

Scilab Textbook Companion for  
Problems In Fluid Flow  
by D. J. Brasch And D. Whyman<sup>1</sup>

Created by  
Avik Kumar Das  
Fluid Mechanics & Hydraulics  
Civil Engineering  
IIT Bombay  
College Teacher  
Prof. Deepashree Rajee  
Cross-Checked by  
Ganesh R

July 31, 2019

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

# Book Description

**Title:** Problems In Fluid Flow

**Author:** D. J. Brasch And D. Whyman

**Publisher:** Edward Arnold

**Edition:** 1

**Year:** 1986

**ISBN:** 0-7131-3554-9

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

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

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

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

# Contents

List of Scilab Codes	4
1 Pipe Flow of Liquids	5
2 pipe flow of gasses and gas liquid mixtures	15
3 velocity boundary layers	23
4 Flow Measurement	28
5 Flow measurement in open channel	35
6 pumping of liquids	46
7 Flow Through Packed Beds	55
8 Filtration	59
9 Forces on bodies Immersed in fluids	67
10 Sedimentation and Clssification	74
11 Fluidisation	83
12 Pneumatic Conveying	90



# List of Scilab Codes

Exa 1.1.1	laminar turbulent pipe flow and Reynolds number . . . . .	5
Exa 1.1.2	conditions in pipeline while liquid passes in steady motion through it . . . . .	6
Exa 1.1.3	laminar flow and Hagen Poiseuille equation	7
Exa 1.1.4	velocity distribution in fluid in laminar motion in pipe . . . . .	8
Exa 1.1.5	comparison of laminar and turbulent flow .	8
Exa 1.1.6	power required for pumping local pressure in pipeline and the effects on both of an increase in pipe roughness . . . . .	9
Exa 1.1.7	power required for pumping when pipe system contains resistances to flow . . . . .	10
Exa 1.1.8	fluid flow rate and use of friction and chart .	11
Exa 1.1.9	time taken to drain a tank . . . . .	12
Exa 1.1.10	minimum pipe diameter to obtain a given fluid flow . . . . .	13
Exa 2.1.1	gas flow through pipe line when compressibility must be considered . . . . .	15
Exa 2.1.2	flow of ideal gas at maximum velocity under isothermal and adiabatic condition . . . . .	16
Exa 2.1.3	flow of a non ideal gas at maximum velocity under adiabatic condition . . . . .	18
Exa 2.1.4	venting of gas from pressure vessel . . . . .	19
Exa 2.1.5	gas flow measurement with veturimeter . . .	20
Exa 2.1.6	pressure drop required for flow of a gas liquid mixture through pipe . . . . .	21
Exa 3.1.1	streamline flow over a flat plate . . . . .	23

Exa 3.1.2	turbulent flow over a plate . . . . .	24
Exa 3.1.3	streamline and turbulent flow through and equations of universal velocity profile . . . . .	25
Exa 4.1.1	use of pitot tube to measure flow rate . . . . .	28
Exa 4.1.2	use of pitot tube to measure flow of gas . . . . .	29
Exa 4.1.3	use of orifice and manometer to measure flow . . . . .	30
Exa 4.1.4	determination of orifice size for flow measure- ment and pressure drop produced by orifice and venturi meters . . . . .	31
Exa 4.1.5	use of rotatometer for flow measurement . . . . .	32
Exa 4.1.6	mass of float required to measure fluid rate in rotatometer . . . . .	33
Exa 5.1.1	use of manning and chezy formulae . . . . .	35
Exa 5.1.2	stream depth in trapezoid channel . . . . .	36
Exa 5.1.3	optimum base angle of a Vshaped channel Slope of a channel . . . . .	36
Exa 5.1.4	stream depth and maximum velocity and flow rate in a pipe . . . . .	37
Exa 5.1.5	flow measurement with sharp crested weir . . . . .	38
Exa 5.1.6	equation of specific energy and analysis of tranquil and shooting flow . . . . .	39
Exa 5.1.7	alternate depth of stream gradient of mild and steep slope . . . . .	40
Exa 5.1.8	critical flw condition . . . . .	41
Exa 5.1.9	flow measurement with broad crested weir . . . . .	42
Exa 5.1.10	gradually varied flow behind a weir . . . . .	42
Exa 5.1.11	analysis of hydraulic jump . . . . .	43
Exa 6.1.1	cavitation and its avoidance in suction pipes . . . . .	46
Exa 6.1.2	specific speed of a centrifugal pump . . . . .	48
Exa 6.1.3	theoretical and effective characteristic of cen- trifugal pump flow rate . . . . .	48
Exa 6.1.4	flow rate when cetrifugal pumps operate singly and in parallel . . . . .	50
Exa 6.1.5	pumping with a reciprocating pump . . . . .	51
Exa 6.1.6	pumping with a air lift pump . . . . .	53
Exa 7.1.1	determination of particle size and specific sur- face area for a sample of powder . . . . .	55
Exa 7.1.2	rate of flow through packed bed . . . . .	56

Exa 7.1.3	determination of pressure drop to drive fluid through a packed bed of raschig rings then of similar size spheres and the determination of total area of surface presented with two types of packing . . . . .	57
Exa 8.1.1	constant rate of filtration in a plate and frame filter process . . . . .	59
Exa 8.1.2	Constant rate and pressure drop filtration .	60
Exa 8.1.3	determination of characteristic of filtration system . . . . .	61
Exa 8.1.4	constant pressure drop filtration of suspension which gives rise to a compressible filter cake . . . . .	62
Exa 8.1.5	filtration on a rotatory drum filter . . . . .	63
Exa 8.1.6	filtration of centrifugal filter . . . . .	65
Exa 9.1.1	drag forces and coefficient . . . . .	67
Exa 9.1.2	lift force and lift coefficient . . . . .	68
Exa 9.1.3	Particle diameter and terminal settling velocity . . . . .	69
Exa 9.1.4	terminal settling velocity of sphere . . . . .	69
Exa 9.1.5	effect of shape on drag force . . . . .	70
Exa 9.1.6	estimation of hindered settling velocity . . .	71
Exa 9.1.7	acceleration of settling particle in gravitational feild . . . . .	72
Exa 10.1.1	determination of settling velocity from a single batch sedimentation . . . . .	74
Exa 10.1.2	Minimum area required for a continuous thickener . . . . .	75
Exa 10.1.3	classification of materials on basis of settling velocities . . . . .	76
Exa 10.1.4	density variation of settling suspension . . .	78
Exa 10.1.5	determination of particle size distribution using a sedimentation method . . . . .	79
Exa 10.1.6	determination of particle size distribution of a suspended solid . . . . .	80
Exa 10.1.7	decanting of homogeneous suspension to obtain particle size of a given size range . . .	81
Exa 11.1.1	particulate and aggregative fluidisation . . .	83

