2=0.711*(g*db2/100)^0.5
51 ub3=uo-umf+ubr1;
52 ub4=uo-umf+ubr2;
53
54 //OUTPUT
55 printf('\nBubble Size');
56 mprintf('\nInitial bubble size from Fig.5.14 for %fm
        /s = %fcm',uo-umf, dbo);
57 printf('\n\n\tMethod 1.Werthers procedure for
        finding bubble size');
58 printf('\n\n\t\tHeight of bed(cm)');
59 printf('\t\t\tBubble size(cm)');
60 m=length(z1);
61 j=1;
62 while j<=m
63     mprintf('\n\t\t\t%f', z1(j));

```

```

64     mprintf( '\t\t\t\t%f', db(j));
65     j=j+1;
66 end
67 printf( '\n\n\tMethod 2.Mori and Wens procedure for
        finding bubble size ');
68 mprintf( '\n\t\tMaximum expected bubble size=%fcm',
        dbm);
69 mprintf( '\n\t\tBubble size=%fcm', db2);
70 printf( '\nBubble Velocity ');
71 printf( '\n\n\tMethod 1.Procedure using Eqn.(12) ');
72 mprintf( '\n\t\tBubble velocity=%fm/s', ub1);
73 printf( '\n\n\tMethod 2.Werthers procedure ');
74 mprintf( '\n\t\tBubble velocity=%fm/s', ub2);
75 printf( '\nComparing the above results with the
        expressions of the simple two-phase theory ');
76 printf( '\n\tWerthers bubble size ');
77 mprintf( '\tBubble rise velocity=%fm/s\tBubble
        velocity=%fm/s', ubr1, ub3);
78 printf( '\n\tMori & Wens bubble size ');
79 mprintf( '\tBubble rise velocity=%fm/s\tBubble
        velocity=%fm/s', ubr2, ub4);
80
81 //=====END OF PROGRAM

```

Scilab code Exa 6.3 Scale down of a Commercial Chlorinator

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-6, Example 3, Page 153
4 //Title: Scale-down of a Commercial Chlorinator

```

```

5 //


---




---


6 clear all
7 clc
8
9 //INPUT
10 dpbar=53; //Average particle size in micrometer
11 s=[1;2]; //Size of Bermuda rock in cm
12 rhosbar=3200; //Average solid density of the coke-
    zircon mixture in kg/m^3
13 ephsilonm=0.5; //Void fraction for fixed bed
14 ephsilonf=0.75; //Void fraction for bubbling bed
15 rhogbar=0.64; //Average density of gas in kg/m^3
16 uo=14; //Superficial gas velocity in cm/s
17 myu=5E-5; //Viscosity of gas in kg/m s
18 T=1000; //Temperature in degree C
19 P=1; //Pressure in atm
20 dt=91.5; //ID of bed in cm
21 sh=150; //Slumped height in cm
22
23 //CALCULATION
24 rhog2=1.2; //Density of ambient air
25 myu2=1.8E-5; //Viscosity of ambient air
26 rhos2=rhog2*(rhosbar/rhogbar); //For the requirement
    of constant density ratio
27 m=((rhogbar*myu2)/(rhog2*myu))^(2/3); //Scale factor
    by usin Eqn.(16)
28 u2=(m^0.5)*uo; //Superficial gas velocity by using
    Eqn.(17)
29 //OUTPUT
30 printf('\nFor the model use');
31 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));
32 mprintf('\nFluidizing gas: ambient air at %fatm',P);
33 mprintf('\nSolids: \tzirconia , Average particle size
    =%fmicrometers ',m*dpbar);

```

```

34 mprintf(' \nEntering gas:\tSuperficial velocity=%fcm/
    s ',u2);
35
36 //=====END OF PROGRAM

```

Scilab code Exa 6.4 Reactor Scale up for Geldart A Catalyst

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-6, Example 4, Page 159
4 //Title: Reactor Scale-up for Geldart A Catalyst
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 dtb=20;//ID of bench-scale reactor
11 dtp=1;//ID of pilot reactor
12 dpbar=52;//Average particle size in micrometer
13 ephsilonm=0.45;//Void fraction for fixed bed
14 ephsilonmf=0.50;//Void fraction at minimum
    fluidization condition
15 ephsilonmb=0.60;//Void fraction
16 uo=30;//Superficial gas velocity in cm/s
17 Lmb=2;//Length of fixed bed in m
18 umf=0.33;//Velocity at minimum fluidization
    condition in cm/s
19 umb=1;//Velocity at in cm/s

```

```

20 db=3; //Equilibrium bubble size in cm
21 g=9.80; //Acceleration due to gravity in m/s^2
22 pi=3.142857;
23
24 //CALCULATION
25 ubr=0.711*(g*db/100)^0.5; //Rise velocity of bubble
    using Eqn.(7)
26
27 //Bubble velocity for the bench unit
28 ubb1=1.55*((uo-umf)/100)+14.1*((db/100)+0.005))*((
    dtb/100)^0.32)+ubr; //Bubble velocity using Eqn
    .(11)
29 si=1;
30 ubb2=si*((uo-umf)/100)+(3.2*((dtb/100)^(1/3)))*ubr;
    //Bubble velocity using Eqn.(9)
31 ubb=(ubb1+ubb2)/2; //Average bubble velocity
32
33 //Bubble velocity for the pilot unit
34 ubp1=1.55*((uo-umf)/100)+14.1*((db/100)+0.005))*((
    dtp^0.32)+ubr; //Bubble velocity using Eqn.(11)
35 si=1;
36 ubp2=si*((uo-umf)/100)+(3.2*(dtp^(1/3)))*ubr; //
    Bubble velocity using Eqn.(9)
37 ubp=(ubp1+ubp2)/2; //Average bubble velocity
38
39 //Rise velocity of upflowing emulsion
40 ueb=ubb-ubr; //For the bench unit
41 uep=ubp-ubr; //For the pilot unit
42
43 //Scale-Up Alternative 1.
44 dteb=20; //Effective bubble diameter
45 dib=[5;10;15;20]; //Different outside diameters
46 n=length(dib);
47 i=1;
48 while i<=n
49     li(i)=sqrt(((pi*dib(i)*dteb)/4)+((pi/4)*(dib(i))
        ^2)); //Pitch using Eqn.(13)
50     i=i+1;

```

```

51 end
52
53 //Scale-Up Alternative 2.
54 Lmp=Lmb*(ubp/ubb); //Static bed height of commercial
    unit
55 dtep=100; //Effective bubble diameter
56 dip=[10;15;20;25]; //Different outside diameters
57 m=length(dip);
58 i=1;
59 while i<=m
60     lip(i)=sqrt(((pi*dip(i)*dtep)/4)+(pi/4)*dip(i));
        //Pitch using Eqn.(13)
61     i=i+1;
62 end
63
64 //Height of Bubbling beds
65 //For bench unit
66 deltab=((uo/100)-(umb/100))/(ubb-(umb/100)); //
    Fraction of bed in bubbles using Eqn.(28)
67 ephsilonfb=deltab+(1-deltab)*ephsilonmb; //Void
    fraction of bubbling bed using Eqn.(20)
68 Lfb=Lmb*(1-ephsilonm)/(1-ephsilonfb); //Hieght of
    bubbling bed usnig Eqn.(19)
69 //For pilot unit
70 deltap=((uo/100)-(umb/100))/(ubp-(umb/100)); //
    Fraction of bed in bubbles using Eqn.(28)
71 ephsilonfp=deltap+(1-deltap)*ephsilonmb; //Void
    fraction of bubbling bed using Eqn.(20)
72 Lfp=Lmp*(1-ephsilonm)/(1-ephsilonfp); //Hieght of
    bubbling bed usnig Eqn.(19)
73
74 //OUTPUT
75 mprintf('\nRise velocity of bubble=%fm/s',ubr);
76 printf('\nFor the bench unit');
77 mprintf('\n\tWith Eqn.(11), Rise velocity=%fm/s',
    ubb1);
78 mprintf('\n\tWith Werthers procedure, Rise velocity=
    %fm/s',ubb2);

```

```

79 mprintf( '\n\tAverage rise velocity=%fm/s',ubb);
80 mprintf( '\n\tRise velocity of upflowing emulsion=%fm
/s',ueb);
81 printf( '\nFor the pilot unit');
82 mprintf( '\n\tWith Eqn.(11), Rise velocity=%fm/s',
ubp1);
83 mprintf( '\n\tWith Werthers procedure, Rise velocity=
%fm/s',ubp2);
84 mprintf( '\n\tAverage rise velocity=%fm/s',ubp);
85 mprintf( '\n\tRise velocity of upflowing emulsion=%fm
/s',uep);
86 printf( '\nScale-Up Alternative 1. ');
87 printf( '\n\tOuter diameter of tube(cm)');
88 printf( '\tPitch(cm)');
89 n=length(dib);
90 j=1;
91 while j<=n
92     mprintf( '\n\t\t%f',dib(j));
93     mprintf( '\t\t\t%f',li(j));
94     j=j+1;
95 end
96 printf( '\n\tSuitable arrangement');
97 mprintf( '\n\t\tOuter Diameter=%fcm\tPitch:Diameter
ratio=%f',dib(2),(li(2)/dib(2)));
98 printf( '\nScale-Up Alternative 2. ');
99 mprintf( '\n\tStatic bed height for commercial unit=
%fm',Lmp);
100 printf( '\n\tOuter diameter of tube(cm)');
101 printf( '\tPitch(cm)');
102 n=length(dip);
103 j=1;
104 while j<=n
105     mprintf( '\n\t\t%f',dip(j));
106     mprintf( '\t\t\t%f',lip(j));
107     j=j+1;
108 end
109 printf( '\n\tSuitable arrangement');
110 mprintf( '\n\t\tOuter Diameter=%fcm\tPitch:Diameter

```



```

13 ephsilonmf=0.50; //Void fraction at minimum
    fluidization condition
14 ephsilonmb=0.50; //Void fraction
15 uo=30; //Superficial gas velocity in cm/s
16 Lmb=2; //Length of fixed bed in m
17 umf=3; //Velocity at minimum fluidization condition
    in cm/s
18 umb=3; //Velocity at in cm/s
19 g=9.80; //Acceleration due to gravity in m/s^2
20 pi=3.142857;
21
22 //CALCULATION
23 //In the small bench unit
24 c=1;
25 ubb=c*((uo-umf)/100)+0.35*(g*(dtb/100))^0.5; //
    Velocity using Eqn.(5.22)
26 zsb=60*(dtb)^0.175; //Height using Eqn.(5.24)
27
28 //In the large pilot unit
29 ubp=c*((uo-umf)/100)+0.35*(g*dtp)^0.5; //Velocity
    using Eqn.(5.22)
30 zsp=60*(dtp*100)^0.175; //Height using Eqn.(5.24)
31
32 //OUTPUT
33 printf('\nCondition at which bubbles transform into
    slugs ');
34 mprintf('\nFor tha small bench unit\n\t\tVelocity=
    %fm/s\n\t\tHeight above distributor plate=%fm',
    ubb,zsb/100);
35 mprintf('\nFor tha large pilot unit\n\t\tVelocity=
    %fm/s\n\t\tHeight above distributor plate=%fm',
    ubp,zsp/100);
36
37 //=====END OF PROGRAM

```

Chapter 7

Entrainment and Elutriation from Fluidized Beds

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

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-7, Example 1, Page 179
4 //Title: Entrainment from Fine Particle Beds with
   High Freeboard
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 rhog=5.51;//Density of gas in kg/m^3
11 rhos=1200;//Density of solid in kg/m^3
12 dpbar=130;//Average size of particles in micrometer
13 uo=0.61;//Superficial gas velocity in m/s
```

```

14 g=9.80; // Acceleration due to gravity in m/s^2
15
16 //CALCULATION
17 //Assuming that freeboard is higher than TDH,
    computation of entrainment rate by Zenz & Weil's
    method
18 x=(uo^2)/(g*(dpbar*10^-6)*rhos^2); // Calculation of
    value of x-axis for Fig.(6), page 175
19 y=1.2; // Value of y-axis from Fig.(6)
20 Gsstar=y*rhog*uo; // Computation of rate of
    entrainment
21
22 //OUTPUT
23 mprintf('\nRate of entrainment=%fkg/m^2s',Gsstar);
24
25 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering (II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-7, Example 2, Page 180
4 //Title: Entrainment from Large Particle Beds with
    High Freeboard
5 //

```

```

6 clear all
7 clc

```

```

8
9 //INPUT
10 x=0.2;//Fraction of fines in the bed
11 Gstar=4.033320//Rate of entrainment in kg/m^2s(from
    Exa.1)
12
13 //CALCULATION
14 Gstar1=x*Gstar;//Rate of entrainment by Eqn.(3)
15
16 //OUTPUT
17 mprintf('\nRate of entrainment=%fkg/m^2s',Gstar1);
18
19 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-7, Example 3, Page 181
4 //Title: Entrainment from Beds with a Wide Size
    Distribution of Solids
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 rhog=5.51;//Density of gas in kg/m^3

```

```

11 rhos=1200;//Density of solid in kg/m^3
12 uo=0.61;//Superficial gas velocity in m/s
13 g=9.80;//Acceleration due to gravity in m/s^2
14 dp=[10;30;50;70;90;110;130]);//Diameter of particle
    in micrometer
15 p=[0;0.0110;0.0179;0.0130;0.0058;0.0020;0];
16 pi=3.142857;
17 dt=6;
18
19 //CALCULATION
20 n=length(dp);
21 i=1;
22 while i<=n
23     x(i)=(uo^2)/(g*(dp(i)*10^-6)*rhos^2);//
        Computation of value of x-axis for Fig.(6),
        page 175)
24     i=i+1;
25 end
26 y=[40;12;6;3.2;2.;1.3;1]);//Value of y-axis
    corresponding to each value of x-axis
27 y1 = y .* p;
28 i=1;
29 k=0;
30 while i<n
31     y1(i)=(y(i)*p(i));
32     k=k+((0.5)*(dp(i+1)-dp(i))*(y1(i+1)+y1(i)));//
        Integration using Trapezoidal rule
33     i=i+1;
34 end
35 rhosbar=k*rhog;//Computation of solid loading
36 te=(pi/4)*(dt^2)*rhosbar*uo;//Computation of total
    entrainment
37
38 //OUTPUT
39 mprintf('\nSolid loading =%fkg/m^3',rhosbar);
40 mprintf('\nTotal Entrainment =%fkg/s',te);
41
42 //=====END OF PROGRAM

```

Scilab code Exa 7.4 kstar from steady state experiments

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-7, Example 4, Page 181
4 //Title: k* from Steady State Experiments
5 //
6 clear all
7 clc
8
9 //INPUT
10 dp=[40;60;80;100;120]; //Diameter of particle in
   micrometer
11 uo=0.381; //Superficial gas velocity in m/s
12
13 //CALCULATION
14 Gs=0.9; //Rate of entrainment in kg/m^2 s from Fig.3(
   a)
15 pb=(1/100)*[0.45;1.00;1.25;1.00;0.60]; //Size
   distribution for bed particles from Fig.3(b)
16 pe=(1/100)*[1.20;2.00;1.25;0.45;0.10]; //Size
   distribution for entrained particles from Fig.3(b
   )
17 n=length(dp);
18 i=1;
19 while i<=n
20     ki(i)=(Gs*pe(i))/pb(i); //Calculation of ki*
```

```

                using Eqn.(13)
21         i=i+1;
22     end
23
24     //OUTPUT
25     printf( '\ndpi(micrometer) ');
26     printf( '\t100pb(dpi)(micrometer^-1) ');
27     printf( '\t100pe(dpi)(micrometer^-1) ');
28     printf( '\tki*(kg/m^2 s) ');
29     j=1;
30     while j<=n
31         mprintf( '\n%f', dp(j));
32         mprintf( '\t%f', 100*pb(j));
33         mprintf( '\t\t\t%f', 100*pe(j));
34         mprintf( '\t\t\t%f', ki(j));
35         j=j+1;
36     end
37
38     //=====END OF PROGRAM

```

Scilab code Exa 7.5 Comparing predictions for kstar

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-7, Example 5, Page 181
4 //Title: Comparing Predictions for k*
5 //

```

```

6 clear all

```

```

7  clc
8
9  //INPUT
10 rhog=1.217; //Density of gas in kg/m^3
11 myu=1.8E-5; //Viscosity of gas in kg/m s
12 umf=0.11; //Velocity at minimum fluidization
    condition in m/s
13 rhos=2000; //Density of solid in kg/m^3
14 uo=1.0; //Superficial gas velocity in m/s
15 g=9.80; //Acceleration due to gravity in m/s^2
16 dp=[30;40;50;60;80;100;120]; //Diameter of particle
    in micrometer
17 uti=[0.066;0.115;0.175;0.240;0.385;0.555;1.0]; //
    Terminal velocity of particles in m/s
18
19 //CALCULATION
20 n=length(dp);
21 i=1;
22 while i<=n
23     //Using Yagi & Aochi's correlation
24     Ret(i)=(rhog*(uti(i))*dp(i)*10^-6)/myu;
25     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)));
26     //Using Wen & Hasinger's correlation
27     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;
28     //Using Merrick & Highley's correlation
29     kistar3(i)=uo*rhog*(0.0001+130*exp(-10.4*((uti(i)
        )/uo)^0.5)*((umf/(uo-umf))^0.25)));
30     //Using Geldart's correlation
31     kistar4(i)=23.7*uo*rhog*exp(-5.4*(uti(i)/uo));
32     //Using Zenz & Weil's procedure
33     x1(i)=(uo^2)/(g*(dp(i)*10^-6)*rhos^2); //
        Computation of value of x-axis for Fig.(6),
        page 175)
34     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
35     kistar5(i)=y1(i)*rhog*uo;
36     //Using Gugnoni & Zenz's procedure
37     x2(i)=(uo-uti(i))/((g*dp(i)*10^-6)^0.5);//
        Computation of value of x-axis for Fig.(6),
        page 175)
38     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
39     kistar6(i)=y(i)*rhog*uo;
40     i=i+1;
41 end
42
43 i=1;
44 printf('dp(micrometer)');
45 printf('\tYagi & Aochi');
46 printf('\tWen & Hashinger');
47 printf('\t\tMerrick & Highley');
48 printf('\tGeldart et al. ');
49 printf('\t\tZenz & Well');
50 printf('\t\tGugnoni & Zenz');
51 while i<=n
52     mprintf('\n%f',dp(i));
53     mprintf('\t%f',kistar1(i));
54     mprintf('\t%f',kistar2(i));
55     mprintf('\t\t%f',kistar3(i));
56     mprintf('\t\t%f',kistar4(i));
57     mprintf('\t\t%f',kistar5(i));
58     mprintf('\t\t%f',kistar6(i));
59     i=i+1;
60 end
61
62 //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.
63 //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.
64
65 //=====END OF PROGRAM

```

Scilab code Exa 7.6 Entrainment from a short vessel

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering (II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-7, Example 6, Page 190
4 //Title: Entrainment from a Short Vessel Ht<TDH
5 //

```