Exa 11.1.2	calculation of minimum flow rates . . . . .	84
Exa 11.1.3	calculation of flow rates in fluidised beds . .	85
Exa 11.1.4	estimation of vessel diameters and height for fluidisation operations . . . . .	86
Exa 11.1.5	power required for pumping in fluidised beds	87
Exa 11.1.6	wall effect in fluidised beds . . . . .	88
Exa 11.1.7	effect of particle size on the ratio of terminal velocity . . . . .	89
Exa 12.1.1	flow pattern in pneumatic conveying . . . .	90
Exa 12.1.2	prediction of choking velocity and choking chok- ing voidage in a vertical transport line . . .	91
Exa 12.1.3	prediction of pressure drop in horizontal pneu- matic transport . . . . .	92
Exa 12.1.4	prediction of pressure drop in vertical pneu- matic transport . . . . .	93
Exa 12.1.5	density phase flow regime for pneumatic trans- port . . . . .	94
Exa 13.1.1	Equations of centrifugal operations . . . . .	96
Exa 13.1.2	fluid pressure in tubular bowl centrifuge . .	97
Exa 13.1.3	particle size determination of fine particles .	98
Exa 13.1.4	flow rates in continuous centrifugal sedimen- tation . . . . .	98
Exa 13.1.5	separation of two immiscible liquid by cen- trifugation . . . . .	99
Exa 13.1.6	Cyclone Separators . . . . .	100
Exa 13.1.7	efficiency of cyclone separators . . . . .	100

# Chapter 1

## Pipe Flow of Liquids

Scilab code Exa 1.1.1 laminar turbulent pipe flow and Reynolds number

```
1
2
3 //exapple 1.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 //part1
7 mu=6.3/100;//viscosity
8 rho=1170;//density
9 d=.3;//diameter of pipe
10 b=0.142;//conversion factor
11 pi=3.14;
12 //calculation
13 Q=150000*b/24/3600//flow rate
14 u=Q/pi/d^2*4//flow speed
15 Re=rho*u*d/mu
16 if Re>4000 then
17     disp(Re,"the system is in turbulent motion as
18         reynolds no is greater than 4000:");
19 elseif Re<2100 then
20     disp(Re,"the system is in laminar motion" );
21 else
```

```

21     disp(Re, "the system is in transition motion");
22 end
23 //part 2
24 mu=5.29/1000;
25 d=0.06;
26 G=0.32;//mass flow rate
27 Re= 4*G/pi/d/mu;
28 if    Re>4000    then
29     disp(Re,"the system is in turbulent motion as
           reynolds no is greater than 4000:");
30 elseif Re<2100 then
31     disp(Re,"the system is in laminar motion as Re
           is less than 2100" );
32 else
33     disp(Re, "the system is in transition motion");
34 end

```

---

Scilab code Exa 1.1.2 conditions in pipeline while liquid passes in steady motion

```

1
2
3 //exapple 1.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 G=21.2;//mass flow rate
7 rho=1120;//density
8 d=0.075;//diameter
9 l=50;
10 g=9.81;
11 pi=3.14;
12 delz=24/100;//head difference
13 //calculation
14 delP=delz*rho*g;//differece of pressure
15 u=4*G/pi/d^2/rho;
16 phi=delP/rho*d/l/u^2/4*50;

```

```

17 disp(phi,"The Stanton–Pannel friction factor per
    unit of length:");
18 R=phi*rho*u^2;
19 disp(R , "shear stress exerted by liquid on the pipe
    wall in (N/m^2):");
20 F=pi*d*l*R;
21 disp(F , "Total shear force exerted on the pipe in (
    N):");
22 Re=(.0396/phi)^4;//reynold's no.
23 mu=rho*u*d/Re;
24 disp(mu , "viscosity of liquid in (kg/m/s):")

```

---

### Scilab code Exa 1.1.3 laminar flow and Hagen Poiseuille equation

```

1
2
3 //exapple 1.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 pi=3.14;
7 g=9.81;
8 d=0.00125;
9 Re=2100;
10 l=0.035;
11 rhoc=779;//density of cyclohexane
12 rhow=999;//density of water
13 muc=1.02/1000;//viscosity of cyclo hexane
14 //calculation
15 u=Re*muc/rhoc/d;//speed
16 Q=pi*d^2*u/4;//volumetric flow rate
17 delP=32*muc*u*l/d^2;//pressure difference
18 delz=delP/(rho*g);
19 disp(delz*100 , "the difference between the rise
    levels of manometer in (cm):")

```

---

Scilab code Exa 1.1.4 velocity distribution in fluid in laminar motion in pipe

```
1
2
3 //exapple 1.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 d=0.05;
7 l=12;
8 per=100-2;
9 pi=3.1428
10 //calculation
11 s=sqrt(per/100/4*d^2); //radius of core of pure
    material
12 V=pi*d^2/4*l/(2*(1-(2*s)^2/d^2));
13 disp(V, "The volume of pure material so that 2%
    technical material appears at the end in (m^3):")
```

---

Scilab code Exa 1.1.5 comparison of laminar and turbulent flow

```
1
2
3 //exapple 1.5
4 clc; funcprot(0);
5 // Initialization of Variable
6 //part 1
7 a=1/2*(1-1/sqrt(2));
8 disp(a*100, "The percent value of d for which where
    pitot tube is kept show average velocity in
    streamline flow in (%):");
9 //part 2
10 a=(49/60)^7/2;
```

```

11 disp(a*100, "The percent value of d for which where
    pitot tube is kept show average velocity in
    turbulent flow in (%):");
12 //part 3
13 //on equating coefficient of r
14 y=a*2; //y=a/100*2*r
15 s=1-y; //s=r-y
16 //on equating coeff. of 1/4/mu*del(P)/del(l)
17 E=(1-s^2-.5)/.5;
18 disp(E, "The error shown by pitot tube at new
    position if value of streamlined flow flow was to
    be obtained in (%) :");
19 disp("The - sign indicates that it will display
    reduced velocity than what actually is");

```

---

**Scilab code Exa 1.1.6** power required for pumping local pressure in pipeline and the

```

1 //exapple 1.6
2 clc; funcprot(0);
3 // Initialization of Variable
4 rhon=1068; //density of nitric acid
5 mun=1.06/1000 //viscosity of nitric acid
6 g=9.81;
7 l=278;
8 pi = %pi;
9 d=0.032;
10 alpha=1;
11 h2=57.4; //height to be raised
12 h1=5; //height from which to be raised
13 e=.0035/1000; //roughness
14 G=2.35 //mass flow rate
15 //calculations
16 //part 1
17 u=4*G/rhon/pi/d^2;
18 Re=rhon*d*u/mun;