```

6 clear all
7 clc
8
9 //INPUT
10 dpbar=60; //Average size of particles in micrometer
11 rhog=1.3; //Density of gas in kg/m^3
12 rhos=1500; //Density of solid in kg/m^3
13 umf=0.003; //Velocity at minimum fluidization
  condition in m/s
14 uo=0.503; //Superficial gas velocity in m/s
15 g=9.80; //Acceleration due to gravity in m/s^2
16 Hf=2; //Height at which the cyclone inlet is to be
  located in m
17
18 //CALCULATION
19 y=(uo^2)/(g*(dpbar*10^-3)*rhos^2); //Calculation of
  value of y-axis for Fig.(6), page 175

```

```

20 x=1; //Value of x-axis from Fig.(6), page 175
21 Gsstar=x*rhog*uo; //Computation of rate of
    entrainment
22 Gsuo=5.0; //Ejection rate pf particles in kg/m^2 s
    from Fig.(11), page 188
23 a=0.72/uo; //From Fig.(12), page 189
24 Gs=Gsstar+(Gsuo-Gsstar)*exp(-a*Hf);
25 p=((Gs-Gsstar)/Gsstar)*100;
26
27 //OUTPUT
28 mprintf('\nRate of entrainment from short bed=%fkg/m
    ^2s',Gs);
29 mprintf('\nThis entrainment is %f percent higher
    than it would be if the gas exit were at the TDH'
    ,p);
30
31 //=====END OF PROGRAM

```

Chapter 8

High velocity Fluidization

Scilab code Exa 8.1 Performance of a Fast Fluidized Vessel

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-8, Example 1, Page 206
4 //Title: Performance of a Fast Fluidized Vessel
5 //


---


6 clear all
7 clc
8
9 //INPUT
10 Lmf=2.4;//Length of bed at minimum fluidized
   condition in m
11 uo=[2;4;6];//Superficial gas velocity in m/s
12 GsII=100;//Solid circulation rate in kg/m^2 s for
   Mode II
13 uoIII=4;//Superficial gas velocity in m/s for Mode
   III
14 GsIII=[42;50;100;200;400];//Solid circulation rate
```

```

    in kg/m2 s for Mode III
15 GsIV=[70;100;120]; //Solid circulation rate in kg/m2
    s for Mode IV
16 dt=0.4; //Column diameter in m
17 Ht=10; //Height of column in m
18 rhos=1000; //Density of solid in kg/m3
19 dpbar=55; //Particle diameter in micrometer
20 ephsilonmf=0.5; //Void fraction at minimum
    fluidization condition
21
22 //CALCULATION
23 //Mode I
24 ephsilonstar=0.01; //Saturation carrying capacity of
    gas
25 ephsilonsd=[0.2;0.16;0.14]; //Solid holdup in lower
    dense region from Fig.8(b) for various uo
26 n=length(uo);
27 i=1;
28 Hfguess=2; //Guess value of height
29 while i<=n
30     a(i)=3/uo(i); //Decay constant
31     function [fn]=solver_func(Hf) //Function defined
        for solving the system
32         fn=Lmf*(1-ephsilonmf)-((ephsilonsd(i)-(
            ephsilonstar+(ephsilonsd(i)-ephsilonstar)
            *exp(-a(i)*Hf)))/a(i))-Ht*ephsilonsd(i)+
            Hf*(ephsilonsd(i)-ephsilonstar);
33     endfunction
34     [Hf(i)]=fsolve(Hfguess,solver_func,1E-6); //Using
        inbuilt function fsolve for solving Eqn.(10)
        for Hf
35     Hd(i)=Ht-Hf(i); //Height of lower dense region
36     ephsilonse(i)=ephsilonstar+(ephsilonsd(i)-
        ephsilonstar)*exp(-a(i)*Hf(i)); //Solid holdup
        at exit
37     GsI(i)=rhos*uo(i)*ephsilonse(i); //Solid
        circulation rate from Eqn.(4)
38     i=i+1;

```

```

39 end
40
41 //Mode II
42 i=1;
43 Hfguess2=2;//Guess value of height
44 while i<=n
45     ephsilonseII(i)=GsII/(rhos*uo(i));//Solid holdup
        at exit
46     function [fn]=solver_func1(Hf)//Function defined
        for solving the system
47     fn=ephsilonseII(i)-ephsilonstar-(ephsilonsd(
        i)-ephsilonstar)*exp(-a(i)*Hf);//From Eqn
        .(7)
48     endfunction
49     [HfII(i)]=fsolve(Hfguess2,solver_func1,1E-6);//
        Using inbuilt function fsolve for solving Eqn
        .(10) for Hf
50     HdII(i)=Ht-HfII(i);//Height of lower dense
        region
51     //Length of bed minimum fluidization condtion
52     LmfII(i)=(1-ephsilonmf)^-1*[((ephsilonsd(i)-
        ephsilonseII(i))/a(i))+Ht*ephsilonsd(i)-HfII(
        i)*(ephsilonsd(i)-ephsilonstar)];
53     i=i+1;
54 end
55
56 //Mode III
57 aIII=3/uoIII;//Decay constant
58 ephsilonsdIII=0.16;//Solid holdup at lower dense
        region
59 i=1;
60 m=length(GsIII);
61 Hfguess3=2;//Guess value of height
62 while i<=m
63     ephsilonseIII(i)=GsIII(i)/(rhos*uoIII);//Solid
        holdup at exit
64     function [fn]=solver_func2(Hf)//Function defined
        for solving the system

```

```

65         fn=ephsilonseIII(i)-ephsilonstar-(
           ephsilonsdIII-ephsilonstar)*exp(-aIII*Hf)
           ;//From Eqn.(7)
66     endfunction
67     [HfIII(i)]=fsolve(Hfguess3,solver_func2,1E-6);//
           Using inbuilt function fsolve for solving Eqn
           .(10) for Hf
68     HdIII(i)=Ht-HfIII(i);//Height of lower dense
           region
69     //Length of bed at minimum fluidization
           condition
70     LmfIII(i)=(1-ephsilonmf)^-1*[((ephsilonsdIII-
           ephsilonsdIII(i))/aIII)+Ht*ephsilonsdIII-
           HfIII(i)*(ephsilonsdIII-ephsilonstar)];
71     i=i+1;
72 end
73
74 //Mode IV
75 i=1;
76 Hfguess4=2;//Guess value of height
77 while i<=n
78     aIV(i)=3/uo(i);//Decay constant
79     ephsilonsdIV(i)=GsIV(i)/(rhos*uo(i));//Solid
           holdup at exit
80     function [fn]=solver_func3(Hf)//Function defined
           for solving the system
81         fn=ephsilonseIV(i)-ephsilonstar-(ephsilonsd(
           i)-ephsilonstar)*exp(-aIV(i)*Hf);//From
           Eqn.(7)
82     endfunction
83     [HfIV(i)]=fsolve(Hfguess4,solver_func3,1E-6);//
           Using inbuilt function fsolve for solving Eqn
           .(10) for Hf
84     HdIV(i)=Ht-HfIV(i);//Height of lower dense
           region
85     //Length of bed at minimum fluidization
           condition
86     LmfIV(i)=(1-ephsilonmf)^-1*[((ephsilonsd(i)-

```

```

            ephsilonseIV(i))/aIV(i))+Ht*ephsilonsd(i)-
            HfIV(i)*(ephsilonsd(i)-ephsilonstar)];
87     i=i+1;
88 end
89
90 //OUTPUT
91 printf( '\nMode I ');
92 printf( '\n\tuo(m/s)\t\ttephsilonse(-)\tHf(m)\t\tHd(m)
\t\tGs(kg/m^2 s) ');
93 i=1;
94 while i<=n
95     mprintf( '\n\t%f\t%f\t%f\t%f\t%f',uo(i),
        ephsilonse(i),Hf(i),Hd(i),GsI(i));
96     i=i+1;
97 end
98 printf( '\nMode II ');
99 printf( '\n\tuo(m/s)\t\ttephsilonse(-)\tHf(m)\t\tHd(m)
\t\tLmf(m) ');
100 i=1;
101 while i<=n
102     mprintf( '\n\t%f\t%f\t%f\t%f\t%f',uo(i),
        ephsilonseII(i),HfII(i),HdII(i),LmfII(i));
103     i=i+1;
104 end
105 printf( '\nMode III ');
106 printf( '\n\tGs(kg/m^ s)\ttephsilonse(-)\tHf(m)\t\tHd(
m)\t\tLmf(m) ');
107 i=1;
108 while i<=m
109     mprintf( '\n\t%f\t%f\t%f\t%f\t%f',GsIII(i),
        ephsilonseIII(i),HfIII(i),HdIII(i),LmfIII(i))
        ;
110     i=i+1;
111 end
112 printf( '\nMode IV ');
113 printf( '\n\tuo(m/s)\t\tGs(kg/m^2 s)\ttephsilonse(-)\t
\tHf(m)\t\tLmf(m) ');
114 i=1;

```



```
115 while i<=n
116     mprintf( '\n\t%f\t%f\t%f\t%f\t%f',uo(i),GsIV(i),
               ephsilonseIV(i),HfIV(i),LmfIV(i));
117     i=i+1;
118 end
119
120 //=====END OF PROGRAM
```

Chapter 9

Solid Movement Mixing Segregation and Staging

Scilab code Exa 9.1 Vertical Movement of Solids

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-9, Example 1, Page 218
4 //Title: Vertical Movement of Solids
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 umf=0.015; //Velocity at minimum fluidization
   condition in m/s
12 ephsilonmf=0.5; //Void fraction at minimum
   fluidization condition
13 uo=0.1; //Superficial gas velocity in m/s
```

```

14 delta=0.2; //Bed fraction in bubbles
15 db=0.06; //Equilibrium bubble size in m
16 dt=[0.1;0.3;0.6;1.5]; //Various vessel sizes in m
17 ub=[0.4;0.75;0.85;1.1]; //Bubble velocity in m/s
18 Dsv=[0.03;0.11;0.14;0.23]; //Reported values of
    vertical dispersion coefficient
19
20 //CALCULATION
21 n=length(ub);
22 i=1;
23 fw1=2; //Wake fraction from Hamilton et al.
24 fw2=0.32; //Wake fraction from Fig.(5.8)
25 fw=(fw1+fw2)*0.5; //Average value of wake fraction
26 while i<=n
27     Dsv1(i)=12*((uo*100)^0.5)*((dt(i)*100)^0.9); //
        Vertical distribution coefficient from Eqn
        .(3)
28     Dsv2(i)=(fw^2*ephsilonmf*delta*db*ub(i)^2)/(3*
        umf); //Vertical distribution coefficient from
        Eqn.(12)
29     i=i+1;
30 end
31
32 //OUTPUT
33 printf('\n\t\tVertical dispersion coefficient (m^2/s)
    ');
34 printf('\nVessel Size(m) ');
35 printf('\tFrom Experiment ');
36 printf('\tFrom Eqn.(3) ');
37 printf('\tFrom Eqn.(12) ');
38 i=1;
39 while i<=n
40     mprintf('\n%f',dt(i));
41     mprintf('\t%f',Dsv(i));
42     mprintf('\t%f',Dsv1(i)/10^4);
43     mprintf('\t%f',Dsv2(i));
44     i=i+1;
45 end

```

46

47 //=====END OF PROGRAM

Scilab code Exa 9.2 Horizontal Drift Of Solids

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-9, Example 2, Page 222
4 //Title: Horizontal Drift Of Solids
5 //
6
7 clear all
8 clc
9
10 //INPUT
11 Lmf=0.83;//Length of bed at minimum fluidization
   condition in m
12 dp=450;//Average particle size in micrometer
13 ephsilonmf=0.42;//Void fraction at minimum
   fluidization condition
14 umf=0.17;//Velocity at minimum fluidization
   condition in m/s
15 uo=[0.37;0.47;0.57;0.67];//Superficial gas velocity
   in m/s
16 Dsh=[0.0012;0.0018;0.0021;0.0025];//Horizontal Drift
   Coefficient from Experiment in m^2/s
17 db=[0.10;0.14];//Equilibrium bubble size in m
18 g=9.81;//Acceleration due to gravity in m/s^2
```

```

19
20
21 //CALCULATION
22 n=length(uo);
23 m=length(db);
24 j=1;
25 i=1;
26 k=1;
27 alpha=0.77;//Since we are not dealing with Geldart A
    or AB solids
28 uf=umf/epsilonmf;
29 for j = 1:m
30     for i = 1:n
31         ubr(k)=0.711*(db(j)*g)^0.5;//Rise
            velocity of a single bubble in m/s
32         ub(k)=uo(i)-umf+ubr(k);//Rise velocity
            of bubbles in a bubbling bed
33         delta(k)=(uo(i)-umf)/(ub(k)+umf);//Bed
            fraction in bubbles
34         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)
35         else Dsh(k)=(3/16)*(delta/(1-delta))*
            (alpha^2*umf*db/epsilonmf);//
            Horizontal Distribution coeff. from
            Eqn.(15)
36         end
37         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)
38         i=i+1;
39         k=k+1;
40     end
41     i=1;

```

```

42     j=j+1;
43 end
44
45 //OUTPUT
46 i=1;
47 j=1;
48 k=1;
49 while k<=m*n
50     mprintf('\nSnce we do not have ub=%fm/s>>uf=%fm/
           s we use Eqn.(14). ',ub(k),uf)
51     printf('\nGas Velocity(m/s) ');
52     printf('\tHorizontal Drift Coefficient
           Calculated(m^2/s) ');
53     printf('\tHorizontal Drift Coefficient from
           Experiment(m^2/s) ');
54     while j<=m
55         mprintf('\ndb=%fm',db(j));
56         while i<=n
57             mprintf('\n%f',uo(i));
58             mprintf('\t\t%f',Dshc(k));
59             mprintf('\t\t\t\t\t%f',Dsh(i));
60             i=i+1;
61             k=k+1;
62         end
63         i=1;
64         j=j+1;
65     end
66 end
67
68 //=====END OF PROGRAM

```

Scilab code Exa 9.3 Design of Baffle Plates

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-9, Example 3, Page 232
4 //Title: Design of Baffle Plates
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 Gsup=1.5; //Solid interchange rate in kg/m^2plate s
12 dor=19.1; //Orifice diameter in mm
13 dp=210; //Particle size in micrometer
14 uo=0.4; //Superficial gas velocity in m/s
15 fopen=[0.12;0.17;0.26]; //Open area fraction
16 pi=3.14;
17
18 //CALCULATION
19 n=length(fopen);
20 i=1;
21 while i<=n
22     uor(i)=uo/fopen(i); //Gas velocity through the
        orifice
23     ls1(i)=Gsup/fopen(i); //Flux of solids through
        the holes
24     i=i+1;
25 end
26 ls2=[12;20;25]; //Flux of solids through holes from
        Fig.13(c) for different uor values
27 fopen1=0.12; //Open area fraction which gives
        reasonable fit
28 lor=sqrt(((pi/4)*dor^2)/fopen1); //Orifice spacing
29
30 //OUTPUT

```

```

31 printf( '\n fopen ');
32 printf( '\t\t uor(m/s) ');
33 printf( '\t ls from Eqn.(18) ');
34 printf( '\t ls from Fig.13(c) ');
35 i=1;
36 while i<=n
37     mprintf( '\n%f', fopen(i));
38     mprintf( '\t%f', uor(i));
39     mprintf( '\t%f', ls1(i));
40     mprintf( '\t\t%f', ls2(i));
41     i=i+1;
42 end
43 mprintf( '\n\n For square pitch, the orifice spacing
         should be %fmm', lor);
44
45 //=====END OF PROGRAM

```

Chapter 10

Gas Dispersion and Gas Interchange in Bubbling Beds

Scilab code Exa 10.1 Estimate Interchange Coefficients in Bubbling Beds

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-10, Example 1, Page 253
4 //Title: Estimate Interchange Coefficients in
   Bubbling Beds
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 umf=[0.01;0.045]; //Velocity at minimum fluidization
   condition in m/s
12 ephsilomf=[0.5;0.5]; //Void fraction at minimum
   fluidization condition
```

```

13 D=[2E-5;7E-5]; // Diffusion coefficient of gas in m^2/
    s
14 g=9.81; // Acceleration due to gravity in m/s^2
15
16 //CALCULATION
17 db=[5;10;15;20];
18 n=length(umf);
19 m=length(db)';
20 for i = 1:n
21     for j = 1:m
22         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)
23         Kce(i,j)=6.77*((D(i)*epsilonumf(i)
                *0.711*(g*db(j))^0.5)/db(j)^3)^0.5; //
                Gas interchange coefficient between
                emulsion and cloud from Eqn.(34)
24         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)
25     end;
26 end
27
28 //OUTPUT
29 i=1;
30 j=1;
31 k=1;
32 while k<=m*n
33     printf('\n\t\tKbc for fine particles and He');
34     printf('\t\tKbc for coarse particles and ozone');
35     printf('\t\tKbe for fine particles and He');
36     printf('\t\tKbe for coarse particles and ozone');
37     while j<=m
38         mprintf('\t\tndb=%fm', db(j)*10^-2);
39         while i<=n
40             mprintf('\t\t%f', Kbc(k));

```

```

41         mprintf( '\t\t\t%f', Kbe(k));
42         i=i+1;
43         k=k+1;
44         printf( '\t\t\t');
45     end
46     i=1;
47     j=j+1;
48     end
49 end
50 Kbe=Kbe';
51 Kbc=Kbc';
52 plot2d("ll",db,[Kbc Kbe]);
53 xtitle('Plot of Kbc,Kbe vs db','db',['Kbc','Kbe']);
54 printf('\nComparing the points with the plot of Kbc,
        Kbe vs db in Fig.(12), we can conclude the
        following:');
55 printf('\nKbc for fine particles and helium: line 2
        in Fig.(12)');
56 printf('\nKbc for coarser particles and ozone: line
        3 in Fig.(12)');
57 printf('\nKbe for fine particles and helium: line 4
        in Fig.(12)');
58 printf('\nKbe for coarser particles and ozone: line
        5 in Fig.(12)');
59
60 //=====END OF PROGRAM

```

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

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

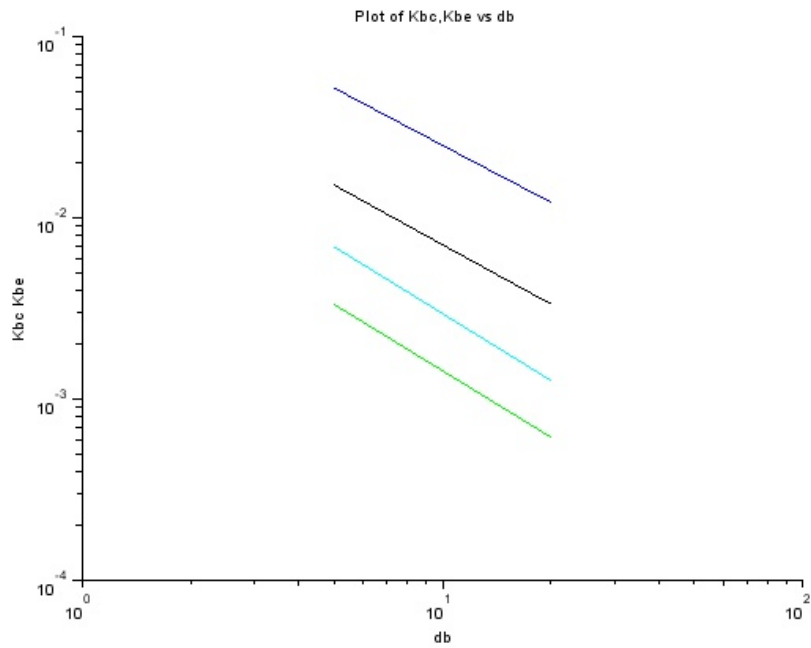


Figure 10.1: Estimate Interchange Coefficients in Bubbling Beds

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

```
2
3 //Chapter –10, Example 2, Page 254
4 //Title: Compare the Relative Importance of Kbc and
      Kce
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 D=0.69; //Diffusion coefficient of gas in cm2/s
12 umf=1.0; //Velocity at minimum fluidization condition
      in cm/s
13 ephsilonmf=0.5; //Void fraction at minimum
      fluidization condition
14 db=[5;15]; //Equilibrium bubble size in cm
15 g=980; //Acceleration due to gravity in cm/s2
16
17 //CALCULATION
18 n=length(db);
19 i=1;
20 while i<=n
21     Kbc(i)=4.5*(umf/db(i))+5.85*((D0.5*g0.25)/db(i)
      )(5/4)); //Gas interchange coefficient
      between bubble and cloud from Eqn.(27)
22     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)
23     Kbe(i)=(Kbc(i)*Kce(i))/(Kbc(i)+Kce(i)); //Gas
      interchange coefficient between bubble and
      emulsion from Eqn.(14)
24     e(i)=(Kce(i)-Kbe(i))/Kbe(i); //Error when minor
      resistance is ignored
25     i=i+1;
```

```

26 end
27
28 //OUTPUT
29 printf( '\ndb(cm) ');
30 printf( '\t\tCalculated Kbc ');
31 printf( '\tCalculated Kce ');
32 printf( '\t\tKbe from Eqn.(14) ');
33 printf( '\tError when minor resistance is ignored (in
    percentage) ');
34 i=1;
35 while i<=n
36     mprintf( '\n%f ', db(i));
37     mprintf( '\t%f ', Kbc(i));
38     mprintf( '\t%f ', Kce(i));
39     mprintf( '\t\t%f ', Kbe(i));
40     mprintf( '\t\t%f ', e(i)*100);
41     i=i+1;
42 end
43
44 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-10, Example 3, Page 255
4 //Title: Compare Interchange Rates for Adsorbed and
    Nonadsorbed Gases
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 Kbe=[0.028;0.05]; //Reported range for gas
    interchange coefficient between bubble and
    emulsion
12 uo=0.30; //Superficial gas velocity in m/s
13 db=0.13; //Equilibrium bubble size in m
14 m=7;
15 ephsilonmf=0.5; //Void fraction at minimum
    fluidization condition
16 umf=0.0018; //Velocity at minimum fluidization
    condition in m/s
17 D=[9E-6;22E-6]; //Diffusion coefficient of gas in m
    ^2/s
18 g=9.81; //Acceleration due to gravity in m/s^2
19
20 //CALCULATION
21 n=length(Kbe);
22 i=1;
23 while i<=n
24     Kbem(i)=(6/db)*Kbe(i); //Gas interchange
        coefficient between bubble and emulsion from
        Eqn.(19)
25     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)
26     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)
27     Kbe(i)=(Kbc(i)*Kce(i))/(Kbc(i)+Kce(i)); //Gas
        interchange coefficient between bubble and
        emulsion from Eqn.(14)
28     c(i)=(Kbem(i)/Kbe(i));

```

```

29     i=i+1;
30 end
31
32 //OUTPUT
33 printf( '\nKbe from Eqn.(19) ');
34 printf( '\tKbc from Eqn.(27) ');
35 printf( '\tKce from Eqn.(34) ');
36 printf( '\tKbe from Eqn.(14) ');
37 printf( '\tComparison of Kbe from Eqn.(19) and that
    from Eqn.(14) ');
38 i=1;
39 while i<=n
40     mprintf( '\n%f', Kbem(i));
41     mprintf( '\t\t%f', Kbc(i));
42     mprintf( '\t\t%f', Kce(i));
43     mprintf( '\t\t%f', Kbe(i));
44     mprintf( '\t\t%f', c(i));
45     i=i+1;
46 end
47
48 //=====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

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-11, Example 1, Page 265
4 //Title: Fitting Reported Mass Transfer Data with
   the Bubbling Bed Model
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 db=0.37;//Equilibrium bubble size in cm
12 dp=0.028;//Particle size in cm
13 rhos=1.06;//Density of solids in g/cc
```

```

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

27
28 //CALCULATION
29 n=length(uo);
30 i=1;
31 Rept=(dp*ut*rhog)/myu;
32 Shstar=2+(0.6*(Rept^0.5)*(Sc^(1/3)));//Sherwood no.
    from Eqn.(1)
33 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)
34 ubr=0.711*(g*db)^0.5;//Rise velocity of the bubble
35 while i<=n
36     x(i)=(uo(i)-umf)/(ubr*(1-ephsilonmf));//The term
        delta/(1-epshilonf) after simplification
37     Shbed(i)=x(i)*[(gammab*Shstar*etad)+((phis*dp^2*
        y)/(6*D))*Kbc];//Sherwood no. from Eqn.(11)
38     Rep(i)=(dp*uo(i)*rhog)/myu;//Reynolds of the
        particle
39     i=i+1;
40 end
41

```

```

42 //OUTPUT
43 printf('\n\nThe desired result is the relationship
      between Shbed and Rep The points gives a
      straight line of the form y=mx+c');
44 printf('\nRep');
45 printf('\t\tShbed');
46 i=1;
47 while i<=n
48     printf('\n%f',Rep(i));
49     printf('\t%f',Shbed(i));
50     i=i+1;
51 end
52 plot(Rep,Shbed);
53 xlabel("Rep");
54 ylabel("Shbed");
55
56 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering(II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-11, Example 2, Page 267
4 //Title: The Effect of m on Bubble-Emulsion
  Interchange
5 //

```

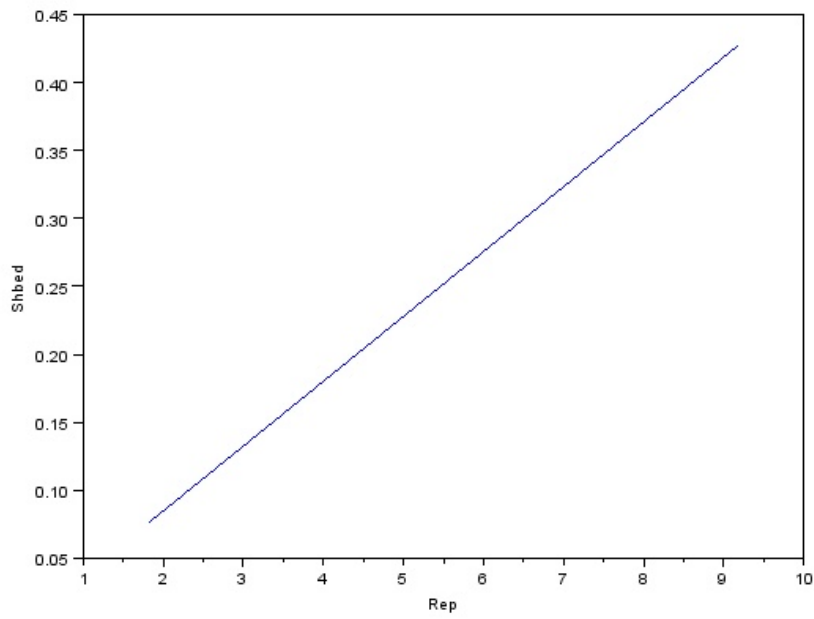


Figure 11.1: Fitting Reported Mass Transfer Data with the Bubbling Bed Model

```

6
7 clear all
8 clc
9
10 //INPUT
11 umf=0.12; //Velocity at minimum fluidization
    condition in cm/s
12 uo=40; //Superficial gas velocity in cm/s
13 ub=120; //Velocity of the bubble in cm/s
14 D=0.7; //Diffusion coefficient of gas in cm^2/s
15 abkbe1=1; //Bubble-emulsion interchange coefficient
    for non absorbing particles(m=0)
16 abkbe2=18; //Bubble-emulsion interchange coefficient
    for highly absorbing particles(m=infinity)
17 g=980; //Acceleration due to gravity in square cm/s^2
18
19 //CALCULATION
20 //For non absorbing particles m=0,etad=0
21 Kbc=(ub/uo)*(abkbe1);
22 dbguess=2; //Guess value of db
23 function [fn]=solver_func(db) //Function defined for
    solving the system
24     fn=abkbe1-(uo/ub)*(4.5*(umf/db)+5.85*(D^0.5*g
        ^0.25)/(db^(5/4))); //Eqn.(10.27)
25 endfunction
26 [d]=fsolve(dbguess,solver_func,1E-6); //Using inbuilt
    function fsolve for solving Eqn.(10.27) for db
27 //For highly absorbing particles m=infinity, etad=1
28 M=abkbe2-(uo/ub)*Kbc;
29 //For intermediate condition
30 alpha=100;
31 m=10;
32 etad=1/(1+(alpha/m)); //Fitted adsorption efficiency
    factor from Eqn.(23)
33 abkbe3=M*etad+(uo/ub)*Kbc;
34
35 //OUTPUT
36 mprintf('\nFor non absorbing particles:\n\tDiameter

```

```

    of bubble=%fcm\n\tBubble-cloud interchange
    coefficient=%fs^-1',d,Kbc);
37 mprintf('\nFor highly absorbing partilces:\n\tM=%f',
    M);
38 mprintf('\nFor intermediate condition:\n\tFitted
    adsorption efficiency factor:%f\n\tBubble-
    emulsion interchange coefficient:%fs^-1',etad,
    abkbe3);
39
40 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering (II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-11, Example 3, Page 273
4 //Title: Fitting Reported Heat Transfer Data with
    the Bubbling Bed Model
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 rhos=1.3;//Density of solids in g/cc
12 phis=0.806;//Sphericity of solids
13 gammab=0.001;//Ratio of volume of dispersed solids

```

```

    to that of bubble phase
14 rhog=1.18E-3; //Density of air in g/cc
15 Pr=0.69; //Prandtl number
16 myu=1.8E-4; //Viscosity of gas in g/cm s
17 Cpg=1.00; //Specific heat capacity of gas in J/g K
18 ephsilonmf=0.45; //Void fraction at minimum
    fluidization condition
19 kg=2.61E-4; //Thermal concuctivity of gas in W/cm k
20 dp=0.036; //Particle size in cm
21 umf=6.5; //Velocity at minimum fluidization condition
    in cm/s
22 ut=150; //Terminal velocity in cm/s
23 db=0.4; //Equilibrium bubble size in cm
24 etah=1; //Efficiency of heat transfer
25 uo=[10;20;30;40;50]; //Superficial gas velocity in cm
    /s
26 g=980; //Acceleration due to gravity in square cm/s^2
27
28 //CALCULATION
29 Nustar=2+[((dp*ut*rhog)/myu)^0.5*Pr^(1/3)]; //Nusselt
    no. from Eqn.(25)
30 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)
31 ubr=0.711*(g*db)^0.5; //Rise velocity of the bubble
    from Eqn.(6.7)
32 n=length(uo);
33 i=1;
34 while i<=n
35     x(i)=(uo(i)-umf)/(ubr*(1-ephsilonmf)); //The term
        delta/(1-ephsilonf) after simplification
36     Nubed(i)=x(i)*[gammab*Nustar*etah+(phis*dp^2/(6*
        kg))*Hbc]; //Nusselt no. from Eqn.(36)
37     Rep(i)=(dp*uo(i)*rhog)/myu; //Reynolds of the
        particle
38     i=i+1;
39 end
40

```

```

41 //OUTPUT
42 printf('\n\nThe desired result is the relationship
      between Nubed and Rep which is in the form of a
      straight line y=mx+c');
43 printf('\nRep');
44 printf('\t\tNubed');
45 i=1;
46 while i<=n
47     printf('\n%f',Rep(i));
48     printf('\t%f',Nubed(i));
49     i=i+1;
50 end
51 plot(Rep,Nubed);
52 xlabel("Rep");
53 ylabel("Nubed");
54
55 //=====END OF PROGRAM

```

Scilab code Exa 11.4 Heating a Particle in a Fluidized Bed

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering(II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-11, Example 4, Page 274
4 //Title: Heating a Particle in a Fluidized Bed
5 //

```

6

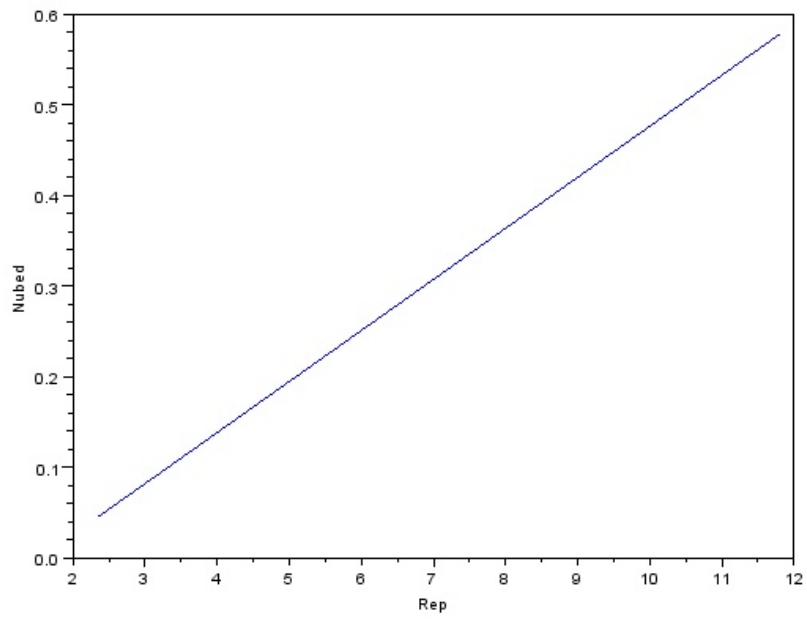


Figure 11.2: Fitting Reported Heat Transfer Data with the Bubbling Bed Model

```

7 clear all
8 clc
9
10 //INPUT
11 rhog=1.2; //Density of air in kg/m^3
12 myu=1.8E-5; //Viscosity of gas in kg/m s
13 kg=2.6E-2; //Thermal conductivity of gas in W/m k
14 dp=1E-4; //Particle size in m
15 rhos=8920; //Density of solids in kg/m^3
16 Cps=390; //Specific heat capacity of the solid in J/
    kg K
17 ephsilonf=0.5; //Void fraction of the fluidized bed
18 umf=0.1; //Velocity at minimum fluidization condition
    in m/s
19 uo=0.1; //Superficial gas velocity in m/s
20 pi=3.14
21
22 //CALCULATION
23 to=0; //Initial temperature of the bed
24 T=100; //Temperature of the bed
25 t=0.99*T; //Particle temperature i.e. when it
    approaches 1% of the bed temperature
26 mp=(pi/6)*dp^3*rhos; //Mass of the particle
27 A=pi*dp^2; //Surface area of the particle
28 Rep=(dp*uo*rhog)/myu; //Reynold's no. of the particle
29 Nubed=0.0178; //Nusselt no. from Fig.(6)
30 hbed1=(Nubed*kg)/dp; //Heat transfer coefficient of
    the bed
31 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)
32 hbed2=140*hbed1; //Since from Fig.(6) Nup is 140
    times Nubed
33 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)
34
35 //OUTPUT

```

```
36 printf('\nCase(a):Using the whole bed coefficient
    from Fig.(6) ');
37 mprintf('\n\tTime needed for the particle approach 1
    percentage of the bed temperature is %fs',t1);
38 printf('\nCase(b):Uisng the single-particle
    coefficient of Eqn.(25),also shown in Fig.(6) ');
39 mprintf('\n\tTime needed for the particle approach 1
    percentage of the bed temperature is %fs',t2);
40
41 //=====END OF PROGRAM
```

Chapter 12

Conversion of Gas in Catalytic Reactions

Scilab code Exa 12.1 Fine Particle Geldart A Bubbling Bed Reactor

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-12, Example 1, Page 293
4 //Title: Fine Particle (Geldart A) Bubbling Bed
   Reactor
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 Kr=10; //rate constant in m3 gas/m3 cat s
12 D=2E-5; //Diffusion coefficient of gas in m2/s
13 dpbar=68; //Average particle size in micrometers
14 epsilon=0.5; //Void fraction of fixed bed
```

```

15  gammab=0.005; //Ratio of volume of dispersed solids
    to that of bubble phase
16  ephsilonmf=0.55; //Void fraction at minimum
    fluidization condition
17  umf=0.006; //Velocity at minimum fluidization
    condition in m/s
18  db=0.04; //Equilibrium bubble size in m
19  Lm=0.7; //Length of the bed in m
20  uo=0.1; //Superficial gas velocity in m/s
21  dbed=0.26; //Diameter of the bed in m
22  g=9.81; //Acceleration due to gravity in square m/s^2
23
24  //CALCULATION
25  ubr=0.711*(g*db)^0.5; //Rise velocity of bubble from
    Eqn.(6.7)
26  ub=uo-umf+ubr; //Velocity of bubbles in bubbling beds
    in Eqn.(6.8)
27  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)
28  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)
29  delta=uo/ub; //Fraction of bed in bubbles from Eqn
    .(6.29)
30  fw=0.6; //Wake volume to bubble volume from Fig.(5.8)
31  gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw
    ); //Volume of solids in cloud to that of the
    bubble from Eqn.(6.36)
32  gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-
    gammac; //Volume of solids in emulsion to that of
    the bubble from Eqn.(6.35)
33  ephsilonf=1-(1-delta)*(1-ephsilonmf); //Void fraction
    of fixed bed from Eqn.(6.20)
34  Lf=(1-ephsilonm)*Lm/(1-ephsilonf); //Length of fixed
    bed from Eqn.(6.19)
35  Krtou=Kr*Lm*(1-ephsilonm)/uo; //Dimensionless
    reaction rate group from Eqn.(5)

```

```

36 Kf=gammab*Kr+1/(((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)
    +(1/(gammae*Kr)))))); //Reaction rate for fluidized
    bed from Eqn.(14)
37 XA=1-exp(-1*Kf*Lf/ub); //Conversion from Eqn.(16)
38
39 //OUTPUT
40 mprintf('\nThe dimnesionless reaction rate group: %f
    ',Krtou);
41 mprintf('\nThe reaction rate for fluidized bed: %fs
    ^-1',Kf);
42 mprintf('\nConversion: %f',XA);
43
44 //=====END OF PROGRAM

```

Scilab code Exa 12.2 Commercial Sized Phthalic Anhydride Reactor

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering (II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-12, Example 2, Page 298
4 //Title: Commercial-Sized Phthalic Anhydride Reactor
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 umf=0.005; //Velocity at minimum fluidization
    condition in m/s

```