```

```

19 rr=e/d;//relative roughness
20 //Reading's from Moody's Chart
21 phi=.00225;//friction coeff.
22 W=u^2/2+g*(h2-h1)+4*phi*l*u^2/d;//The work done/kg
    of fluid flow in J/kg
23 V=abs(W)*G;
24 disp(abs(V)/1000, "The Power required to pump acid
    in kW :");
25 //part 2
26 P2=-u^2*rhon/2+g*(h1)*rhon+abs(W+2)*rhon;;
27 disp(P2/1000,"The gauge pressure at pump outlet when
    piping is new in (kPa)" );
28 //part 3
29 e=.05/1000;
30 Re=rhon*d*u/mun;
31 rr=e/d;
32 //Reading's from Moody's Chart
33 phi=0.0029;
34 W=u^2/2+g*(h2-h1)+4*phi*l*u^2/d;
35 Vnew=abs(W)*G;
36 Pi=(Vnew-V)/V*100;
37 disp(Pi , "The increase in power required to
    transfer in old pipe in (%):");
38 //part 4
39 P2=-u^2*rhon/2+g*(h1)*rhon+abs(W+2)*rhon;
40 disp(P2/1000,"The gauge pressure at pump outlet when
    piping is old in (kPa)" );

```

---

Scilab code Exa 1.1.7 power required for pumping when pipe system contains resistances

```

1
2
3 //exapple 1.7
4 clc; funcprot(0);
5 // Initialization of Variable

```

```

6 rho=990;
7 mu=5.88/10000;
8 g=9.81;
9 pi=3.14;
10 temp=46+273
11 e=1.8/10000//absolute roughness
12 Q=4800/1000/3600;
13 l=155;
14 h=10.5;
15 d=0.038;
16 delh=1.54//head loss at heat exchanger
17 effi=0.6//efficiency
18 //calculations
19 //part 1
20 u=Q*4/pi/d^2;
21 Re=rho*d*u/mu;
22 rr=e/d;//relative roughness
23 //from moody's diagram
24 phi=0.0038//friction factor
25 alpha=1//constant
26 leff=l+h+200*d+90*d;
27 Phe=g*delh//pressure head lost at heat exchanger
28 W=u^2/2/alpha+Phe+g*h+4*phi*leff*u^2/d;//work done
    by pump
29 G=Q*rho;//mass flow rate
30 P=W*G;//power required by pump
31 Pd=P/effi//power required to drive pump
32 disp(Pd/1000,"power required to drive pump in (kW)")
    ;
33 //part 2
34 P2=(-u^2/2/alpha+W)*rho;
35 disp(P2/1000,"The gauge pressure in (kPa):")

```

---

Scilab code Exa 1.1.8 fluid flow rate and use of friction and chart

```

1
2
3 //exapple 1.8
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=908;
7 mu=3.9/100;
8 g=9.81;
9 pi=3.14;
10 d=0.105;
11 l=87;
12 h=16.8;
13 e=0.046/1000;//absolute roughness
14 //calculations
15 //part1
16 P=-rho*g*h;//change in pressure
17 a=-P*rho*d^3/4/l/mu^2//a=phi*Re^2
18 //using graph given in book(appendix)
19 Re=8000;
20 u=mu*Re/rho/d;
21 Q=u*pi*d^2/4;
22 disp(Q,"Volumetric flow rate initial (m^3/s):");
23 //part 2
24 W=320;
25 Pd=W*rho;//pressure drop by pump
26 P=P-Pd;
27 a=-P*rho*d^3/4/l/mu^2//a=phi*Re^2
28 //using graph given in book(appendix)
29 Re=15000;
30 u=mu*Re/rho/d;
31 Q=u*pi*d^2/4;
32 disp(Q,"Volumetric flow rate final(part 2) (m^3/s):"
    );

```

---

Scilab code Exa 1.1.9 time taken to drain a tank

```

1 //exapple 1.9
2 clc; funcprot(0);
3 // Initialization of Variable
4 rho=1000;
5 mu=1.25/1000;
6 g=9.81;
7 pi=3.14
8 d1=0.28;//diameter of tank
9 d2=0.0042;//diameter of pipe
10 l=0.52;//length of pipe
11 rr=1.2/1000/d2;;//relative roughness
12 phid=0.00475;
13 disp(phid,"It is derived from tyhe graph giben in
    appedix and can be seen is arying b/w 0.0047 &
    ; 0.0048 dependent on D which varies from 0.25 to
    0.45")
14 //calculations
15 function [a]=intregrate()
16     s=0;
17     for i=1:1000
18         D=linspace(0.25,0.45,1000);
19         y=sqrt(((pi*d1^2/pi/d2^2)^2-1)/2/9.81+(4*
                phid*l*(pi*d1^2/pi/d2^2)^2)/d2/9.81)
                *((0.52+D(i))^-0.5)*2/10000;
20         s=s+y;
21
22     end
23     a=s;
24 endfunction
25 b=intregrate();
26 disp(b,"Time required to water level to fall in the
    tank in (s):");

```

---

Scilab code Exa 1.1.10 minimum pipe diameter to obtain a given fluid flow

```

1
2
3 //exapple 1.10
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1000;
7 mu=1.42/1000;
8 g=9.81;
9 pi=3.14;
10 l=485;
11 h=4.5
12 e=8.2/100000;
13 Q=1500*4.545/1000/3600;
14 disp("assume d as 6cm");
15 d=0.06;
16 u=4*Q/pi/d^2;
17 Re=rho*d*u/mu;
18 rr=e/d;//relative roughness
19 //using moody's chart
20 phi=0.0033//friction coeff.
21 d=(64*phi*l*Q^2/pi^2/g/h)^0.2;
22 disp(d*100, "The calculated d after (1st iteration
    which is close to what we assume so we do not do
    any more iteration) in(cm) ")

```

---

## Chapter 2

# pipe flow of gasses and gas liquid mixtures

Scilab code Exa 2.1.1 gas flow through pipe line when compressibility must be cons.

```
1
2
3 //exapple 2.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 pi=3.1428;
7 mmm=16.04/1000;//molar mass of methane
8 mV=22.414/1000;//molar volume
9 R=8.314;
10 mu=1.08/10^5;
11 r=4.2/100;//radius
12 rr=0.026/2/r;//relative roughness
13 Pfinal=560*1000;
14 tfinal=273+24;
15 l=68.5;
16 m=2.35;//mass flow rate
17 //calculation
18 A=pi*r^2;
19 A=round(A*10^5)/10^5;
```

```

20 rho=mmm/mV;
21 rho24=mmm*Pfinal*273/mV/101.3/tfinal;//density at
    24'C
22 u=m/rho24/A;
23 Re=u*rho24*2*r/mu;
24 //from graph
25 phi=0.0032;
26 //for solving using fsolve we copy numerical value
    of constant terms
27 //using back calculation
28 //as pressure maintained should be more than Pfinal
    so guessed value is Pfinal;
29 function [y]=eqn(x)
30     y=m^2/A^2*log(x/Pfinal)+(Pfinal^2-x^2)/2/R/
        tfinal*mmm+4*phi*1/2/r*m^2/A^2;
31 endfunction
32 [x,v,info]=fsolve(560*10^3,eqn);
33 disp(x/1000,"pressure maintained at compressor in (
    kN/m^2):");

```

---

Scilab code Exa 2.1.2 flow of ideal gas at maximum velocity under isothermal and a

```

1 //exapple 2.2
2 clc; funcprot(0);
3 // Initialization of Variable
4 M=28.8/1000;
5 mu=1.73/10^5;
6 gamm=1.402;
7 P1=107.6*10^3;
8 l = 68.5;
9 pi = %pi;
10 V=22.414/1000;
11 R=8.314;
12 temp=285;
13 d=4/1000;

```

```

14 rr=0.0008;
15 phi=0.00285;
16 //calculation
17 //constant term of equation
18 //part1
19 a=1-8*phi*l/d;//constant term in deff
20 deff('y=f(x)', 'y=log(x^2)-x^2+2.938');
21 [x,v,info]=fsolve(1,f);
22 z=1/x;
23 z=round(z*1000)/1000;
24 disp(z,"ratio of Pw/P1");
25 //part2
26 Pw=z*P1;
27 nuw=V*P1*temp/Pw/M/273;
28 Uw=sqrt(nuw*Pw);
29 disp(Uw,"maximum velocity in (m/s):")
30 //part3
31 Gw=pi*d^2/4*Pw/Uw;
32 disp(Gw,"maximum mass flow rate in(kg/s):");
33 //part4
34 G=2.173/1000;
35 J=G*Uw^2/2;
36 disp(J,"heat taken up to maintain isothermal
    condition(J/s):");
37 //part5
38 nu2=2.79;//found from graph
39 nu1=R*temp/M/P1;
40 P2=P1*(nu1/nu2)^gamm;
41 disp(P2/P1,"critical pressure ratio in adiabatic
    condition:");
42 //part6
43 Uw=sqrt(gamm*P2*nu2);
44 disp(Uw,"velocity at adiabatic condition in (m/s):")
    ;
45 //part7
46 Gw=pi*d^2/4*Uw/nu2;
47 disp(Gw,"mass flow rate at adiabatic condition in (
    kg/s):");

```

```

48 //part8
49 //polynomial in T of the form ax^2+bx+c=0;
50 c=gamm/(gamm-1)*P1*nu1+.5*Gw^2/pi^2/d^4*16*nu1^2;
51 b=gamm/(gamm-1)*R/M;
52 a=.5*Gw^2/pi^2/d^4*16*(R/M/P2)^2;
53 y=poly([-c b a], 'x', 'coeff');
54 T2=roots(y);
55 disp(T2(2)-273,"temperature of discharging gas in (
    Celcius)");

```

---

Scilab code Exa 2.1.3 flow of a non ideal gas at maximum velocity under adiabatic

```

1
2
3 //exapple 2.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 //1 refer to initial condition
7 R=8.314;
8 P1=550*10^3;
9 T1=273+350;
10 M=18/1000;
11 d=2.4/100;
12 pi=3.1428;
13 A=pi*d^2/4;
14 gamm=1.33;
15 roughness=0.096/1000/d;
16 l=0.85;
17 phi=0.0035//assumed value of friction factor
18 //calculation
19 nu1=R*T1/M/P1;
20 Pw=0.4*P1;//estimation
21 nuw=(P1/Pw)^0.75*nu1;
22 enthalpy=3167*1000;
23 Gw=sqrt(enthalpy*A^2/(gamm*nuw^2/(gamm-1)-nu1^2/2-

```

```

    nuw^2/2));
24 function [y]=eqn(x)
25   y=log(x/nu1)+(gamm-1)/gamm*(enthalpy/2*(A/Gw)^2*(1/
    x^2-1/nu1^2)+0.25*(nu1^2/x^2-1)-.5*log(x/nu1))
    +4*phi*l/d;
26 endfunction
27 deff('y=f(x)', 'eqn');
28 [x,v,info]=fsolve(0.2,eqn);
29
30 if x~=nuw then
31     disp("we again have to estimate Pw/P1");
32     disp("new estimate assumed as 0.45")
33     Pw=0.45*P1;//new estimation
34     nuw=(P1/Pw)^0.75*nu1;
35 // & we equalise nu2 to nuw
36 nu2=nuw;
37 Gw=sqrt(enhalpy*A^2/(gamm*nuw^2/(gamm-1)-nu1^2/2-
    nuw^2/2));
38 printf("mass flow rate of steam through pipe (kg/s):
    %.2f",Gw);
39 //part 2
40 disp(Pw/1000,"pressure of pipe at downstream end in
    (kPa):");
41
42 else
43     disp("our estimation is correct");
44
45 end
46 //part3
47 enthalpyw=2888.7*1000;//estimated from steam table
48 Tw=sqrt((enthalpy-enthalpyw+.5*Gw^2/A^2*nu1^2)*2*A
    ^2/Gw^2/R^2*M^2*Pw^2);
49 disp(Tw-273,"temperature of steam emerging from pipe
    in (Celcius):")

```

---

### Scilab code Exa 2.1.4 venting of gas from pressure vessel

```
1 //exapple 2.4
2 clc; funcprot(0);
3 // Initialization of Variables
4 pi = %pi;
5 M=28.05/1000;
6 gamm=1.23;
7 R=8.314;
8 atm=101.3*1000;
9 P1=3*atm;
10 //calculation
11 //part1
12 P2=P1*(2/(gamm+1))^(gamm/(gamm-1));
13 disp(P2/1000,"pressure at nozzle throat (kPa):")
14 //part2
15 temp=273+50;
16 nu1=R*temp/P1/M;
17 G=18; //mass flow rate
18 nu2=nu1*(P2/P1)^(-1/gamm);
19 A=G^2*nu2^2*(gamm-1)/(2*gamm*P1*nu1*(1-(P2/P1)^((gamm-1)/gamm)));
20 d=sqrt(4*sqrt(A)/pi);
21 disp(d*100,"diameter required at nozzle throat in (cm)")
22 //part3
23 vel=sqrt(2*gamm*P1*nu1/(gamm-1)*(1-(P2/P1)^((gamm-1)/gamm)));
24 disp(vel,"sonic velocity at throat in(m/s):");
```