```

12 ephsilonm=0.52;//Void fraction of fixed bed
13 ephsilonmf=0.57;//Void fraction at minimum
    fluidization condition
14 DA=8.1E-6;//Diffusion coefficient of gas in m2/s
15 DR=8.4E-6;//Diffusion coefficient of gas in m2/s
16 Lm=5;//Length of the bed in m
17 dte=1;//Diameter of tube in m
18 Kr1=1.5;//rate constant in m3 gas/m3 cat s
19 Kr3=0.01;//rate constant in m3 gas/m3 cat s
20 gammab=0.005;//Ratio of volume of dispersed solids
    to that of bubble phase
21 uo=0.45;//Superficial gas velocity in m/s
22 db=0.05;//Equilibrium bubble size in m from Fig
    .(6.8)
23 ub=1.5;//Velocity of bubbles in bubbling bed in m/s
    from Fig.(6.11(a))
24 g=9.81;//Acceleration due to gravity in square m/s2
25
26 //CALCULATION
27 ubr=0.711*(g*db)0.5;//Rise velocity of bubble from
    Eqn.(6.7)
28 KbcA=4.5*(umf/db)+5.85*((DA0.5*g0.25)/db(5/4));//
    Gas interchange coefficient between bubble and
    cloud from Eqn.(10.27)
29 KceA=6.77*((DA*ephsilonmf*0.711*(g*db)0.5)/db3)
    0.5;//Gas interchange coefficient between
    emulsion and cloud from Eqn.(10.34)
30 KbcR=4.5*(umf/db)+5.85*((DR0.5*g0.25)/db(5/4));//
    Gas interchange coefficient between bubble and
    cloud from Eqn.(10.27)
31 KceR=6.77*((DR*ephsilonmf*0.711*(g*db)0.5)/db3)
    0.5;//Gas interchange coefficient between
    emulsion and cloud from Eqn.(10.34)
32 delta=uo/ub;//Fraction of bed in bubbles from Eqn
    .(6.29)
33 fw=0.6;//Wake volume to bubble volume from Fig.(5.8)
34 gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw
    );//Volume of solids in cloud to that of the

```

```

    bubble from Eqn.(6.36)
35 gammae=((1-epsilonmf)*((1-delta)/delta))-gammab-
    gammac;//Volume of solids in emulsion to that of
    the bubble from Eqn.(6.35)
36 epsilonf=1-(1-delta)*(1-epsilonmf);//Void fraction
    of fixed bed from Eqn.(6.20)
37 Lf=(1-epsilonm)*Lm/(1-epsilonf);//Length of fixed
    bed from Eqn.(6.19)
38 Krtou=Kr1*Lm*(1-epsilonm)/uo;//Dimensionless
    reaction rate group from Eqn.(5)
39 Kr12=Kr1;//Since the reactions are a special case of
    Denbigh scheme
40 Kr34=Kr3;
41 Kf1=(gammab*Kr12+1/((1/KbcA)+(1/(gammac*Kr12+1/((1/
    KceA)+(1/(gammae*Kr12)))))))*(delta/(1-epsilonf)
    );//Rate of reaction 1 for fluidized bed from Eqn
    .(14)
42 Kf3=(gammab*Kr34+1/((1/KbcR)+(1/(gammac*Kr34+1/((1/
    KceR)+(1/(gammae*Kr34)))))))*(delta/(1-epsilonf)
    );//Rate of reaction 2 for fluidized bed from Eqn
    .(14)
43 Kf12=Kf1;
44 Kf34=Kf3;
45 KfA=[[KbcR*KceA/gammac^2+(Kr12+KceA/gammac+KceA/
    gammae)*(Kr34+KceR/gammac+KceR/gammae)]*delta*
    KbcA*Kr12*Kr34/(1-epsilonf)]/[[Kr12+KbcA/gammac
    ]*(Kr12+KceA/gammae)+Kr12*KceA/gammac]*[(Kr34+
    KbcR/gammac)*(Kr34+KceR/gammae)+Kr34*KceR/gammac
    ]];//Rate of reaction with respect to A from Eqn
    .(35)
46 KfAR=Kr1/Kr12*Kf12-KfA;//Rate of reaction from Eqn
    .(34)
47 tou=Lf*(1-epsilonf)/uo;//Residence time from Eqn
    .(5)
48 XA=1-exp(-Kf1*tou);//Conversion of A from Eqn.(26)
49 XR=1-((KfAR/(Kf12-Kf34))*[exp(-Kf34*tou)-exp(-Kf12*
    tou)]);//Conversion of R from Eqn.(27)
50 SR=(1-XR)/XA;//Selectivity of R

```



```

51
52 //OUTPUT
53
54 mprintf('\nRate of reaction 1 for fluidized bed:%f',
          Kf1);
55 mprintf('\nRate of reaction 2 for fluidized bed:%f',
          Kf3);
56 mprintf('\nRate of reaction 1 with respect to A:%f',
          KfA);
57 mprintf('\nThe Conversion of Napthalene:%f
          percentage ',XA*100);
58 mprintf('\nThe selectivity of Phthalic anhydride:%f
          percentage ',SR*100);
59
60 //=====END OF PROGRAM

```

Scilab code Exa 12.3 Bubbling Bed Reactor for Intermediate Sized Reactor

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-12, Example 3, Page 302
4 //Title: Bubbling Bed Reactor for Intermediate Sized
   Reactor
5 //

```

```

6
7 clear all
8 clc

```

```

9
10 //INPUT
11 Kr=3; //rate constant in m3 gas/m3 cat s
12 db=0.12; //Equilibrium bubble size in m
13 D=9E-5; //Diffusion coefficient of gas in m2/s
14 dpbar=68; //Average particle size in micrometers
15 epsilonm=0.42; //Void fraction of fixed bed
16 uo=0.4; //Superficial gas velocity in m/s
17 Lm=0.8; //Length of the bed in m
18 epsilonmmf=0.45; //Void fraction at minimum
    fluidization condition
19 umf=0.21; //Velocity at minimum fluidization
    condition in m/s
20 gammab=0; //Ratio of volume of dispersed solids to
    that of bubble phase
21 g=9.81; //Acceleration due to gravity in square m/s2
22
23 //CALCULATION
24 ubr=0.711*(g*db)0.5; //Rise velocity of bubble from
    Eqn.(6.7)
25 ub=uo-umf+ubr; //Velocity of bubbles in bubbling beds
    in Eqn.(6.8)
26 ubstar=ub+3*umf; //Rise velocity of the bubble gas
    from Eqn.(45)
27 delta=(uo-umf)/(ub+umf); //Fraction of bed in bubbles
    from Eqn.(6.46)
28 Kbe=4.5*(umf/db); //Interchange coefficient between
    bubble and emulsion from Eqn.(47)
29 Lf=Lm*(1-epsilonm)/((1-delta)*(1-epsilonmmf)); //
    Length of fixed bed
30 phi=[(Kr/Kbe)2*{(1-epsilonmmf)-gammab*(umf/ubstar)
    }2+((delta/(1-delta))+umf/ubstar)2+2*(Kr/Kbe)
    *{(1-epsilonmmf)-gammab*(umf/ubstar)}*((delta/(1-
    delta))-umf/ubstar)]0.5; //From Eqn.(52)
31 q1=0.5*Kr/umf*{(1-epsilonmmf)+gammab*(umf/ubstar)
    }+0.5*Kbe/umf*{((delta/(1-delta))+umf/ubstar)-phi
    }; //From Eqn.(50)
32 q2=0.5*Kr/umf*{(1-epsilonmmf)+gammab*(umf/ubstar)

```

```

    }+0.5*Kbe/umf*{((delta/(1-delta))+umf/ubstar)+phi
    }; //From Eqn.(50)
33 si1=0.5-0.5*((1-delta)/delta)*[umf/ubstar-Kr/Kbe
    *{(1-epsilon*umf)-gamma*(umf/ubstar)}-phi]; //From
    Eqn.(51)
34 si2=0.5-0.5*((1-delta)/delta)*[umf/ubstar-Kr/Kbe
    *{(1-epsilon*umf)-gamma*(umf/ubstar)}+phi]; //From
    Eqn.(51)
35 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)
36 Krtou=Kr*Lm*(1-epsilon)/uo; //Dimensionless
    reaction rate group from Eqn.(5)
37
38 //OUTPUT
39 mprintf('\nCOmparing the values of 1-XA = %f and
    Krtou = %f with Fig.(6), we can concluce that
    this operating condition is shown as point A in
    Fig.(3)',1-XA,Krtou);
40 printf('\nLine 2 gives the locus of conversions for
    different values of the reaction rate group for
    this fluidized contacting');
41
42 //=====END OF PROGRAM

```

Scilab code Exa 12.4 Reaction in the Slow Bubble Regime

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2

```

```

3 //Chapter -12, Example 4, Page 305
4 //Title: Reaction in the Slow Bubble Regime
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 uo=0.25; //Superficial gas velocity in m/s
12 db=0.025; //Equilibrium bubble size in m
13 Kr=1.5; //rate constant in m3 gas/m3 cat s
14 umf=0.21; //Velocity at minimum fluidization
    condition in m/s
15 Lm=0.8; //Length of the bed in m
16 ephsilonm=0.42; //Void fraction of fixed bed
17 g=9.81; //Acceleration due to gravity in square m/s2
18
19 //CALCULATION
20 ubr=0.711*(g*db)0.5; //Rise velocity of bubble from
    Eqn.(6.7)
21 ub=uo-umf+ubr; //Velocity of bubbles in bubbling beds
    in Eqn.(6.8)
22 delta=(uo-umf)/(ub+2*umf); //Fraction of bed in
    bubbles from Eqn.(55) since ub/umf<<1
23 XA=1-exp(-Kr*Lm*((1-ephsilonm)/uo)*(umf/uo)*(1-delta
    )); //Conversion from Eqn.(57)
24 Krtou=Kr*Lm*(1-ephsilonm)/uo; //Dimensionless
    reaction rate group from Eqn.(5)
25
26
27 //OUTPUT
28 mprintf('\nComparing the values of 1-XA = %f and
    Krtou = %f with Fig.(6), we can conclude that
    this operating condition is shown as point B in
    Fig.(3)',1-XA,Krtou);
29 printf('\nLine 3 gives the locus of conversions for

```

```

different values of the reaction rate group for
this fluidized contacting');
30
31 //=====END OF PROGRAM
=====

```

Scilab code Exa 12.5 Conversion in the Freeboard of a Reactor

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering(II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-12, Example 5, Page 307
4 //Title: Conversion in the Freeboard of a Reactor
5 //
=====

6
7 clear all
8 clc
9
10 //INPUT
11 uo=0.3;//Superficial gas velocity in m/s
12 Lf=1.1;//Length of fixed bed in m
13 Hf=1.2;//Length of freeboard in m
14 db=0.04;//Equilibrium bubble size in m
15 umf=0.006;//Velocity at minimum fluidization
  condition in m/s
16 ephsilonmf=0.55;//Void fraction at minimum
  fluidization condition
17 gammab=0.005;//Ratio of volume of dispersed solids
  to that of bubble phase
18 Kr=10;//rate constant in m^3 gas/m^3 cat s

```

```

19 D=2E-5; // Diffusion coefficient of gas in m2/s
20 g=9.81; // Acceleration due to gravity in square m/s2
21
22 //CALCULATION
23 ubr=0.711*(g*db)0.5; // Rise velocity of bubble from
    Eqn.(6.7)
24 ub=uo-umf+ubr; // Velocity of bubbles in bubbling beds
    in Eqn.(6.8)
25 Kbc=4.5*(umf/db)+5.85*((D0.5*g0.25)/db(5/4)); //
    Gas interchange coefficient between bubble and
    cloud from Eqn.(10.27)
26 Kce=6.77*((D*ephsilonmf*0.711*(g*db)0.5)/db3)0.5;
    // Gas interchange coefficient between emulsion
    and cloud from Eqn.(10.34)
27 delta=uo/ub; // Fraction of bed in bubbles from Eqn
    .(6.29)
28 ephsilonf=1-(1-delta)*(1-ephsilonmf); // Void fraction
    of fixed bed from Eqn.(6.20)
29 fw=0.6; // Wake volume to bubble volume from Fig.(5.8)
30 gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw
    ); // Volume of solids in cloud to that of the
    bubble from Eqn.(6.36)
31 gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-
    gammac; // Volume of solids in emulsion to that of
    the bubble from Eqn.(6.35)
32 Kf=(gammab*Kr)+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)
    +(1/(gammae*Kr)))))); // Reaction rate for fluidized
    bed from Eqn.(14)
33 XA=1-exp(-1*Kf*Lf/ub); // Conversion at the top of
    dense bed from Eqn.(16)
34 etabed=(Kf*delta)/(Kr*(1-ephsilonf)); // Reactor
    efficiency from Eqn.(22)
35 a=0.6/uo // Since uoa = 0.6 s-1 from Fig.(5)
36 adash=6.62; // From Fig.(5)
37 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)
38 XA2=1-(1-XA1)*(1-XA); // Conversion at the exit from

```

Eqn. (64)

```
39
40 //OUTPUT
41 printf('\nThe conversion:');
42 mprintf('\n\tAt the top pf the dense bed: %f
    percentage',XA*100);
43 mprintf('\n\tAt the reactor exit: %f percentage',XA2
    *100);
44
45 //Disclaimer: The value of kf deviate from the one
    given in textbook, where as it is close to the
    value obtained by manual calculation.
46 //=====END OF PROGRAM
```

Chapter 13

Heat Transfer between Fluidized Beds and Surfaces

Scilab code Exa 13.1 h on a Horizontal Tube Bank

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-13, Example 1, Page 331
4 //Title: h on a Horizontal Tube Bank
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 dp=57; // Particle size in micrometer
12 rhos=940; // Density of solids in kg/m^3
13 Cps=828; // Specific heat capacity of the solid in J/
   kg K
14 ks=0.20; // Thermal conductivity of solids in W/m k
```



```

15 kg=0.035; //Thermal concuctivity of gas in W/m k
16 umf=0.006; //Velocity at minimum fluidization
    condition in m/s
17 ephsilonmf=0.476; //Void fraction at minimum
    fluidization condition
18 do1=0.0254; //Outside diameter of tube in m
19 L=1;
20 uo=[0.05;0.1;0.2;0.35]; //Superficial gas velocity in
    m/s
21 nw=[2;3.1;3.4;3.5]; //Bubble frequency in s^-1
22 g=9.81; //Acceleration due to gravity in square m/s^2
23
24
25 //CALCULATION
26 dte=4*do1*L/2*L; //Hydraulic diameter from Eqn.(6.13)
27 db=(1+1.5)*0.5*dte; //Rise velocity of the bubble
28 ubr=0.711*(g*db)^0.5; //Rise velocity of bubble from
    Eqn.(6.7)
29 phib=0.19; //From Fig.(15) for ks/kg=5.7
30 ke=ephsilonmf*kg+(1-ephsilonmf)*ks*[1/((phib*(ks/kg)
    )+(2/3))]; //Effective thermal conductivity of bed
    from Eqn.(3)
31 n=length(uo);
32 i=1;
33 while i<=n
34     ub(i)=uo(i)-umf+ubr; //Velocity of bubbles in
        bubbling beds in Eqn.(6.8)
35     delta(i)=uo(i)/ub(i); //Fraction of bed in
        bubbles from Eqn.(6.29)
36     h(i)=1.13*[ke*rhos*(1-ephsilonmf)*Cps*nw(i)*(1-
        delta(i))]^0.5; //Heat transfer coefficinet
        from Eqn.(18)
37     i=i+1;
38 end
39
40 //OUTPUT
41 printf('\nSuperficial gas velocity (m/s)');
42 printf('\tHeat transfer coefficient (W/m^2 k)');

```

```

43 i=1;
44 while i<=n
45     mprintf( '\n%f',uo(i));
46     mprintf( '\t\t\t%f',h(i));
47     i=i+1;
48 end
49
50 //=====END OF PROGRAM

```

Scilab code Exa 13.2 Effect of Gas Properties on h

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-13, Example 2, Page 332
4 //Title: Effect of Gas Properties on h
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 dp=80;// Particle size in micrometer
12 rhos=2550;//Density of solids in kg/m^3
13 Cps=756;//Specific heat capacity of the solid in J/
   kg K
14 ks=1.21;//Thermal conductivity of solids in W/m k
15 kg=[0.005;0.02;0.2];//Thermal concuctivity of gas in
   W/m k

```

```

16 ephsilonmf=0.476; //Void fraction at minimum
    fluidization condition
17
18 //CALCULATION
19 delta=0.5*(0.1+0.3); //For a gently fluidized bed
20 nw=3; //Bubble frequency in s-1 from Fig.(5.12) at
    about 30cm above the distributor
21 n=length(kg);
22 i=1;
23 while i<=n
24     x(i)=ks/kg(i); //To find different values of ks/
        kg
25     i=i+1;
26 end
27 phib=[0.08;0.10;0.20]; //From Fig.(15) for different
    values of ks/kg
28 i=1;
29 while i<=n
30     ke(i)=ephsilonmf*kg(i)+(1-ephsilonmf)*ks*[1/((
        phib(i)*(ks/kg(i)))+(2/3))]; // Effective
        thermal conductivity of bed from Eqn.(3)
31     h1(i)=1.13*[ke(i)*rhos*(1-ephsilonmf)*Cps*nw*(1-
        delta)]^0.5; //Heat transfer coefficient from
        Eqn.(18)
32     i=i+1;
33 end
34
35 //OUTPUT
36 printf('\nThermal conductivity of Gas(W/m K)');
37 printf('\tMax. heat transfer coefficient(W/m2 k)');
38 i=1;
39 while i<=n
40     mprintf('\n%f',kg(i));
41     mprintf('\t\t\t\t\t%f',h1(i));
42     i=i+1;
43 end
44
45 //=====END OF PROGRAM

```

Scilab code Exa 13.3 Effect of Particle Size on h

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-13, Example 3, Page 332
4 //Title: Effect of Particle Size on h
5 //
6
7 clear all
8 clc
9
10 //INPUT
11 rhos=2700;//Density of solids in kg/m^3
12 Cps=755;//Specific heat capacity of the solid in J/
   kg K
13 ks=1.2;//Thermal conductivity of solids in W/m k
14 kg=0.028;//Thermal conductivity of gas in W/m k
15 ephsilonmf=0.476;//Void fraction at minimum
   fluidization condition
16 dp1=10E-3;//Particle size for which h=hmax in m
17 hmax=250;//Max. heat transfer coefficient in W/m^2 K
18 nw=5;//Bubble frequency in s^-1
19 delta=0.1;//Fraction of bed in bubbles
20 deltaw=0.1;//Fraction of bed in bubbles in wall
   region
21 dp=2E-3;//Diameter of particle in m
22
```

```

23 //CALCULATION
24 x=ks/kg;
25 phib=0.11;
26 phiw=0.17;
27 ke=epsilonmf*kg+(1-epsilonmf)*ks*[1/((phib*(ks/kg)
    )+(2/3))]; //Effective thermal conductivity of bed
    from Eqn.(3)
28 hpacket=1.13*[ke*rhos*(1-epsilonmf)*Cps*nw/(1-
    deltaw)]^0.5; //Heat transfer coefficient for the
    packet of emulsion from Eqn.(11)
29 epsilonw=epsilonmf; //Void fraction in the wall
    region
30 kew=epsilonw*kg+(1-epsilonw)*ks*[(phiw*(ks/kg)
    +(1/3))^-1]; //Effective thermal conductivity in
    the wall region with stagnant gas from Eqn.(4)
31 y=(2*kew/dp1)+(hmax*hpacket)/(((1-deltaw)*hpacket)-
    hmax); //Calculating the term alphaw*Cpg*rhog*uo
    from Eqn.(16) by rearranging it
32 h=(1-deltaw)/((2*kew/dp+y*(dp/dp1)^0.5)^-1+hpacket
    ^-1); //Heat transfer coefficient from Eqn.(11) by
    using the value of y
33
34 //OUTPUT
35 mprintf('\nThe heat transfer coefficient for paricle
    size of %fm = %fW/m^2 K',dp,h);
36
37 //=====END OF PROGRAM

```

Scilab code Exa 13.4 Freeboard Heat Exchange

```

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

```

MA, pp 491

```
2
3 //Chapter-13, Example 4, Page 334
4 //Title: Freeboard Heat Exchange
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 Hf=4; //Height of freeboard in m
12 uo=2.4; //Superficial gas velocity in m/s
13 ho=350; //Heat transfer coefficient at the bottom of
    freeboard region in W/m2 K
14 hg=20; //Heat transfer coefficient in equivalent gas
    stream, but free of solids in W/m2 K
15
16 //CALCULATION
17 zf=[0;0.5;1;1.5;2;2.5;3;3.5;Hf]; //Height above the
    top of the dense bubbling fluidized bed
18 hr=0; //Assuming heat transfer due to radiation is
    negligible
19 a=1.5/uo; //Since decay coefficient from Fig.(7.12),
    a*uo=1.5s-1
20 n=length(zf);
21 i=1;
22 while i<=n
23     h(i)=(hr+hg)+(ho-hr-hg)*exp(-a*zf(i)/2); //Heat
        transfer coefficient from Eqn.(24) for zf=Hf
24     i=i+1;
25 end
26 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)
27
28 //OUTPUT
```

```

29 printf('\nThe required relationship is h(W/m^2 K) vs
   . zf(m) as in Fig.(9a)');
30 printf('\nHeight above the dense bubbling fluidized
   bed(m)');
31 printf('\tHeat transfer coefficient(W/m^2 k)');
32 i=1;
33 while i<=n
34     mprintf('\n%f',zf(i));
35     mprintf('\t\t\t\t\t%f',h(i));
36     i=i+1;
37 end
38 mprintf('\n\nThe mean heat transfer coefficient for
   the 4-m high freeboard =%fW/m^2 K',hbar);
39
40 //=====END OF PROGRAM

```

Chapter 14

The RTD and Size Distribution of Solids in Fluidized Beds

Scilab code Exa 14.1 Flow with Elutriation

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-14, Example 1, Page 343
4 //Title: Flow with Elutriation
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 Fo=2.7; //Feed rate in kg/min
12 Fof=0.9; //Feed rate of fines in feed in kg/min
13 Foc=1.8; //Feed rate of coarse in feed in kg/min
14 W=17; //Bed weight in kg
15 kf=0.8; //Elutriation of fines in min-1
```



```

16 kc=0.0125; //Elutriation of coarse in min-1
17
18 //CALCULATION
19 F1guess=1; //Guess value of F1
20 function [fn]=solver_func(F1) //Function defined for
    solving the system
21     fn=F1-(Fof/(1+(W/F1)*kf))-(Foc/(1+(W/F1)*kc)); //
    Eqn.(17)
22 endfunction
23 [F1]=fsolve(F1guess,solver_func,1E-6); //Inbuilt
    function fsolve to solve for F1
24 F1f=Fof/(1+(W/F1)*kf); //Flow rate of fines in
    entrained streams from Eqn.(16)
25 F1c=Foc/(1+(W/F1)*kc); //Flow rate of coarse in
    entrained streams from Eqn.(16)
26 F2f=Fof-F1f; //Flow rate of fines in overflow streams
    from Eqn.(9)
27 F2c=Foc-F1c; //Flow rate of coarse in overflow
    streams from Eqn.(9)
28 tbarf=1/((F1/W)+kf); //Mean residence time of fines
    from Eqn.(12)
29 tbarc=1/((F1/W)+kc); //Mean residence time of coarse
    from Eqn.(12)
30
31 //OUTPUT
32 fprintf('\nFlow rate in entrained stream:\n\tFines:
    %fkg/min\n\tCoarse:%fkg/min',F1f,F1c);
33 fprintf('\nFlow rate in overflow stream:\n\tFines:
    %fkg/min\n\tCoarse:%fkg/min',F2f,F2c);
34 fprintf('\nMean residence time:\n\tFines:%fmins\n\t
    Coarse:%fmins',tbarf,tbarc);
35
36 //=====END OF PROGRAM

```

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

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth–Heinemann,
   MA, pp 491
2
3 //Chapter–14, Example 2, Page 344
4 //Title: Flow with Elutriation and Change in Density
   of Solids
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 dt=4;//Diameter of reactor in m
12 ephsilonm=0.4;//Void fraction of static bed
13 rhos=2500;//Density of solid in the bed in kg/m^3
14 Lm=1.2;//Height of static bed in m
15 Fo=3000;//Feed rate in kg/hr
16 beta1=1.2;//Increase in density of solids
17 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
18 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;2.6;1.9;1.3;0.8;0.3;0];
   //Size distribution of solids in mm^-1
19 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;0;0;0];
   //Elutriation constant in s^-1
20 pi=3.14;
```

```

21
22 //CALCULATION
23 W=(pi/4*dt^2)*Lm*(1-epsilon)*rhos; //Weight of
    solids in bed
24 n=length(dp);
25 i=1;
26 F1guess=1000; //Guess value for F1
27 F1c=2510:10:2700;
28 while i<=n
29     function [fn]=solver_func(F1) //Function defined
        for solving the system
30         if k(i)==0 then x(i)=0; break
31         else x(i)=(po(i)/(W*k(i)/F1))
                *log(1+(W*k(i)/F1));
32         end
33     fn=F1/(Lm*Fo)-x(i);
34     endfunction
35     [F1(i)]=fsolve(F1guess,solver_func,1E-6); //Using
        inbuilt function fsolve for solving Eqn.(20)
        for F1
36     c(i)=F1c(i)/(Lm*Fo);
37     if F1(i)==0 then a(i)=0;
38     else a(i)=(po(i)/(W*k(i)/F1(i)))*log(1+(W*k(
        i)/F1(i)));
39     end
40     i=i+1;
41 end
42 plot(F1,a,F1,c);
43 xtitle('F1 vs a,c','F1','a,c');
44 F1n=2500; //The point where both the curves meet
45 F2=beta1*Fo-F1n; //Flow rate of the second leaving
    stream
46 j=1;
47 m=length(dp);
48 while j<=m
49     p1(j)=(1/F1n)*((Fo*po(j))/(1+(W/F1n)*k(j))); //
        Size distribution of stream 1 in mm-1 from
        Eqn.(16)

```

```

50     p2(j)=k(j)*W*p1(j)/F2; // Size distribution of
        stream 2 in mm-1 from Eqn.(7)
51     if p1(j)==0 & p2(j)==0 then tbar(j)=0;
52     else if p1(j)==0 then tbar(j)=(W*p1(j))/(F2*p2
        (j));
53         else if p2(j)==0 then tbar(j)=(W*p1(j))/(
            F1n*p1(j));
54         else tbar(j)=(W*p1(j))/(F1n*p1(j)+F2*p2(
            j)); // Average time in hr from Eqn
            .(11)
55             end
56         end
57     end
58     j=j+1;
59 end
60
61 //OUTPUT
62 printf('\nFlow rate of stream 1:%fkg/hr ',F1n);
63 printf('\nFlow rate of stream 2:%fkg/hr ',F2);
64 j=1;
65 mprintf('\ntbar(hr)');
66 while j<=m
67     mprintf('\n%f',tbar(j));
68     j=j+1;
69 end
70
71 //=====END OF PROGRAM

```

```

72 //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
        .

```

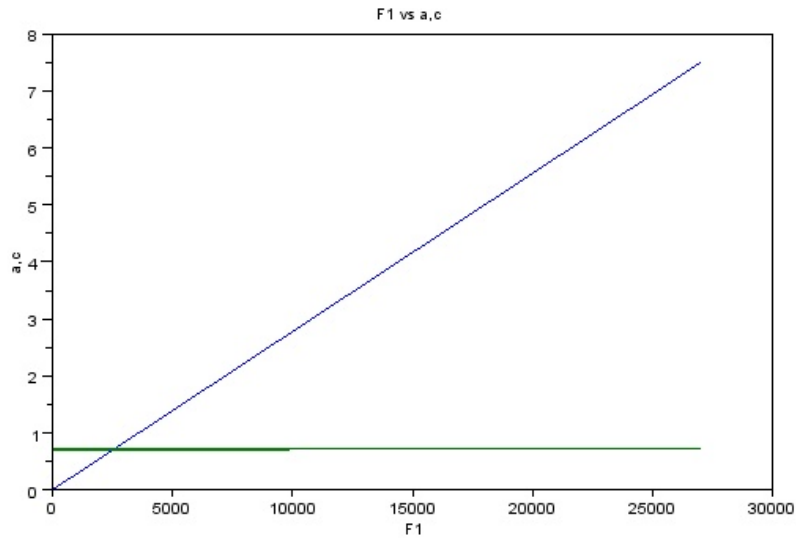


Figure 14.1: Flow with Elutriation and Change in Density of Solids

Scilab code Exa 14.3 Single Size Feed of Shrinking Particles

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-14, Example 3, Page 351
4 //Title: Single-Size Feed of Shrinking Particles
5 //


---


6
7 clear all

```

```

8  clc
9
10 //INPUT
11 dp=1; //Particle size in mm
12 Fo=10; //Feed rate in kg/min
13 k=0.1; //Particle shrinkage rate in mm/min
14
15 //CALCULATION
16 R=k/2; //Particle shrinkage rate in terms of radius
17 W=(Fo*dp/2)/(4*R); //Bed weight from Eqn.(42)
18
19 //OUTPUT
20 printf('\nWeight of bed:%fkg',W);
21
22 //=====END OF PROGRAM

```

Scilab code Exa 14.4 Wide Size Distribution of Shrinking Particle

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-14, Example 4, Page 352
4 //Title: Wide Size Distribution of Shrinking
   Particle
5 //

```

```

6
7 clear all
8 clc
9

```

```

10 //INPUT
11 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
12 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
13 k=[0;0;0;0;0;0;0;0;2.0;12.5;62.5]*10^-5; //
    Elutriation constant in s^-1
14 R=-1.58*10^-5; //Rate of particle shrinkage in mm/s
15 deldpi=0.1; //Size intervals in mm
16
17 //CALCULATION
18 n=length(dpi);
19 m=2; //Starting with the largest value size interval
    that contains solids
20 W(m-1)=0;
21 while m<=n
22     W(m)=(Fo(m)-R*W(m-1)/deldpi)/(k(m)-R/deldpi-3*R/
        dpi(m)); //From Eqn.(33)
23     m=m+1;
24 end
25 Wt=sum(W); //Total sum
26
27 //OUTPUT
28 printf('\nTotal mass in the bed:%fkg',Wt);
29
30 //=====END OF PROGRAM

```

Scilab code Exa 14.5 Elutriation and Attrition of Catalyst

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

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

```
2
3 //Chapter-14, Example 5, Page 353
4 //Title: Elutriation and Attrition of Catalyst
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 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
12 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
13 y=[0;0;0;0;0;0;0.625;10.225;159.25]*10^-6; //
    Elutriation and cyclone efficiency k(dpi)(1-eta(
    dpi))
14 F=0.01; //Rate at which solids are withdrawn in kg/s
15 W=40000; //Weight of bed in kg
16 dp1=0.11 //Initial size in mm
17 dp2=0.085; //Size after shrinking in mm
18 dpmin=0.01; //Minimum size in mm
19 deldpi=2*10^-2; //Size interval in mm
20 t=20.8; //Time in days
21 si=1;
22
23 //CALCULATION
24 kdash=log((dp1-dpmin)/(dp2-dpmin))/(t*24*3600); //
    Rate of particle shrinkage from Eqn.(24)
25 n=length(dpi);
26 m=2;
27 Fo=0.05; //Initial value of Fo
28 F1(m-1)=0;
29 s=0;
30 c=0;
```



```

31 t=1E-6;
32 while m<=n
33     R(m)=-kdash*(dpi(m)-dpmin); //Rate of size change
34     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)
35     F1(m)=x(m)*F;
36     c=c+x(m);
37     m=m+1;
38     if abs(c-1)<t then break
39     end
40     Fo=Fo+0.0001; //Incrementing Fo
41 end
42
43 //OUTPUT
44 mprintf('\nFeed rate with deldpi=%fmm is %fg/hr',
        deldpi,Fo);
45 i=1;
46 mprintf('\nBed composition');
47 while i<=n
48     printf('\n%f',x(i)*100);
49     i=i+1;
50 end
51
52 //=====END OF PROGRAM

```

Chapter 15

Circulation Systems

Scilab code Exa 15.1 Circulation Rate when Deactivation Controls

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-15, Example 1, Page 369
4 //Title: Circulation Rate when Deactivation Controls
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 thalf=1;//Half life of catalyst in s
12 F=960;//Feed rate of oil in tons/day
13 W=50;//Weight of the bed in tons
14 a=0.5;//Activity after time equal to half life
15 abar=0.01;//Average activity of the catalyst
16
17 //CALCULATION
```

```

18 Ka=-log(a)/thalf;//Rate constant is s-1, assuming I
    order kinetics from Eqn.(12)
19 Fs=Ka*W*abar/(1-abar);//Circulation rate of solids
    from Eqn.(16)
20 x=(Fs*60*60*24)/F;//Circulation rate per feed of oil
21
22 //OUTPUT
23 mprintf('\nSolid recirculation per feed of oil =
    %ftons of solid circulated/ton feed oil',x);
24
25 //=====END OF PROGRAM

```

Scilab code Exa 15.2 Circulation Rate when Heat Duty Controls

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-15, Example 2, Page 370
4 //Title: Circulation Rate when Heat Duty Controls
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 deltaHr1=1260;//Enthalpy change during endothermic
    reaction in kJ/kg
12 deltaHr2=-33900;//Enthalpy change during exothermic
    reaction in kJ/kg

```

```

13 H1=703; //Enthalpy of feed oil in kJ/kg
14 T1=260; //Temperature of feed oil in degree celcius
15 H3=1419; //Enthalpy of cracked product in kJ/kg
16 T3=500; //Temperature of cracked product in degree
    celcius
17 Ta=20; //Temperature of entering air in degree
    celcius
18 Cpa=1.09; //Specific heat of entering air in kJ/kg K
19 Cpf=1.05; //Specific heat of flue gases in kJ/kg K
20 Cps=1.01; //Specific heat of solids in kJ/kg K
21 Cpv=3.01; //Specific heat of vaporized feed in kJ/kg
    K
22 T4=[520;540;560;580;600;620;640;660]; //Temperature
    of flue gas in degree celcius
23 V=22.4; //Volume of 1 mole of Carbon dioxide gas in N
    -m^3
24 M=12; //Molecular weight of carbon in kg
25 rho=1.293; //Density of carbon dioxide gas in kg/N-m
    ^3
26 xa=0.21; //Mass fraction of oxygen in air
27 betac=0.07; //Mass fraction of carbon
28
29 //CALCULATION
30 n=length(T4);
31 i=1;
32
33 x2min=betac*(V*rho/(M*xa)); //Minimum amount of air
    required for complete combustion
34 while i<=n
35     x1(i)=(deltaHr1+0.93*H3-H1)/(Cps*(T4(i)-T3)); //
        Fs/F1 by simplifying the overall energy
        balance
36     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
37     if x2(i)>x2min then excess_air(i)=(x2(i)-x2min)/
        x2min; //Excess air used

```

```

38     else excess_air(i)=0;
39     end
40     i=i+1;
41 end
42
43 //OUTPUT
44 printf('\nT4(degree celcius)');
45 printf('\tFs/F1');
46 printf('\t\tF2/F1');
47 printf('\t\tExcess air(percentage)');
48 i=1;
49 while i<=n
50     mprintf('\n%f',T4(i));
51     mprintf('\t\t%f',x1(i));
52     mprintf('\t%f',x2(i));
53     mprintf('\t%f',excess_air(i)*100);
54     i=i+1;
55 end
56
57 //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.
58
59 //=====END OF PROGRAM

```

Scilab code Exa 15.3 Aeration of Fine Particle Downcomer

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2

```

```

3 //Chapter -15, Example 3, Page 379
4 //Title: Aeration of Fine Particle Downcomer
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 Fs=100;//Solid flowrate in kg/s
12 ephsilon1=0.55;
13 ephsilon2=0.5;
14 p1=120;//Pressure at upper level in kPa
15 rhos=1000;//Density of solid in kg/m^3
16 rhog=1;//Density of gas in kg/m^3
17 gc=1;//Conversion factor
18 g=9.81;//Acceleration due to gravity in m/s^2
19 di=0.34;//Diameter of downcomer in m
20 pi=3.14;
21
22 //CALCULATION
23 x=(ephsilon1/ephsilon2)*((1-ephsilon2)/(1-ephsilon1)
    );//To find pressure at lower level using Eqn
    .(30)
24 p2=x*p1;//Pressure at lower level using Eqn.(30)
25 deltap=p2-p1;
26 ephsilonbar=0.5*(ephsilon1+ephsilon2);
27 deltah=(deltap*10^3*gc)/(rhos*(1-ephsilonbar)*g);//
    Static head height from Eqn.(28)
28 At=0.25*pi*di^2;//Area of downcomer
29 Gs=Fs/At;//Flux of solids in downcomer
30 Gg=Gs*(ephsilon1/(1-ephsilon1))*(rhog/rhos)*(x-1);//
    Required gas aeration rate from Eqn.(31)
31 Fg=Gg*At;//Flow rate of gas required
32
33 //OUTPUT
34 mprintf('\nThe required flow rate of gas required

```

```

    for location of %fm below downcomer is %fkg/s',
    deltah,Fg);
35
36 //=====END OF PROGRAM

```

Scilab code Exa 15.4 Circulation in Side by Side Beds

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering(II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-15, Example 4, Page 380
4 //Title: Circulation in Side-by-Side Beds
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 Fs=600;//Solid circulation rate in kg/s
12 dpbar=60;//Mean size of solids in micrometer
13 pA=120;//Pressure in vessel A in kPa
14 pB=180;//Pressure in vessel B in kPa
15 LfA=8;//Bed height in vessel A in m
16 LfB=8;//Bed height in vessel B i m
17 //Bulk densities in kg/m^3
18 rho12=100;
19 rho34=400;
20 rho45=550;
21 rho67=200;

```

```

22 rho78=200;
23 rho910=400;
24 rho1011=400;
25 rho1112=550;
26 rho13=100;
27 deltapdA=7;//Pressure drop across the distributor in
    regenerator in kPa
28 deltapdB=8;//Pressure drop across the distributor in
    reactor in kPa
29 deltap12=(9+4);//Friction loss and pressure
    difference required to accelerate the solids in
    transfer lines in kPa
30 deltap78=(15+3);//Friction loss and pressure
    difference required to accelerate the solids in
    transfer lines in kPa
31 deltap45=20;//Friction loss across the reactor's
    stripper downcomer in kPa
32 deltap1112=4;//Friction loss across the regenerator's
    downcomer in kPa
33 deltapvA=5;//Pressure drop assigned for the control
    valve in regenerator in kPa
34 deltapvB=15;//Pressure drop assigned for the control
    valve in reactor in kPa
35 deltah12=15;//Height of the riser in m
36 deltah86=30;//Height of the riser in m
37 deltah1011=7;//Height difference h10-h11 in m
38 g=9.81;//Acceleration due to gravity in m/s^2
39 gc=1;//Conversion factor
40 pi=3.14;
41
42 //CALCULATION
43 Gs=900;//From Fig.(8), to find dt
44 dt=sqrt((4/pi)*Fs/Gs);//Diameter of the downcomer
45 //Height of downcomer A from Eqn.(7)
46 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];
47 //Height of downcomer B from Eqn.(8)

```