---

### Scilab code Exa 2.1.5 gas flow measurement with veturimeter

```
1 //exapple 2.5
2 clc; funcprot(0);
3 // Initialization of Variables
```

```

4 pi = %pi;
5 T=273+15;
6 rho=999;
7 rhom=13559; //density of mercury
8 g=9.81;
9 P2=764.3/1000*rhom*g;
10 R=8.314;
11 M=16.04/1000;
12 d=4.5/1000;
13 A=pi*d^2/4;
14 G=0.75/1000; //mass flow rate
15 delP=(1-exp(R*T*G^2/2/P2^2/M/A^2))*P2;
16 h=-delP/rho/g;
17 disp(h*100,"height of manometer in (cm)")

```

---

Scilab code Exa 2.1.6 pressure drop required for flow of a gas liquid mixture thro

```

1 //exapple 2.6
2 clc; funcprot(0);
3 // Initialization of Variables
4 pi = %pi;
5 rho1=931;
6 mu=1.55/10000; //viscosity of water
7 Vsp=0.6057; //specific volume
8 T=273+133;
9 mug=1.38/100000; //viscosity of steam
10 P=300*1000;
11 d=0.075;
12 Gg=0.05; //mass flow gas phase
13 Gl=1.5; //mass flow liquid phase
14 A=pi*d^2/4;
15 //calculation
16 rhog=1/Vsp;
17 rhog=round(rhog*1000)/1000;
18 velg=Gg/A/rhog;

```

```

19 velg=round(velg*100)/100;
20 Reg=rhog*velg*d/mug;
21 //using chart
22 phig=0.00245;//friction factor gas phase
23 l=1;
24 delPg=4*phig*velg^2*rhog/d;
25 //consider liquid phase
26 vell=G1/A/rhol;
27 Rel=rhol*vell*d/mu;
28 if Rel>4000 & Reg>4000 then
29     disp("both liquid phase and solid phase in
          turbulent motion");
30     //from chart
31 end
32 PHIg=5;
33 delP=PHIg^2*delPg;
34 disp(delP,"required pressure drop per unit length in
        (Pa)")

```

---

# Chapter 3

## velocity boundary layers

Scilab code Exa 3.1.1 streamline flow over a flat plate

```
1
2
3 //exapple 3.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=998;
7 mu=1.002/1000;
8 x=48/100;
9 u=19.6/100;
10 x1=30/100;
11 b=2.6;
12 //calculation
13 //part1
14 disp("fluid in boundary layer would be entirely in
      streamline motion ");
15 Re=rho*x*u/mu;
16 printf("reynolds no is %.2e",Re);
17 //part 2
18 Re1=rho*x1*u/mu;
19 delta=x1*4.64*Re1^-.5;
20 disp(delta*1000,"boundary layer width in (mm):");
```

```

21 //part3
22 y=0.5*delta;//middle of boundary layer
23 ux=3/2*u*y/delta-.5*u*(y/delta)^3;
24 disp(ux*100,"velocity of water in (cm/s):");
25 //part4
26 R=0.323*rho*u^2*Re1^-0.5;
27 disp(R,"shear stress at 30cm in (N/m^2):");
28 //part5
29 Rms=0.646*rho*u^2*Re^-0.5;
30 disp(Rms,"mean shear stress experienced over whole
    plate in (N/m^2)");
31 //part6
32 F=Rms*x*b;
33 disp(F,"total force experienced by the plate in (N)"
    )

```

---

### Scilab code Exa 3.1.2 turbulent flow over a plate

```

1
2
3 //exapple 3.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 P=102.7*1000;
7 M=28.8/1000;
8 R=8.314;
9 temp=273+18;
10 Recrit=10^5;
11 u=18.4;
12 b=4.7;//width
13 x=1.3;
14 mu=1.827/100000;
15 //calculation
16 //part1
17 rho=P*M/R/temp;

```

```

18 xcrit=Recrit*mu/rho/u;
19 a=1-xcrit/1.65;
20 disp(a*100,"% of surface over which turbulent
    boundary layer exist is :");
21 //part2
22 Rex=rho*u*x/mu;
23 thik=0.375*Rex^-.2*x;
24 disp(thik*100,"thickness of boundary layer in (cm):"
    );
25 y=0.5*thik;
26 ux=u*(y/thik)^(1/7);
27 disp(ux,"velocity of air at mid point is (m/s):")
28 //part4
29 lthik=74.6*Rex^-.9*x;
30 disp(lthik*1000,"thickness of laminar boundary layer
    in (mm):");
31 //part5
32 ub=u*(lthik/thik)^(1/7);
33 disp(ub,"velocity at outer edge of laminar sublayer
    in (m/s):");
34 //part6
35 R=0.0286*rho*u^2*Rex^-0.2;
36 disp(R,"shearforce expericenced in (N/m^2):");
37 //part7
38 x1=1.65;//length of plate
39 Rex1=rho*u*x1/mu;
40 Rms=0.0358*rho*u^2*Rex1^-0.2;
41 disp(Rms,"mean shearforce in (N/m^2):");
42 //part8
43 F=x1*Rms*b;
44 disp(F,"total drag force expericenced by the plate
    is (N):");

```

---

Scilab code Exa 3.1.3 streamline and turbulent flow through and equations of unive

```

1
2
3 //exapple 3.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 Q=37.6/1000000;
7 d=3.2/100;
8 mu=1.002/1000;
9 rho=998;
10 pi=3.14;
11 //calculation
12 //part1
13 u=4*Q/pi/d^2;
14 Re=rho*u*d/mu;
15 disp(Re,"pipe flow reynolds no :");
16 disp("Water will be in streamline motion in the pipe
      ");
17 //part2
18 a=-8*u/d;
19 disp(a,"velocity gradient at the pipe wall is (s^-1)
      :");
20 //part3
21 Ro=-mu*a;
22 printf("Sherastress at pipe wall is (N/m^2) %.2e",Ro
      );
23 //part4
24 Q=2.10/1000;
25 u=4*Q/pi/d^2;
26 u=round(u*1000)/1000;
27 disp(u,"new av. fluid velocity is (m/s):");
28 Re=rho*u*d/mu;
29 phi=0.0396*Re^-0.25;//friction factor
30 phi=round(phi*10^5)/10^5;
31 delb=5*d*Re^-1*phi^-0.5;
32 disp(delb*10^6,"thickness of laminar sublayer in
      (10^-6m):");
33 //part5
34 y=30*d/phi^0.5/Re;//thickness