```

48 deltahB=(1/(rho45*g))*[-(pB-pA)*gc*10^3+(deltap45+
    deltapvB+deltap78+deltapdA)*gc*10^3+rho78*g*
    deltah86+rho910*g*LfA];
49
50 //OUTPUT
51 printf('\nHeight of downcomer for:');
52 mprintf('\n\tRegenerator:%fm',deltahA);
53 mprintf('\n\tReactor:%fm',deltahB);
54
55 //=====END OF PROGRAM

```

Scilab code Exa 15.5 Steam Seal of a Coarse Particle Downcomer

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-15, Example 5, Page 381
4 //Title: Steam Seal of a Coarse Particle Downcomer
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 dp=10^-3;//Particle diameter in m
12 dt=0.8;//Diameter of reactor in m
13 us=0.15;//Descend velocity of solids in m/s
14 L=15;//Length of downcomer
15 deltap1=300;//Pressure in lower vessel in kPa

```

```

16 deltap2=240; //Pressure in upper vessel in kPa
17 phis=0.8; //Sphericity of solids
18 ephsilonm=0.45; //Void fraction of bed
19 myu=4E-5; //Viscosity of gas in kg/m s
20 rhogl=2; //Density of gas in lower vessel in kg/m^3
21 rhogu=1.6; //Density of gas in upper vessel in kg/m^3
22 rhogbar=0.5*(rhogl+rhogu); //Average density in kg/m
    ^3
23 gc=1; //Conversion factor
24
25 //CALCULATION
26 //(a) Without steam seal
27 deltapfr=(deltap1-deltap2)*10^3; //Frictional
    pressure drop between two levels in Pa
28 deluguess=50; //Guess value of deltau
29 function [fn]=solver_func(delu) //Function defined for
    solving the system
30     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)
        );
31 endfunction
32 [delu]=fsolve(deluguess,solver_func,1E-6); //Using
    inbuilt function fsolve for solving Eqn.(25) for
    deltau
33 uo=(delu-us)*ephsilonm; //Superficial gas velocity
34 Fg=rhogbar*uo*(pi/4)*dt^2; //Flow rate of gs up the
    tube
35
36 //(c) With steam seal
37 //For section 1 to 3
38 L1=10;
39 deluguess1=50; //Guess value of deltau
40 function [fn]=solver_func1(delu1) //Function defined
    for solving the system
41     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)

```

```

    ));
42 endfunction
43 [delu1]=fsolve(deluguess1,solver_func1,1E-6); //Using
    inbuilt function fsolve for solving Eqn.(25) for
    deltau
44 uou=(delu1-us)*ephsilonm; //Upward superficial gas
    velocity
45 Fgu=rhogbar*uou*(pi/4)*dt^2; //Upward flow rate of gs
    up the tube
46 //For section 3 to 2
47 ugd=0.15; //Downward velocity of gas
48 uod=ugd*ephsilonm; //Downward superficial gas
    velocity
49 Fgd=rhogbar*uod*(pi/4)*dt^2; //Downward flow rate of
    gas up the tube
50 Fgt=Fgu+Fgd; //Total flow rate of gas
51
52 //OUTPUT
53 printf('\nWithout steam seal');
54 printf('\n\tFlow rate of gas up the tube:%fkg/s',Fg)
    ;
55 printf('\nWith steam seal');
56 printf('\n\tTotal flow rate of gas:%fkg/s',Fgt);
57
58 //=====END OF PROGRAM

```

Chapter 16

Design for Physical Operations

Scilab code Exa 16.1 Single Stage Limestone Calciner

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-16, Example 1, Page 404
4 //Title: Single-Stage Limestone Calciner
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 T=1000; //Operating temperature of calciner in degree
   celcius
12 deltaHr=1795; //Heat of reaction in kJ/kg
13 M1=0.1; //Molecular weight of Calcium carbonate in kg
   /mol
14 M2=0.056; //Molecular weight of CaO in kg/mol
15 M3=0.044; //Molecular weight of Carbon dioxide in kg
```

```

/mol
16 M4=0.029; //Molecular weight of Air in kg/mol
17 M5=0.029; //Molecular weight of Combustion gas in kg/
mol
18 Cp1=1.13; //Specific heat of Calcium carbonate in kJ/
kg K
19 Cp2=0.88; //Specific heat of CaO in kJ/kg K
20 Cp3=1.13; //Specific heat of Carbon dioxide in kJ/kg
K
21 Cp4=1.00; //Specific heat of Air in kJ/kg K
22 Cp5=1.13; //Specific heat of Calcium carbonate in kJ/
kg K
23 Tf=20; //Temperature of feed in degree celcius
24 ma=15; //Air required per kg of fuel in kg
25 Hc=41800; //Net combustion heat of fuel in kJ/kg
26 Tpi=20; //Initial temperature of solids in degree C
27 Tgi=1000; //Initial temperature of gas in degree C
28
29 //CALCULATION
30 mc=1; //Based on 1 kg of Calcium carbonate
31 B=(1/(Hc-(ma+mc)*Cp5*(T-Tpi)))*[M3*Cp3*(T-Tf)+M2*Cp2
*(T-Tf)+deltaHr] //Fuel consumption(kg fuel/kg
calcium carbonate)
32 B1=B*M3/M2; //Fuel consumption(kg fuel/kg Cao)
33 H=Hc*B1; //Heat required for calcination
34 eta=deltaHr/(B*Hc); //Thermal efficiency
35
36 //OUTPUT
37 mprintf('\nFuel consumption:%f kg fuel/kg Cao',B1);
38 mprintf('\nHeat requirement for calcination:%f kJ/kg
Cao',H);
39 mprintf('\nThermal efficiency:%f percentage',eta
*100);
40
41 //=====END OF PROGRAM

```

Scilab code Exa 16.2 Multistage Limestone Calciner

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-16, Example 2, Page 405
4 //Title: Multistage Limestone Calciner
5 //
6
7 clear all
8 clc
9
10 //INPUT
11 F=400; //Feed rate of Calcium carbonate in tons/day
12 T=1000; //Operating temperature of calciner in degree
   celcius
13 deltaHr=1795; //Heat of reaction in kJ/kg
14 M1=0.1; //Molecular weight of Calcium carbonate in kg
   /mol
15 M2=0.056; //Molecular weight of CaO in kg/mol
16 M3=0.044; //Molecular weight of Carbon dioxide in kg
   /mol
17 M4=0.029; //Molecular weight of Air in kg/mol
18 M5=0.029; //Molecular weight of Combustion gas in kg/
   mol
19 Cp1=1.13; //Specific heat of Calcium carbonate in kJ/
   kg K
20 Cp2=0.88; //Specific heat of CaO in kJ/kg K
21 Cp3=1.13; //Specific heat of Carbon dioxide in kJ/kg
   K
22 Cp4=1.00; //Specific heat of Air in kJ/kg K
```

```

23 Cp5=1.17; // Specific heat of Combustion gas in kJ/kg
    K
24 Tf=20; // Temperature of feed in degree celcius
25 ma=15; // Air required per kg of fuel in kg
26 uo=0.8; // Superficial gas velocity in m/s
27 Hc=41800; // Net combustion heat of fuel in kJ/kg
28 Tpi=20; // Initial temperature of solids in degree C
29 Tgi=1000; // Initial temperature of gas in degree C
30 rhoa=1.293; // Density of air in kg/m^3
31 pi=3.14;
32
33 //CALCULATION
34 mc=1; // Based on 1 kg of Calcium carbonate
35 Bguess=2; // Guess value of B
36 function [fn]=solver_func(B) // Function defined for
    solving the system
37     phi=((ma+mc)*Cp5*B+(M3*Cp3))/Cp1;
38     T3=(Tpi+(phi+phi^2+phi^3)*Tgi)/(1+phi+phi^2+phi
        ^3);
39     phiplus=30.6*B
40     Tr=(T+Tpi*phiplus)/(1+phiplus);
41     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;
42     //fn=(1/20800)*(2470-T3-13.34*(Tr-20));
43 endfunction
44 [B]=fsolve(Bguess,solver_func,1E-6); // Using inbuilt
    function fsolve for solving Eqn.(23) for tou
45 phi=((ma+mc)*Cp5*B+(M3*Cp3))/Cp1;
46 //Temperature of various stages
47 T1=(Tpi+(phi)*Tgi)/(1+phi);
48 T2=(Tpi+(phi+phi^2)*Tgi)/(1+phi+phi^2);
49 T3=(Tpi+(phi+phi^2+phi^3)*Tgi)/(1+phi+phi^2+phi^3);
50 phiplus=30.6*B
51 Tr=(T+Tpi*phiplus)/(1+phiplus);
52 eta=deltaHr/(B*Hc); // Thermal efficiency
53 H=B*Hc/M2; // Heat requirement
54 //For lower heat recovery section

```

```

55 Q1=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(Tr+273)));//
    Volumetric flow rate of gas in the lower heat
    recovery section
56 dt1=sqrt(4/pi*Q1/uo);//Diameter of lower bed
57 //For calcination section
58 Qc=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T+273)));//
    Volumetric flow rate of gas in the calcination
    section
59 dtc=sqrt(4/pi*Qc/uo);//Diameter of calcination
    section
60 //For I stage
61 Q1=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T1+273)));//
    Volumetric flow rate of gas in the I stage
62 dt1=sqrt(4/pi*Q1/uo);//Diameter of I stage
63 //For II stage
64 Q2=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T2+273)));//
    Volumetric flow rate of gas in the II stage
65 dt2=sqrt(4/pi*Q2/uo);//Diameter of II stage
66 //For III stage
67 Q3=(F*10^3/(24*3600))*B*ma/(rhoa*(273/(T3+273)));//
    Volumetric flow rate of gas in the III stage
68 dt3=sqrt(4/pi*Q3/uo);//Diameter of III stage
69
70 //OUTPUT
71 printf('\nDiameter of lower bed:%fm',dt1);
72 printf('\nDiameter of calcination section:%fm',dtc);
73 printf('\nBed no.\t\t1\t2\t3');
74 printf('\nDiameter (m)%f\t%f\t%f',dt1,dt2,dt3);
75
76 //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.
77
78 //=====END OF PROGRAM

```

Scilab code Exa 16.3 Multistage Adsorber

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-16, Example 3, Page 413
4 //Title: Multistage Adsorber
5 //
6
7 clear all
8 clc
9
10 //INPUT
11 T=20;//Temperature in degree C
12 M=0.018;//Molecular weight of water in kg/mol
13 Q=10;//Flow rate of dry air in m^3/s
14 R=82.06E-6;//Universal gas constant
15 pi=0.0001;//Initial moisture content in atm
16 pj=0.01;//Final moisture content in atm
17
18 //CALCULATION
19 a=Q*(273+T)/273;//Term At*uo
20 b=a*M/(R*(T+273));//Term C*At*uo
21 //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)
22 m=10.2;
23 Fo=b/m;//Flow rate of solids
24 Q3=(b/Fo)*(pj-pi);//Moisture content of leaving
```

```

        solids
25
26 //OUTPUT
27 printf('\nMoisture content of leaving solids:%f kg
        H2O/kg dry solids ',Q3);
28
29 //=====END OF PROGRAM

```

Scilab code Exa 16.4 Dryer Kinetics and Scale up

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering(II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-16, Example 4, Page 422
4 //Title: Dryer Kinetics and Scale-up
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 Qfi=0.20;//Initial moisture fraction
12 Qfbar=0.04;//Average final moisture fraction
13 rhos=2000;//Density of solid in kg/m^3
14 Cps=0.84;//Specific heat of solids in kJ/kg K
15 Fo=7.6E-4;//Flow rate of solids in kg/m^3
16 Tsi=20;//Initial temperature of solids in degree C
17 rhog=1;//Density of gas in kg/m^3
18 Cpg=1;//Specific heat of gas in kJ/kg K

```

```

19 uo=0.3; // Superficial gas velocity in m/s
20 Tgi=200; // Initial temperature of gas in degree C
21 L=2370; // Enthalpy of liquid in kJ/kg
22 Cpl=4.2; // Specific heat of liquid in kJ/kg K
23 dt=0.1; // Diameter of reactor in m
24 Lm=0.1; // Length of fixed bed in m
25 ephsilonm=0.45; // Void fraction of fixed bed
26 pi=3.14;
27 Fo1=1; // Feed rate for commercial-scale reactor in kg
    /s
28
29 //CALCULATION
30 //(a) Bed temperature
31 Teguess=50; // Guess value of Te
32 function [fn]=solver_func(Te) // Function defined for
    solving the system
33     fn=(pi/4)*dt^2*uo*rhog*Cpg*(Tgi-Te)-Fo*(Qfi-
        Qfbar)*[L+Cpl*(Te-Tsi)]-Fo*Cps*(Te-Tsi);
34 endfunction
35 [Te]=fsolve(Teguess,solver_func,1E-6); // Using
    inbuilt function fsolve for solving Eqn.(53) for
    Te
36
37 //(b) Drying time for a particle
38 xguess=2; // Guess value of x, ie term tou/tbar
39 function [fn]=solver_func1(x) // Function defined for
    solving the system
40     fn=1-(Qfbar/Qfi)-(1-exp(-x))/x;
41 endfunction
42 [x]=fsolve(xguess,solver_func1,1E-6); // Using inbuilt
    function fsolve for solving Eqn.(61) for x
43 W=(pi/4)*dt^2*Lm*(1-ephsilonm)*rhos; // Weight of
    solids in bed
44 tbar=W/Fo; // Mean residence time of solids from Eqn
    .(59)
45 tou=tbar*x; // Time for complete drying of a particle
46
47 //(c) Commercial-scale dryer

```

```

48 W1=Fo1*tbar;
49 Atguess=5;//Guess value of area
50 function [fn]=solver_func3(At)//Function defined for
    solving the system
51     fn=At*uo*rhog*Cpg*(Tgi-Te)-Fo1*(Qfi-Qfbar)*[L+
        Cpl*(Te-Tsi)]-Fo1*Cps*(Te-Tsi);
52 endfunction
53 [At]=fsolve(Atguess,solver_func3,1E-6);//Using
    inbuilt function fsolve for solving Eqn.(53) for
    At
54 dt1=sqrt(4/pi*At);//Diameter of commercial-scale
    dryer
55 Q1=At*uo*rhog;//Flow rate necessary for the
    operation
56
57 //OUTPUT
58 printf('\nBed temperature:%f degree C',Te);
59 printf('\nTime for complete drying of particle:%fs',
    tou);
60 printf('\nFlow rate of gas necessary for Commercial-
    scale dryer:%fkg/s',Q1);
61
62 //=====END OF PROGRAM

```

Scilab code Exa 16.5 Solvent Recovery from Polymer Particles

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering (II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-16, Example 5, Page 425
4 //Title: Solvent Recovery from Polymer Particles

```

```

5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 rhos=1600; //Density of solid in kg/m^3
12 Cps=1.25; //Specific heat of solids in kJ/kg K
13 Fo=0.5; //Flow rate of solids in kg/s
14 Tsi=20; //Initial temperature of solids in degree C
15 Qwi=1; //Initial moisture fraction in water
16 Qwf=0.2; //Final moisture fraction in water
17 Qhi=1.1; //Initial moisture fraction in heptane
18 Qhf=0.1; //Final moisture fraction in heptane
19 Tgi=240; //Initial temperature of gas in degree C
20 Te=110; //Bed temperature in degree C
21 ephsilonm=0.45; //Void fraction of fixed bed
22 ephsilonf=0.75; //Void fraction of fluidized bed
23 uo=0.6; //Superficial gas velocity in m/s
24 di=0.08; //Diameter of tubes in m
25 li=0.2; //Pitch for square arrangement
26 hw=400; //Heat transfer coefficient in W/m^2 K
27 Tc=238; //Temperature at which steam condenses in
    degree C
28 //Specific heats in kJ/kg K
29 Cwl=4.18; //Water liquid
30 Cwv=1.92; //Water vapor
31 Chl=2.05; //Heptane liquid
32 Chv=1.67; //Heptane vapor
33 //Latent heat of vaporization in kJ/kg
34 Lw=2260; //Water
35 Lh=326; //Heptane
36 //Density of vapor in kg/m^3 at operating conditions
37 rhow=0.56; //Water
38 rhoh=3.1; //Heptane
39 Lf=1.5; //Length of fixed bed in m

```

```

40 t=140; //Half-life of heptane in s
41 L=1.5; //Length of tubes in heat exchanger
42 pi=3.14;
43
44 //CALCULATION
45 //(a) Dryer without Internals
46 xw=(Qwi-Qwf)/(Qhi-Qhf); //Water-heptane weight ratio
47 xv=((Qwi-Qwf)/18)/((Qhi-Qhf)/100); //Water-heptane
    volume ratio
48 T=(Qwi-Qwf)/18+(Qhi-Qhf)/100; //Total volume
49 rhogbar=((Qwi-Qwf)/18)/T*rhow+((Qhi-Qhf)/100)/T*rhoh
    ; //Mean density of the vapor mixture
50 Cpgbar=((((Qwi-Qwf)/18)/T)*rhow*Cwv+(((Qhi-Qhf)/100)/
    T)*rhoh*Cwv); //Mean specific heat of vapor mixture
51 //Volumetric flow of recycle gas to the dryer in m
    ^3/s from Eqn.(53)
52 x=(Cpgbar*(Tgi-Te))^-1*[Fo*(Qwi-Qwf)*[Lw+Cwl*(Te-Tsi
    )]+Fo*(Qhi-Qhf)*[Lh+Chl*(Te-Tsi)]+Fo*(Cps*(Te-Tsi
    ))];
53 r=Fo*[(Qwi-Qwf)/rhow+(Qhi-Qhf)/rhoh]; //Rate of
    formation of vapor in bed
54 uo1=uo*(x/(x+r)); //Superficial velocity just above
    the distributor
55 At=x/uo1; //Cross-sectional area of bed
56 dt=sqrt(4/pi*At); //Diameter of bed
57 B=-log(Qwf/Qwi)/t; //Bed height from Eqn.(63)
58 tbar=((Qhi/Qhf)-1)/B; //Mean residence time of solids
59 W=Fo*tbar; //Weight of bed
60 Lm=W/(At*(1-epsilon_m)*rhos); //Static bed height
61 Lf=(Lm*(1-epsilon_m))/(1-epsilon_f); //Height of
    fluidized bed
62
63 //(b) Dryer with internal heaters
64 f=1/8; //Flow rate is 1/8th the flow rate of
    recirculation gas as in part (a)
65 x1=f*x; //Volumetric flow of recycle gas to the dryer
    in m^3/s from Eqn.(53)
66 uo2=uo*(x1/(x1+r)); //Superficial velocity just above

```