```

```

35 tbl=y-delb;
36 disp(tbl*1000,"thickness of buffer layer in (mm):");
37 //part6
38 A=pi*d^2/4;//cross sectional area of pipe
39 dc=d-2*y;//dia of turbulent core
40 Ac=pi*dc^2/4;
41 p=(1-A/Ac)*100;
42 disp(p,"percentage of pipe-s core occupied by
    turbulent core is (%):");
43 //part7
44 uplus=5;//from reference
45 ux=uplus*u*phi^0.5;
46 disp(ux,"velocity where sublayer and buffer layer
    meet is (m/s):");
47 //part8
48 yplus=30;//from reference
49 ux2=u*phi^0.5*(2.5*log(yplus)+5.5);
50 disp(ux2,"velocity where turbulent core and buffer
    layer meet is (m/s):");
51 //part9
52 us=u/0.81;
53 disp(us,"fluid velocity along the pipe axis (m/s):")
    ;
54 //part10
55 Ro=phi*rho*u^2;
56 disp(Ro,"shearstress at pipe wall (N/m^2):");

```

---

# Chapter 4

## Flow Measurement

Scilab code Exa 4.1.1 use of pitot tube to measure flow rate

```
1 //exapple 4.1
2 clc; funcprot(0);
3 // Initialization of Variables
4 pi = %pi;
5 rho=998;
6 rhom=1.354*10^4;//density of mercury
7 M=2.83/100;
8 mu=1.001/1000;
9 mun=1.182/10^5;//viscosity of natural gas
10 R=8.314;
11 g=9.81;
12 h=28.6/100;
13 d=54/100;
14 //part1
15 nu=1/rho;
16 delP=h*g*(rhom-rho);
17 umax=sqrt(2*nu*delP);
18 umax=round(umax*10)/10;
19 disp(umax,"maximum fluid velocity in (m/s)");
20 Re=umax*d*rho/mu;
21 printf("reynold no. is %.2e",Re);
```

```

22 //using chart
23 u=0.81*umax;
24 G=rho*pi*d^2/4*u;
25 disp(G,"mass flow rate in (kg/s):");
26 disp(G/rho,"Volumetric flow rate in (m^3/s):");
27 //part2
28 P1=689*1000;//initial pressure
29 T=273+21;
30 nu1=R*T/M/P1;
31 nu1=round(nu1*10000)/10000;
32 rhog=1/nu1;//density of gas
33 h=17.4/100;
34 P2=P1+h*(rho-rhog)*g;
35 P2=round(P2/100)*100;
36 umax2=sqrt(2*P1*nu1*log(P2/P1));
37 disp(umax2,"maximum fluid velocity in (m/s)");
38 Re=rhog*umax2*d/mun;
39 printf("reynold no. is %.3e",Re);
40 //from table
41 u=0.81*umax2;
42 Q=pi*d^2/4*u;
43 disp(Q,"volumetric flow rate is (m^3/s):");
44 disp(Q*rhog,"mass flow rate in (kg/s):")

```

---

Scilab code Exa 4.1.2 use of pitot tube to measure flow of gas

```

1
2
3 //exapple 4.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 rd=[0 1 2.5 5 10 15 17.5]/100;//radial distance from
   pipe
7 dlv=[0 0.2 0.36 0.54 0.81 0.98 1]/100;//differrnce in
   liquid levels

```

```

8 r=[.175 .165 .150 .125 .075 .025 0];//
9 g=9.81;
10 R=8.314;
11 rho=999;
12 temp=289;
13 P1=148*1000;
14 M=7.09/100;
15 pi=3.12
16 rhoCl2=P1*M/R/temp;//density of Cl2
17 nuCl2=1/rhoCl2;//specific volume of Cl2
18 function [y]=P2(x);
19     y=P1+x*(rho-rhoCl2)*g;
20 endfunction
21 for i=1:7
22     y=P2(dlv(i));
23     u(i)=sqrt(2*P1*nuCl2*log(y/P1));
24     a(i)=u(i)*r(i);
25 end
26 clf();
27 plot(r,a);
28 xtitle("","r (m)","u*r (m^2/s)");
29 s=0;
30 for i=1:6//itegration of the plotted graph
31     s=abs((r(i)-r(i+1))*0.5*(a(i)+a(i+1)))+s;
32 end
33 s=s-0.01;
34 Q=2*pi*s;
35 disp(Q,"volumetric flow rate (m^3/s):");
36 disp(Q*rhoCl2,"mass flow rate of chlorine gas (kg/s)
    ")

```

---

Scilab code Exa 4.1.3 use of orifice and manometer to measure flow

1  
2

```

3 //exapple 4.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 pi=3.14;
7 Cd=0.61;
8 rho=999;
9 rhoo=877;//density of oil
10 g=9.81;
11 h=75/100;
12 d=12.4/100;//dia of orifice
13 d1=15/100;//inside diameter
14 nuo=1/rhoo;//specific volume of oil
15 //calculation
16 //part1
17 delP=h*(rho-rhoo)*g;
18 A=pi*d^2/4;
19 G=Cd*A/nuo*sqrt(2*nuo*delP/(1-(d/d1)^4));
20 disp(G,"mass flow rate in (kg/s)")
21 //part2
22 h=(1+0.5)*d1;
23 delP=rhoo/2*(G*nuo/Cd/A)^2*(1-(d/d1)^4)+h*rhoo*g;
24 disp(delP,"pressuer differnce between tapping points
    ");
25 delh=(delP-h*rhoo*g)/(rho-rhoo)/g;
26 disp(delh,"difference in water levels in manometer i
    (cm)")

```

---

Scilab code Exa 4.1.4 determination of orifice size for flow measurement and press

```

1
2
3 //exapple 4.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 rhom=1.356*10^4;//density mercury

```

```

7  rhon=1266; //density NaOH
8  Cd=0.61;
9  g=9.81;
10 Cdv=0.98; //coeff. of discharge of venturimeter
11 Cdo=Cd; //coeff. of discharge of orificemeter
12 d=6.5/100;
13 pi=3.14;
14 A=pi*d^2/4;
15 Q=16.5/1000;
16 h=0.2; //head differnce
17 //calculation
18 //part1
19 delP=g*h*(rhom-rhon);
20 G=rhon*Q;
21 nun=1/rhon; //specific volume of NaOH
22 Ao=G*nun/Cd*sqrt(1/(2*nun*delP+(G*nun/Cd/A)^2)); //
    area of orifice
23 d0=sqrt(4*Ao/pi)
24 disp(d0*100,"diameter of orifice in (cm):");
25 //part2
26 a=(Cdv/Cdo)^2;
27 disp(a,"ratio of pressure drop ")