```

        the distributor
67 Abed=x1/uo2; //Cross-sectional area of bed
68 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)
69 Aw=q*10^3/(hw*(Tc-Te)); //Total surface area of heat
    exchanger tubes
70 Lt=Aw/(pi*di); //Total length of tubes
71 Nt=Lt/L; //Total number of tubes
72 Atubes=Nt*(pi/4*di^2); //Total cross-sectional area
    of tubes
73 Atotal=Abed+Atubes; //Total cross-sectional area of
    tube filled dryer
74 d=sqrt(Atotal*pi/4); //Diameter of vessel
75 li=sqrt(Atotal/Nt); //Pitch for square array of tubes
76
77 //OUTPUT
78 printf('\n\t\t\tBed diameter(m)\t\tRecycle vapor flow(
    m^3/s)');
79 printf('\nWithout internal heater\t%f\t%f',dt,x);
80 printf('\nWith heating tubes\t%f\t%f',d,x1);
81
82 //=====END OF PROGRAM

```

Chapter 17

Design of Catalytic Reactors

Scilab code Exa 17.1 Reactor Development Program

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-17, Example 1, Page 434
4 //Title: Reactor Development Program
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 dt=[0.081;0.205;3.6]; //Reactor diameter for the
   three reactors in m
12 dte=[0.04;0.12;0.70]; //Equivalent diameters for the
   three reactors in m
13 db=[0.05;0.057;0.07]; //Estimated bubble size in the
   three reactors in m
14 Kr1=1.3889; //Kinetic constant for Reaction 1 in s-1
```



```

15 Kr2=0.6111; //Kinetic constant for Reaction 2 in s-1
16 Kr3=0.022; //Kinetic constant for Reaction 3 in s-1
17 dp=60; //Particle size in micrometer
18 ephsilonm=0.50; //Void fraction of fixed bed
19 ephsilonmf=0.55; //Void fraction at minimum fluidized
    condition
20 umf=0.006; // //Velocity at minimum fluidization
    condition in m/s
21 D=2E-5; //Diffusion coefficient of gas in m2/s
22 gammab=0.005; //Ratio of volume of dispersed solids
    to that of bubble phase
23 uo=0.2; //Superficial gas velocity in m/s
24 XA=0.9; //Conversion
25 g=9.81; //Acceleration due to gravity in square m/s2
26
27 //CALCULATION
28 Kr12=Kr1+Kr2;
29 n=length(dt);
30 i=1;
31 while i<=n
32     //Preliminary Calculations
33     ubr(i)=0.711*(g*db(i))0.5; //Rise velocity of
        bubble from Eqn.(6.7)
34     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)
35     delta(i)=uo/ub(i); //Fraction of bed in bubbles
        from Eqn.(6.29)
36     ephsilonf(i)=1-(1-delta(i))*(1-ephsilonmf); //
        Void fraction of fixed bed from Eqn.(6.20)
37     fw=0.6; //Wake volume to bubble volume from Fig
        .(5.8)
38     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)
39     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)
40 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)
41 Kce(i)=6.77*((D*epsilonmf*0.711*(g*db(i))^0.5)/
    db(i)^3)^0.5; //Gas interchange coefficient
    between emulsion and cloud from Eqn.(10.34)
42 // Effective rate constant from Eqn.(12.32)
43 Kf12(i)=(gammab*Kr12+1/((1/Kbc(i))+1/(gammac(i)
    *Kr12+1/((1/Kce(i))+1/(gammae(i)*Kr12))))))
    *(delta(i)/(1-epsilonf(i)));
44 //Rate of reaction 2 for fluidized bed from Eqn
    .(12.14)
45 Kf3(i)=(gammab*Kr3+1/((1/Kbc(i))+1/(gammac(i)*
    Kr3+1/((1/Kce(i))+1/(gammae(i)*Kr3))))))*(
    delta(i)/(1-epsilonf(i)));
46 //Rate of reaction with respect to A from Eqn
    .(12.35)
47 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-epsilonf(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))];
48 KfAR(i)=((Kr1/Kr12)*Kf12(i))-KfA(i); //Rate of
    reaction from Eqn.(12.34)
49 KfAR1(i)=((Kr1/Kr12)*Kf12(i)); //Since KfA is
    small
50
51 //(b) Relate Selectivity with conversion in three
    reactors
52 x=-log(1-XA); //The term Kf12*tou in Eqn.(12.26)
53 tou(i)=x/Kf12(i); //Residence time from Eqn
    .(12.26)
54 y(i)=(KfAR1(i)/(Kf3(i)-Kf12(i)))*(exp(-x)-exp(-
    tou(i)*Kf3(i))); //CR/CAi from Eqn.(12.27)
55 SR(i)=y(i)/XA; //Selectivity of R

```

```

56
57 // (c) Relate exit composition to space time
58 tou1=5; // Space time in s
59 XA1(i)=1-exp(-Kf12(i)*tou1); // Conversion from
    Eqn.(12.26)
60 y1(i)=((KfAR1(i)/(Kf12(i)-Kf3(i)))*[exp(-Kf3(i)*
    tou1)-exp(-Kf12(i)*tou1)]); // CR/CAi R from
    Eqn.(12.27)
61
62 // (d) Calculate height of bed needed to maximize
    production
63 y2(i)=(KfAR1(i)/Kf12(i))*(Kf12(i)/Kf3(i))^(Kf3(i)
    )/(Kf3(i)-Kf12(i)); // CRmax/CAi R from Eqn
    .(12.37)
64 tou2(i)=log(Kf3(i)/Kf12(i))/(Kf3(i)-Kf12(i)); //
    Space time from Eqn.(38)
65 Lf(i)=(uo/(1-epsilonf(i)))*tou2(i); // Length of
    bed at fully fluidized condition from Eqn
    .(12.5)
66 Lm(i)=Lf(i)*(1-epsilonf(i))/(1-epsilonfm); //
    Length of bed when settled
67 XA2(i)=1-exp(-Kf12(i)*tou2(i)); // Conversion from
    Eqn.(12.26)
68 i=i+1;
69 end
70
71 //OUTPUT
72 printf('\nLet Laboratory, Pilot plant,
    Semicommercial unit be Reactor 1,2 & 3
    respectively ');
73 printf('\n(a) Relation between effective rate
    constant(Kf12) to the gas flow rate(uo)');
74 printf('\n\tReactor No.\tKf12(s^-1)\tuo(m/s)');
75 i=1;
76 while i<=n
77     mprintf('\n\t%1.0f',i);
78     mprintf('\t\t%f',Kf12(i));
79     mprintf('\t\t%f',uo);

```

```

80     i=i+1;
81 end
82 printf('\n(b) Relation between selectivity with
      conversion ');
83 printf('\n\tReactor No.\tKf12(s-1)\tSR(mol R formed
      /mol A reacted)');
84 i=1;
85 while i<=n
86     mprintf('\n\t%1.0f',i);
87     mprintf('\t\t%f',Kf12(i));
88     mprintf('\t%f',SR(i));
89     i=i+1;
90 end
91 printf('\n(c) Relation between exit composition and
      space time ');
92 printf('\n\tReactor No.\tXA\t\tCR/CAi');
93 i=1;
94 while i<=n
95     mprintf('\n\t%1.0f',i);
96     mprintf('\t\t%f',XA1(i));
97     mprintf('\t%f',y1(i));
98     i=i+1;
99 end
100 printf('\n(d) Height of bed needed to maximize the
      production of acrylonitrile ');
101 printf('\n\tReactor No.\tLm(m)\t\tXA');
102 i=1;
103 while i<=n
104     mprintf('\n\t%1.0f',i);
105     mprintf('\t\t%f',Lm(i));
106     mprintf('\t%f',XA2(i));
107     i=i+1;
108 end
109
110 //=====END OF PROGRAM

```

Scilab code Exa 17.2 Design of a Commercial Acrylonitrile Reactor

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-17, Example 2, Page 438
4 //Title: Design of a Commercial Acrylonitrile
   Reactor
5 //


---



---


6
7 clear all
8 clc
9
10 //INPUT
11 deltaHr=5.15E8;//Heat of reaction in J/k mol
12 W=5E4;//Weight of acrylonitrile produced per 334-day
   year in tonnes
13 db=0.07;//Estimated bubble size in m
14 dte=0.7;//Equivalent diameter in m
15 Kf12=0.35;//Effective rate constant in s-1 from
   Example 1
16 dp=60;//Particle size in micrometer
17 ephsilonm=0.50;//Void fraction of fixed bed
18 ephsilonmf=0.55;//Void fraction at minimum fluidized
   condition
19 T=460;//Temperature in reactor in degree C
20 Pr=2.5;//Pressure inside reactor in bar
21 //Feed gas composition
22 x1=1;//Propylene
23 x2=1.1;//Ammonia
24 x3=11;//Air
```

```

25 do1=0.08; //OD of heat exchanger tubes in m\
26 L=7; //Length of tubes in m
27 ho=300; //Outside heat transfer coefficient in W/m^2
    K
28 hi=1800; //Inside heat transfer coefficient in W/m^2
    K
29 Tc=253.4; //Temperature of coolant in degree C
30 pi=3.14;
31
32 //CALCULATION
33 //Preliminary calculation
34 uo=0.46; //Superficial gas velocity from Fig.E1(a)
    for the value of Kf12 & db
35 tou=8; //Space time from Fig.E2(b) for highest
    concentraion of product R
36 Lm=uo*tou/(1-epsilon);
37 y=0.58; //CR/CAi from Fig.E1(c) for the value of tou
    & Kf12
38 XA=0.95 //From Fig.E1(c) for the value of tou & Kf12
39 SR=y/XA; //Selectivity of R
40
41 //Cross-sectional area of the reactor
42 P=W*10^3/(334*24*3600); //Production rate of
    acrylonitrile
43 F=(P/0.053)/(SR*XA/0.042); //Feed rate of propylene
44 V=((F*22.4*(T+273)*(x1+x2+x3))/(42*273*Pr));
45 At=V/uo; //Cross-sectional area of reactor needed for
    the fluidized bed
46
47 //Heat exchanger calculation
48 q=F*XA*deltaHr/42; //Rate of heat liberation in the
    reactor
49 U=(ho^-1+hi^-1)^-1; //Overall heat transfer
    coefficient
50 deltaT=T-Tc; //Driving force for heat transfer
51 Aw=q/(U*deltaT); //Heat exchanger area required to
    remove q
52 Nt=Aw/(pi*do1*L);

```

```

53 li1=(At/Nt)^0.5;//Pitch for square pitch arrangement
54 dte1=4*[li1^2-(pi/4)*do1^2]/(pi*do1);
55 if dte1>dte then li=(pi/4*dte*do1+pi/4*do1^2)^0.5;//
    Pitch if we add dummy tubes
56 end
57 f=li^2-pi/4*do1^2;//Fraction of bed cross section
    taken up by tubes
58 dt1=sqrt(4/pi*At/(1-f));//Reactor diameter including
    all its tubes
59
60 //OUTPUT
61 printf('\nSuperficial gas velocity=%fm/s',uo);
62 printf('\nNo. of %1.0fm tubes required=%1.0f',L,Nt);
63 printf('\nReactor diameter=%fm',dt1);
64
65 //=====END OF PROGRAM

```

Scilab code Exa 17.3 Reactor Regenerator with Circulating Catalyst Catalytic Cracking

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
    Engineering (II Edition). Butterworth-Heinemann,
    MA, pp 491
2
3 //Chapter-17, Example 3, Page 444
4 //Title: Reactor-Regenerator with Circulating
    Catalyst: Catalytic Cracking
5 //

```

```

6
7 clear all

```

```

8  clc
9
10 //INPUT
11 db=0.08; //Estimated bubble size in m
12 dte=2; //Equivalent diameter in m
13 F1=55.6; //Feed rate of oil in kg/s
14 XA=0.63; //Conversion
15 uo=0.6; //Superficial gas velocity in m/s
16 T1=500; //Temperature of reactor in degree C
17 T2=580; //Temperature of regenerator in degree C
18 Fs=F1*23.3; //Solid circulation rate from Ex.(15.2)
19 rhos=1200; //Density of catalyst in kg/m^3
20 dpbar=60; //Average particle size in micrometer
21 ephsilonm=0.50; //Void fraction of fixed bed
22 ephsilonmf=0.55; //Void fraction at minimum fluidized
    condition
23 umf=0.006; //Velocity at minimum fluidization
    condition in m/s
24 dt=8; //Diameter of reactor in m
25 D=2E-5; //Diffusion coefficient of gas in m^2/s
26 Kr=8.6; //Rate constant for reaction at 500 degree C
    in s^-1
27 Ka1=0.06; //Rate constant for deactivation at 500
    degree C in s^-1
28 Ka2=0.012; //Rate constant for regeneration at 580
    degree C in s^-1
29 gammab=0.005; //Ratio of volume of dispersed solids
    to that of bubble phase
30 g=9.81; //Acceleration due to gravity in square m/s^2
31 pi=3.14;
32
33 //CALCULATION
34 //Parameters for the fluidized reactor
35 ubr=0.711*(g*db)^0.5; //Rise velocity of bubble from
    Eqn.(6.7)
36 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)

```



```

37 delta=uo/ub; //Fraction of bed in bubbles from Eqn
    .(6.29)
38 ephsilonf=1-(1-delta)*(1-ephsilonmf); //Void fraction
    of fixed bed from Eqn.(6.20)
39 fw=0.6; //Wake volume to bubble volume from Fig.(5.8)
40 gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw
    ); //Volume of solids in cloud to that of the
    bubble from Eqn.(6.36)
41 gammae=((1-ephsilonmf)*((1-delta)/delta))-gammab-
    gammac; //Volume of solids in emulsion to that of
    the bubble from Eqn.(6.35)
42 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)
43 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)
44
45 //Bed height versus catalyst activity in reactor
46 albar=0.07; //Guess value for average activity in
    reactor
47 x=Kr*a1bar; //Value of Kral to be used in the
    following equation
48 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)
49 tou=-log(1-XA)/Kf; //Space time from Eqn.(12.16)
50 Lm=tou*uo/(1-ephsilonm); //Length of fixed bed for
    guess value of albar
51 albar1=[0.0233;0.0465;0.0698;0.0930;0.116;0.140]; //
    Various activity values to find Lm
52 n=length(albar1);
53 i=1;
54 while i<=n
55     x1(i)=Kr*a1bar1(i);
56     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)
57     tou1(i)=-log(1-XA)/Kf1(i); //Space time from Eqn
        .(12.16)
58     Lm1(i)=tou1(i)*uo/(1-ephsilonm); //Length of
        fixed bed for guess value of albar...
        Condition (i)
59     i=i+1;
60 end
61
62 //Find the optimum size ratio for various albar
63 Lm=[5;6;7;8;10;12];
64 m=length(Lm);
65 i=1;
66 while i<=m
67     W1(i)=(pi/4)*dt^2*rhos*(1-ephsilonm)*Lm(i); //Bed
        weight
68     t1bar(i)=W1(i)/Fs; //Mean residence time of
        solids in reactor
69     t2bar(i)=t1bar(i)*(Ka1/Ka2)^0.5; //Mean residence
        time of soilds at optimum from Eqn.(16)
70     a1bar2(i)=(Ka2*t2bar(i))/(Ka1*t1bar(i)+Ka1*t1bar
        (i)*Ka2*t2bar(i)+Ka2*t2bar(i)); //From Eqn
        .(15) ... Condition (ii)
71     i=i+1;
72 end
73
74 //Final design values
75 Lm4=7.3; //For satisfying condition (i) & (ii)
76 a1bar3=0.0744; //By interpolation
77 x2=a1bar3*Kr;
78 W11=(pi/4)*dt^2*rhos*(1-ephsilonm)*Lm4; //Bed weight
        for reactor
79 t1bar1=W11/Fs; //Mean residence time of solids in
        reactor
80 a2bar=(1+Ka1*t1bar1)*a1bar3; //Average activity in
        regenerator from Eqn.(10)
81 t2bar1=t1bar1*(Ka1/Ka2)^0.5; //Mean residence time of
        solids in regenerator from Eqn.(16)

```

```

82 W2=W11*(t2bar1/t1bar1); //Bed weight for regenerator
83 dt2=dt*(W2/W11)^0.5; //Diameter of regenerator
    assuming same static bed height for reactor and
    regenerator
84
85 //OUTPUT
86 printf('\nBed height versus catalyst activity in
    reactor ');
87 printf('\n\tAverage activity ');
88 printf('\tLength of fixed bed(m) ');
89 i=1;
90 while i<=n
91     mprintf('\n\t%f',a1bar1(i));
92     mprintf('\t\t%f',Lm1(i));
93     i=i+1;
94 end
95 printf('\nOptimum size ratio for various activity in
    reactor ');
96 printf('\n\tLength of fixed bed(m) ');
97 printf('\tAverage activity ');
98 i=1;
99 while i<=m
100     mprintf('\n\t%f',Lm(i));
101     mprintf('\t\t%f',a1bar2(i));
102     i=i+1;
103 end
104 printf('\nFinal design values ');
105 printf('\n\tDiameter of reactor(m):%f',dt);
106 printf('\n\tBed weight for reactor(tons):%f',W11
    /10^3);
107 printf('\n\tBed weight for regenerator(tons):%f',W2
    /10^3);
108 printf('\n\tDiameter of regenerator(m):%f',dt2);
109 printf('\n\tSolid circulation rate(tons/hr):%f',Fs
    *3.6);
110
111 //=====END OF PROGRAM

```



Chapter 18

The Design of Noncatalytic Gas Solid Reactors

Scilab code Exa 18.1 Kinetics of Zinc Blende Roasting

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering (II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-18, Example 1, Page 456
4 //Title: Kinetics of Zinc Blende Roasting
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 xA=0.08; //Fraction of oxygen in stream
12 dp=[2;0.1]; //Particle diameter in mm
13 rhos=4130; //Density of catalyst in kg/m^3
14 Ds=8E-6; //Diffusion coefficient of solid in m^2/s
15 kc=0.02; //Reaction rate constant in m/s
```

```

16 P=10^5; //Pressure in bar\
17 R=8.314; //Universal gas constant
18 T=900; //Temperature in degree C
19 mB=0.09745; //Molecular weight of ZnS in kg/mol
20
21 //CALCULATION
22 b=2/3; //Stoichiometric coefficient of ZnS in the
    reaction equation
23 CA=xA*P/(R*(T+273)); //Concentration of Oxygen
24 rhob=rhos/mB; //Molar density of pure solid
25 n=length(dp);
26 i=1;
27 while i<=n
28     kbar(i)=(kc^-1+(dp(i)*10^-3/(12*Ds)))^-1; //
        Average reaction rate constant from Eqn.(11)
29     tou(i)=rhob*dp(i)*10^-3/(2*b*kbar(i)*CA); //Time
        for complete reaction in seconds from Eqn.(9)
30     i=i+1;
31 end
32
33 //OUTPUT
34 printf('\nParticle Size(mm)\tAverage rate constant(m
    /s)\tTime for complete reaction(min)');
35 i=1;
36 while i<=n
37     mprintf('\n%f\t\t%f\t\t\t%f', dp(i), kbar(i), tou(i)
        )/60);
38     i=i+1;
39 end
40
41 //=====END OF PROGRAM

```

Scilab code Exa 18.2 Kinetics of Carbon Burning

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-18, Example 2, Page 457
4 //Title: Kinetics of Carbon Burning
5 //


---


6
7 clear all
8 clc
9
10 //INPUT
11 xA=0.08;//Fraction of oxygen in stream
12 dp=1;//Particle diameter in mm
13 rhos=2200;//Density of catalyst in kg/m^3
14 kc=0.2;//Reaction rate constant in m/s
15 mC=0.012;//Molecular weight of carbon in kg/mol
16 P=10^5;//Pressure in bar\
17 R=8.314;//Universal gas constant
18 T=900;//Temperature in degree C
19
20 //CALCULATION
21 b=1;//Stoichiometric coefficient of C in the
   reaction equation
22 CA=xA*P/(R*(T+273));//Concentration of Oxygen
23 rhob=rhos/mC;//Molar density of pure solid reactant
24 tou=rhob*10^-3/(2*b*kc*CA);//Time required for
   complete reaction in seconds
25
26 //OUTPUT
27 mprintf('\nThe time required for complete combustion
   :%fmins',tou/60);
28
29 //=====END OF PROGRAM
```