```

---

Scilab code Exa 4.1.5 use of rotatometer for flow measurement

```

1 //exapple 4.5
2 clc; funcprot(0);
3 // Initialization of Variables
4 pi = %pi;
5 M=3.995/100;
6 g=9.81;
7 R=8.314;
8 Cd=0.94;
9 temp=289;
10 df=9.5/1000; //diameter of float

```

```

11 Af=pi*df^2/4; //area of float
12 P=115*10^3;
13 V=0.92/10^6;
14 rhoc=3778; //density of ceramic
15 //calculation
16 rho=P*M/R/temp;
17 nu=1/rho;
18 P=V*(rhoc-rho)*g/Af;
19 disp(P,"pressure drop over the float in (Pa):");
20 //part2
21 x=.15/25*(25-7.6);
22 L=df*100+2*x;
23 L=L/100;
24 A1=pi*L^2/4;
25 A0=A1-Af;
26 G=Cd*A0*sqrt(2*rho*P/(1-(A0/A1)^2));
27 printf("mass flow rate in (kg/s) is %.3e",G);
28 Q=G/rho;
29 disp(Q,"Volumetric flow rate in (m^3/s):")

```

---

Scilab code Exa 4.1.6 mass of float required to measure fluid rate in rotatometer

```

1
2
3 //exapple 4.6
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=999;
7 rhos=8020; //density of steel
8 g=9.81;
9 pi=3.14;
10 df=14.2/1000; //dia of float
11 Af=pi*df^2/4; //area of float
12 Cd=0.97;
13 nu=1/rho;

```

```
14 Q=4/1000/60;
15 G=Q*rho;
16 // calculation
17 x=0.5*(18.8-df*1000)/280*(280-70);
18 L=df*1000+2*x;
19 L=L/1000;
20 A1=pi*L^2/4;
21 A0=A1-Af;
22 Vf=Af/g/(rhos-rho)/2/nu*(G*nu/Cd/A0)^2*(1-(A0/A1)^2)
    ;
23 m=Vf*rhos;
24 disp(m*1000,"mass of float equired in (g):")
```

---

# Chapter 5

## Flow measurement in open channel

Scilab code Exa 5.1.1 use of manning and chezy formulae

```
1
2
3 //exapple 5.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=999.7;
7 g=9.81;
8 mu=1.308/1000;
9 s=1/6950;
10 b=0.65;
11 h=32.6/100;
12 n=0.016;
13 // calculation
14 //part1
15 A=b*h;
16 P=b+2*h;
17 m=A/P;
18 u=s^.5*m^(2/3)/n;
19 Q=A*u
```

```

20 disp(Q,"volumetric flow rate (m^3/s):");
21 C=u/m^0.5/s^0.5;
22 disp(C,"chezy coefficient (m^0.5/s):");
23 a=-m*rho*g*s/mu;//delu/dely
24 disp(a,"velocity gradient in the channel (s^-1):")

```

---

#### Scilab code Exa 5.1.2 stream depth in trapezoid channel

```

1
2
3 //exapple 5.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 Q=0.885;
7 pi=3.1428;
8 s=1/960;
9 s=round(s*1000000)/1000000;
10 b=1.36;
11 n=0.014;
12 theta=55*pi/180;
13 // calculation
14 function [y]=flow(x);
15     a=(x*(b+x/tan(theta)))/(b+2*x/sin(theta));
16     y=a^(2/3)*s^(1/2)*(x*(b+x/tan(theta)))/n-Q;
17 endfunction
18 x=fsolve(0.1,flow);
19 disp(x,"depth of water in (m):")

```

---

#### Scilab code Exa 5.1.3 optimum base angle of a Vshaped channel Slope of a channel

```

1
2
3 //exapple 5.3

```

```

4 clc; funcprot(0);
5 // Initialization of Variable
6 n=0.011;
7 h=0.12;
8 Q=25/10000;
9 // calculation
10 deff('y=f(x)', 'y=1/x^2-1');
11 x=fsolve(0.1,f);
12 theta=2*atan(x);
13 A=h*2*h/tan(theta/2)/2;
14 P=2*h*sqrt(2);
15 s=Q^2*n^2*P^(4/3)/A^(10/3);
16 disp(s,"the slope of channel in (radians):")

```

---

Scilab code Exa 5.1.4 stream depth and maximum velocity and flow rate in a pipe

```

1
2
3 //exapple 5.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 //part1
7 //maximizing equation in theta & get a function
8 function [y]=theta(x)
9     y=(x-.5*sin(2*x))/2/x^2-(1-cos(2*x))/2/x;
10 endfunction
11 x=fsolve(2.2,theta);
12 x=round(x*1000)/1000;
13 a=(1-cos(x))/2;
14 printf("velocity will be maximum when stream depth
    in times of diameter is %.3f",a);
15 //part2
16 //maximizing equation in theta & get a function
17 function [y]=theta2(x)
18     y=3*(x-.5*sin(2*x))^2*(1-cos(2*x))/2/x-(x-.5*sin

```

```

                (2*x))^3/2/x^2 ;
19 endfunction
20 x1=fsolve(2.2,theta2);
21 x1=round(x1*1000)/1000;
22 a=(1-cos(x1))/2;
23 disp("")
24 printf("volumetric flow will be maximum when stream
        depth in times of diameter is %.3f",a);
25 //part3
26 r=1;
27 A=1*x-0.5*sin(2*x);
28 s=0.35*3.14/180;
29 P=2*x*r;
30 C=78.6;
31 u=C*(A/P)^0.5*s^0.5;
32 disp(u,"maximum velocity of obtained fluid (m/s):");
33 //part4
34 disp(x1,"maximum flow rate obtained at angle in (
        radians):")

```

---

#### Scilab code Exa 5.1.5 flow measurement with sharp crested weir

```

1
2
3 //exapple 5.5
4 clc; funcprot(0);
5 // Initialization of Variable
6 g=9.81;
7 h=28/100;
8 Cd=0.62;
9 B=46/100;
10 Q=0.355;
11 n=2;//from francis formula
12 //calcualtion
13 //part1

```

```

14 u=sqrt(2*g*h);
15 disp(u,"velocity of fluid (m/s):");
16 //part2a
17 H=(3*Q/2/Cd/B/(2*g)^0.5)^(2/3);
18 disp(H,"fluid depth over weir in (m):");
19 //part2b
20 //using francis formula
21 function [y]=root(x)
22     y=Q-1.84*(B-0.1*n*x)*x^1.5;
23 endfunction
24 x=fsolve(0.2,root);
25 disp(x,"fluid depth over weir in if SI units used in
      (m):");
26 //part3
27 H=18.5/100;
28 Q=22/1000;
29 a=15*Q/8/Cd/(2*g)^0.5/H^2.5;
30 theta=2*atan(a);
31 disp(theta*180/3.14,"base angle of the notch of weir
      (degrees)")