Scilab code Exa 18.3 Roasting Kinetics from Flowing Solids Data

```
1 //Kunii D., Levenspiel O., 1991. Fluidization
   Engineering(II Edition). Butterworth-Heinemann,
   MA, pp 491
2
3 //Chapter-18, Example 3, Page 462
4 //Title: Roasting Kinetics from Flowing Solids Data
5 //
6
7 clear all
8 clc
9
10 //INPUT
11 dp=110;//Particle size in micrometer
12 T=900;//Temperature of roaster in degree C
13 tbar1=[3;10;30;50]);//Reported average time in min
14 XBbarr=[0.840;0.940;0.985;0.990]);//Reported value of
   average conversion
15 tbar=3;
16 XBbar=0.840;//Average conversion for tbar = 3 mins
17
18 //CALCULATION
19 //Uniform-Reaction Model
20 x=(1/tbar)*(1/(1- XBbar) -1);//Term KrCA of Eqn.(20)
21 n=length(tbar1);
22 i=1;
23 while i<=n
24     XBbar1(i)=1-1/(1+x*tbar1(i));//Average
```



```

        conversion using calculated value of KrCA
        from Eqn.(20)
25     i=i+1;
26 end
27
28 //Shrinking-Core, Rection Control
29 touguess=2;//Guess value of tou
30 function [fn]=solver_func(tou)//Function defined for
    solving the system
31     fn=(1-XBbar)-(0.25*tou/tbar)+(0.05*(tou/tbar)^2)
        -((1/120)*(tou/tbar)^3);
32 endfunction
33 [tou]=fsolve(touguess,solver_func,1E-6);//Using
    inbuilt function fsolve for solving Eqn.(23) for
    tou
34 i=1;
35 while i<=n
36     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)
37     i=i+1;
38 end
39
40 //Shrinking-Core, Diffusion Control
41 touguess1=2;//Guess value of tou
42 function [fn]=solver_func1(tou)//Function defined for
    solving the system
43     fn=(1-XBbar)-(1/5*tou/tbar)+(19/420*(tou/tbar)
        ^2)-(41/4620*(tou/tbar)^3)+(0.00149*(tou/tbar
        )^4);
44 endfunction
45 [tou1]=fsolve(touguess1,solver_func1,1E-6);//Using
    inbuilt function fsolve for solving Eqn.(23) for
    tou
46 i=1;
47 while i<=n
48     //Average conversion using calculated value of

```



```

6
7 clear all
8 clc
9
10 //INPUT
11 W=1; //Bed weight in kg
12 F1=0.01; //Solid feed rate in kg/min
13 dp=[200;600]; //Particle size in micrometer
14 XBbar=[0.85;0.64]; //Average conversion for
    corresponding particle sizes
15 rhos=2500; //Density of solid in kg/m^3
16 ephsilonm=0.4; //Void fraction of fixed bed
17 F11=4; //Feed rate of solids in tons/hr
18 XBbar1=0.98;
19 dp1=600;
20 pi=3.14;
21
22 //CALCULATION
23 //Shrinking-Core, Reaction Control
24 n=length(dp);
25 i=1;
26 touguess=2; //Guess value of tou
27 while i<=n
28     function [fn]=solver_func2(tou) //Function defined
        for solving the system
29         fn=(1-XBbar(i))-(0.25*tou/107)+(0.05*(tou
            /107)^2)-((1/120)*(tou/107)^3);
30     endfunction
31     [tou(i)]=fsolve(touguess,solver_func2,1E-6); //
        Using inbuilt function fsolve for solving Eqn
        .(23) for tou
32     i=i+1;
33 end
34 tou1=tou(2);
35
36 //For a single stage fluidized roaster
37 tbar1=0.25*(tou1/(1-XBbar1))/60; //Mean residence

```

```

        time of solids in reactor in hr from Eqn.(24)
38 W1=F11*tbar1;
39 dtguess=2;//Guess value of tou
40 function [fn]=solver_func3(dt)//Function defined for
    solving the system
41     fn=W1*10^3-(pi/4)*dt^2*0.5*dt*rhos*(1-epsilon)
        ;//Since Lm=0.5dt
42 endfunction
43 [dt]=fsolve(dtguess,solver_func3,1E-6);//Using
    inbuilt function fsolve for solving Eqn.(23) for
    tou
44 Lm=dt/2;//Length of bed required
45
46 //For a two-stage fluidized roaster
47 tbar2=tou1*sqrt(1/(20*(1-XBbar1)))/60;//Mean
    residence time of solids in reactor in hr from
    Eqn.(30)
48 W2=F11*tbar2;
49 dtguess1=2;//Guess value of tou
50 function [fn]=solver_func4(dt)//Function defined for
    solving the system
51     fn=W2*10^3-(pi/4)*dt^2*0.5*dt*rhos*(1-epsilon)
        ;//Since Lm=0.5dt
52 endfunction
53 [dt1]=fsolve(dtguess,solver_func4,1E-6);//Using
    inbuilt function fsolve for solving Eqn.(23) for
    tou
54 Lm1=dt1/2;//Length of bed required
55
56 //OUTPUT
57 printf('\nSingle stage fluidized roaster');
58 printf('\n\tWeight of bed needed:%ftons',W1);
59 printf('\n\tDiameter of reactor:%fm',dt);
60 printf('\n\tLength of bed:%fm',Lm);
61 printf('\nTwo-stage fluidized roaster');
62 printf('\n\tWeight of bed needed:%ftons',W2);
63 printf('\n\tDiameter of reactor:%fm',dt1);
64 printf('\n\tLength of bed:%fm',Lm1);

```

```

65 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);
66
67 //=====END OF PROGRAM

```

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

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering(II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-18, Example 5, Page 468
4 //Title: Design of a Roaster for Finely Ground Ore
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 T=900;//Temperature in roaster in degree C
12 P=101325;//Pressure in Pa
13 R=8.314;//Universal gas constant
14 dpbar=150;//Average particle size in micrometer
15 rhosbar=4130;//Average particle density in kg/m^3
16 kc=0.015//Rate constant in m/s for reaction which
    follows shrinking core model
17 Ds=8E-6;//Diffusion coefficient of solid in m^2/s
18 uo=0.6;//Superficial gas velocity in m/s
19 D=2.3E-4;//Diffusion coefficient of gas in m^2/s

```

```

20 Lm=1; //Length of fixed bed in m
21 dte=0.4; //Equivalent diameter of bed
22 umf=0.025; //Velocity at minimum fluidization
    condition in m/s
23 ephsilonm=0.45; //Void fraction of fixed bed
24 ephsilonmf=0.50; //Void fraction at minimum fluidized
    condition
25 db=0.2; //Estimated bubble size in m
26 gammab=0.005; //Ratio of volume of dispersed solids
    to that of bubble phase
27 Fo=2; //Feed rate of solids in kg/s
28 XA=0.6677; //Conversion of Oxygen
29 xA=0.21; //Mole fraction of oxygen in feed
30 mB=0.09744; //Molecular weight of ZnS
31 F=0.85; //Fraction of open area
32 g=9.81; //Acceleration due to gravity in square m/s^2
33 pi=3.14;
34
35 //CALCULATION
36 //(a)Extreme Calculation
37 a=3/2; //Stoichiometric coefficient of Oxygen in the
    reaction equation
38 At=(Fo/mB)*(a)/(uo*(273/(T+273))*(XA*xA)/0.0224);
39 dt=sqrt(At/F*4/pi);
40
41 //(b)The Three-Step Procedure
42 //Step 1. Conversion of gas
43 ubr=0.711*(g*db)^0.5; //Rise velocity of bubble from
    Eqn.(6.7)
44 ub=1.6*{(uo-umf)+1.13*db^0.5}*dte^1.35+ubr; //Bubble
    rise velocity for Geldart B particle
45 delta=uo/ub; //Fraction of bed in bubbles from Eqn
    .(6.29)
46 ephsilonf=1-(1-delta)*(1-ephsilonmf); //Void fraction
    of fixed bed from Eqn.(6.20)
47 fw=0.15; //Wake volume to bubble volume from Fig
    .(5.8)
48 gammac=(1-ephsilonmf)*((3/(ubr*ephsilonmf/umf-1))+fw

```

```

    );//Volume of solids in cloud to that of the
    bubble from Eqn.(6.36)
49 gammae=((1-epsilonmf)*((1-delta)/delta))-gammab-
    gammac;//Volume of solids in emulsion to that of
    the bubble from Eqn.(6.35)
50 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)
51 Kce=6.77*((D*epsilonmf*0.711*(g*db)^0.5)/db^3)^0.5;
    //Gas interchange coefficient between emulsion
    and cloud from Eqn.(10.34)
52 x=delta*Lm*(1-epsilonm)/((1-epsilonf)*uo);//Term
    Lf/ub of Eqn.(12.16) from Eqn.(6.19)
53 CAi=xA*P/(R*(T+273));//Initial concentration of
    oxygen
54
55 //Step 2. Conversion of solids
56 rhob=rhosbar/mB;//Density of ZnS
57 kbar=(kc^-1+(dpbar*10^-6/(12*Ds))^-1)^-1;//Modified
    rate constant from Eqn.(11)
58 tbar=At*Lm*(1-epsilonm)*rhosbar/Fo;//Mean residence
    time of solids
59 Krguess=2;//Guess value of Kr
60 function [fn]=solver_func(Kr)//Function defined for
    solving the system
61 Kf=gammab*Kr+1/((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)
    +(1/(gammae*Kr))))));//Reaction rate for
    fluidized bed from Eqn.(14)
62 XA=1-exp(-x*Kf);//Conversion of oxygen from Eqn
    .(42)
63 CAbar=(CAi*XA*uo)/(Kr*Lm*(1-epsilonm));//
    Average concentration of oxygen from Eqn.(43)
64 tou=rhob*dpbar*10^-6*a/(2*kbar*CAbar);//Time for
    complete reaction from Eqn.(9)
65 y=tbar/tou;//Term tbar/tou
66 XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average
    conversion of ZnS from Eqn.(22)
67 //Step 3. Material balance of both streams

```

```

68     fn=(Fo/mB)*XBbar-(At*uo*CAi*XA/a); //From Eqn.(44
        b)
69 endfunction
70 [Kr]=fsolve(Krguess,solver_func,1E-6); //Using
        inbuilt function fsolve for solving for Kr
71 Kf=gammab*Kr+1/(((1/Kbc)+(1/(gammac*Kr+1/((1/Kce)
        +(1/(gammae*Kr)))))); //Reaction rate for
        fluidized bed from Eqn.(14)
72 XA=1-exp(-x*Kf); //Conversion of oxygen from Eqn.(42)
73 CAbar=(CAi*XA*uo)/(Kr*Lm*(1-epsilomf)); //Average
        concentration of oxygen from Eqn.(43)
74 tou=rhob*dpbar*10^-6*a/(2*kbar*CAbar); //Time for
        complete reaction from Eqn.(9)
75 y=tbar/tou; //Term tbar/tou
76 XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y)); //Average
        conversion of ZnS from Eqn.(22)
77
78
79 //(c) For other feed rates of solids
80 F1=[2;2.5;3;3.5]; //Various feed rates of solids in
        kg/s
81 n=length(F1)
82 i=1;
83 Krguess1=2; //Guess value of Kr
84 while i<=n
85     tbar1(i)=At*Lm*(1-epsilom)*rhobar/F1(i); //
        Mean residence time of solids
86     function [fn]=solver_func1(Kr) //Function defined
        for solving the system
87         Kf1=gammab*Kr+1/(((1/Kbc)+(1/(gammac*Kr
            +1/((1/Kce)+(1/(gammae*Kr)))))); //
            Reaction rate for fluidized bed from Eqn
            .(14)
88         XA1=1-exp(-x*Kf1); //Conversion of oxygen
            from Eqn.(42)
89         CAbar1=(CAi*XA1*uo)/(Kr*Lm*(1-epsilom)); //
            Average concentration of oxygen from Eqn
            .(43)

```



```

90         tou1=rhob*dpbar*10^-6*a/(2*kbar*CAbar1);//
           Time for complete reaction from Eqn.(9)
91         y1(i)=tbar1(i)/tou1;//Term tbar/tou
92         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)
93         //Step 3. Material balance of both streams
94         fn=(F1(i)/mB)*XBbar1(i)-(At*uo*CAi*XA1/a);//
           From Eqn.(44b)
95     endfunction
96     [Kr1(i)]=fsolve(Krguess1,solver_func1,1E-6);//
           Using inbuilt function fsolve for solving Eqn
           .(23) for tou
97     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)
98     XA1(i)=1-exp(-x*Kf1(i));//Conversion of oxygen
           from Eqn.(42)
99     CAbar1(i)=(CAi*XA1(i)*uo)/(Kr1(i)*Lm*(1-
           ephsilonmf));//Average concentration of
           oxygen from Eqn.(43)
100    tou1(i)=rhob*dpbar*10^-6*a/(2*kbar*CAbar1(i));//
           Time for complete reaction from Eqn.(9)
101    y1(i)=tbar1(i)/tou1(i);//Term tbar/tou
102    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)
103    i=i+1;
104 end
105
106 //OUTPUT
107 printf('\nExtreme Calculation');
108 printf('\n\tDiameter of tube with all its internals:
           %fm',dt);
109 printf('\nThree step procedure');
110 printf('\n\tConversion of ZnS:%f',XBbar);
111 printf('\nFor other feed rates of solids');
112 printf('\n\tFeed(kg/s)\ttbar(s)\t\tXBbar/XA\tKrbar(s)

```

```

      ^-1)\tCAbar/CAi\ttou(s)\t\tXA\t\tXB');
113 i=1;
114 while i<=n
115     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));
116     i=i+1;
117 end
118
119 //=====END OF PROGRAM

```

Scilab code Exa 18.6 Design of a Roaster for Coarse Ore

```

1 //Kunii D., Levenspiel O., 1991. Fluidization
  Engineering(II Edition). Butterworth-Heinemann,
  MA, pp 491
2
3 //Chapter-18, Example 5, Page 471
4 //Title: Design of a Roaster for Coarse Ore
5 //

```

```

6
7 clear all
8 clc
9
10 //INPUT
11 T=900;//Temperature in roaster in degree C
12 P=101325;//Pressure in Pa
13 R=8.314;//Universal gas constant
14 dp=750;//Particle size in micrometer5
15 Fo=2.5;//Feed rate of solids in kg/s

```

```

16 uo=0.6; // Superficial gas velocity in m/s
17 W=80140; // Weight of bed in kg
18 ephsilonmf=0.50; // Void fraction at minimum fluidized
    condition
19 umf=0.5; // Velocity at minimum fluidization condition
    in m/s
20 db=0.2; // Estimated bubble size in m
21 g=9.81; // Acceleration due to gravity in square m/s^2
22 Lm=1; // Length of fixed bed in m
23 ephsilonm=0.45; // Void fraction of fixed bed
24 xA=0.21; // Mole fraction of oxygen in feed
25 kc=0.015 // Rate constant in m/s for reaction which
    follows shrinking core model
26 Ds=8E-6; // Diffusion coefficient of solid in m^2/s
27 rhosbar=4130; // Average particle density in kg/m^3
28 mB=0.09744; // Molecular weight of ZnS
29 a=3/2; // Stoichiometric coefficient of Oxygen in the
    reaction equation
30
31 // CALCULATION
32 // Selection of models to represent reactor
33 ubr=0.711*(g*db)^0.5; // Rise velocity of bubble from
    Eqn.(6.7)
34 f=ubr/(umf/ephsilonmf);
35
36 // Step 1.
37 ub=uo-umf+ubr; // Rise velocity of bubbles from Eqn
    .(6.8)
38 delta=(uo-umf)/(ub+2*umf); // Fraction of the bed in
    bubbles from Eqn.(6.26)
39 Krguess=2; // Guess value of Kr
40 x=Lm*(1-ephsilonm)*umf*(1-delta)/uo^2;
41 CAi=xA*P/(R*(T+273)); // Initial concentration of
    oxygen
42
43 // Step 2.
44 kbar=(kc^-1+(dp*10^-6/(12*Ds))^(-1))^(-1); // Modified
    rate constant from Eqn.(11)

```

```

45 tbar=W/Fo;//Mean residence time of solids from Eqn
    .(14.2)
46 rhob=rhosbar/mB;//Density of ZnS
47 function [fn]=solver_func1(Kr)//Function defined for
    solving the system
48     XA=1-exp(-x*Kr);//Conversion from Eqn.(42)
49     CAbar=(CAi*XA*uo^2)/(Kr*Lm*(1-ephpsilonm)*umf*(1-
        delta));//Average concentration of oxygen
        from Eqn.(43)
50     tou=rhob*dp*10^-6*a/(2*kbar*CAbar);//Time for
        complete reaction from Eqn.(9)
51     y=tbar/tou;//Term tbar/tou
52     XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average
        conversion of ZnS from Eqn.(22)
53     //Step 3.
54     fn=XBbar-1.2*XA;//From Table E5, for Fo=2.5kg/s
55 endfunction
56 [Kr]=fsolve(Krguess,solver_func1,1E-6);//Using
    inbuilt function fsolve for solving for Kr
57 XA=1-exp(-x*Kr);//Conversion from Eqn.(42)
58 CAbar=(CAi*XA*uo^2)/(Kr*Lm*(1-ephpsilonm)*umf*(1-
    delta))//Average concentration of oxygen from Eqn
    .(43)
59 tou=rhob*dp*10^-6*a/(2*kbar*CAbar);//Time for
    complete reaction from Eqn.(9)
60 y=tbar/tou;//Term tbar/tou
61 XBbar=3*y-6*y^2+6*y^3*(1-exp(-1/y));//Average
    conversion of ZnS from Eqn.(22)
62
63 //OUTPUT
64 printf('\nSelection of models to represent reactor')
    ;
65 printf('\n\tSince ratio ubr/(umf/ephpsilonmf)= %f <1,
    the reactor is operating in slow bubble regime',
    f);
66 printf('\n\tSince particle size =%f micrometer, they
    react according to shrinking-core model',dp);
67 printf('\n\tConversion obtained for %f micrometer

```

```
        particle : %f', dp, XBbar);  
68  
69 //=====END OF PROGRAM
```