```

---

Scilab code Exa 5.1.6 equation of specific energy and analysis of tranquil and sho

```

1
2
3 //exapple 5.6
4 clc; funcprot(0);
5 // Initialization of Variable
6 Q=0.675;
7 B=1.65;
8 D=19.5/100;
9 g=9.81;
10 //caculation
11 u=Q/B/D;
12 u=round(u*1000)/1000;

```

```

13 E=D+u^2/2/g;
14 y=poly([8.53/1000 0 -E 1], 'x', 'coeff');
15 x=roots(y);
16 disp(x(1), "alternative depth in (m)");
17 disp("It is shooting flow");
18 Dc=2/3*E;
19 Qmax=B*(g*Dc^3)^0.5;
20 disp(Qmax, "maximum volumetric flow (m^3/s)");
21 Fr=u/sqrt(g*D);
22 disp(Fr, "Froude no.");
23 a=(E-D)/E;
24 disp(a*100, "% of kinetic energy in initial system");
25 b=(E-x(1))/E;
26 disp(b*100, "% of kinetic energy in final system");

```

---

Scilab code Exa 5.1.7 alternate depth of stream gradient of mild and steep slope

```

1
2
3 //exapple 5.7
4 clc; funcprot(0);
5 // Initialization of Variable
6 G=338; //mass flow rate
7 rho=998;
8 q=G/rho;
9 E=0.48;
10 n=0.015;
11 g=9.81;
12 B=0.4;
13 y=poly([5.85/1000 0 -E 1], 'x', 'coeff');
14 x=roots(y);
15 disp(x(1), x(2), "alternate depths (m):");
16 s=(G*n/rho/x(2)/(B*x(2)/(B+2*x(2)))^(2/3))^2
17 disp(s, "slode when depth is 12.9cm");
18 s=(G*n/rho/x(1)/(B*x(1)/(B+2*x(1)))^(2/3))^2

```

```
19 disp(s,"slode when depth is 45.1cm");
```

---

### Scilab code Exa 5.1.8 critical flw condition

```
1
2
3 //exapple 5.8
4 clc; funcprot(0);
5 // Initialization of Variable
6 pi=3.14;
7 theta=pi/3;
8 h=1/tan(theta);
9 B=0.845;
10 E=0.375;
11 g=9.81;
12 // calculation
13 //part1
14 //deducing a polynomial(quadratic) in Dc
15 a=5*h;
16 b=3*B-4*h*E;
17 c=-2*E*B;
18 y=poly([c b a], 'x', 'coeff');
19 x=roots(y);
20 disp(x(2),"critical depth in (m):");
21 //part2
22 Ac=x(2)*(B+x(2)*tan(theta/2));
23 Btc=B+x(2)*tan(theta/2)*2;
24 Dcbar=Ac/Btc;
25 uc=sqrt(g*Dcbar);
26 disp(uc,"critical velocity (m/s):");
27 //part3
28 Qc=Ac*uc;
29 disp(Qc,"Critical volumetric flow (m^3/s):");
```

---

Scilab code Exa 5.1.9 flow measurement with broad crested weir

```
1
2
3 //exapple 5.9
4 clc; funcprot(0);
5 // Initialization of Variable
6 B2=1.60;//breadth at 2
7 D2=(1-0.047)*1.27;//depth at 2
8 g=9.81;
9 B1=2.95;//breadth at 1
10 D1=1.27;//depth at 1
11 Z=0;
12 //calculation
13 Q=B2*D2*(2*g*(D1-D2-Z)/(1-(B2*D2/B1/D1)^2))^0.5;
14 disp(Q,"volumetric flow rate over flat topped weir
      over rectangular section in non uniform width(m
      ^3/s)");
15 //next part
16 B2=12.8;
17 D1=2.58;
18 Z=1.25;
19 Q=1.705*B2*(D1-Z)^1.5;
20 disp(Q,"volumetric flow rate over flat topped weir
      over rectangular section in uniform width (m^3/s)
      :")
```

---

Scilab code Exa 5.1.10 gradually varied flow behind a weir

```
1 //exapple 5.10
2 clc; funcprot(0);
3 // Initialization of Variable
```

```

4 g = 9.8;
5 pi=%pi;
6 n=0.022;
7 B=5.75;
8 s=0.15*pi/180;
9 Q=16.8;
10 function [y]=normal(x)
11     y=Q-B*x/n*(B*x/(B+2*x))^(2/3)*s^0.5;
12 endfunction
13 x=fsolve(1.33,normal);
14 disp(x,"Normal depth in (m):");
15 Dc=(Q^2/g/B^2)^(1/3);
16 disp(Dc,"Critical depth in (m):");
17 delD=.1;
18 D=1.55:.1:2.35
19 su=0;
20 for i=1:9
21     delL=delD/s*(1-(Dc/D(i))^3)/(1-(x/D(i))^3.33);
22     su=su+delL
23 end
24 disp(su,"distance in (m) from upstream to that place
      :")

```

---

#### Scilab code Exa 5.1.11 analysis of hydraulic jump

```

1
2
3 //exapple 5.11
4 clc; funcprot(0);
5 // Initialization of Variable
6 g=9.81;
7 q=1.49;
8 pi=3.14;
9 // calculation
10 //part1

```

```

11 Dc=(q^2/g)^.333;
12 disp(Dc,"critical depth in (m):");
13 //part2
14 n=0.021;
15 su=1.85*pi/180;//slope upstream
16 sd=0.035*pi/180;//slope downstream
17 Dnu=(n*q/sqrt(su))^(3/5);
18 Dnu=round(Dnu*1000)/1000;
19 disp(Dnu,"normal depth upstream in (m):");
20 Dnd=(n*q/sqrt(sd))^(3/5);
21 disp(Dnd,"normal depth downstream in (m):");
22 //part3
23 D2u=-0.5*Dnu*(1-sqrt(1+8*q^2/g/Dnu^3));
24 D2u=round(D2u*1000)/1000;
25 disp(D2u,"conjugate depth for upstream in (m):");
26 D1d=-0.5*Dnd*(1-sqrt(1+8*q^2/g/Dnd^3));
27 disp(D1d,"conjugate depth for downstream in (m):");
28 //part4
29 //accurate method
30 delD=.022;
31 D=0.987:.022:1.141
32 dis=0;
33 for i=1:8
34     delL=delD/su*(1-(Dc/D(i))^3)/(1-(Dnu/D(i))^3.33)
35     ;
36     dis=dis+delL
37 end
38 disp(dis,"distance in (m) of occurrence of jump by
39     accurate method:");
40 //not so accurate one
41 E1=D2u+q^2/2/g/D2u^2;
42 E2=Dnd+q^2/2/g/Dnd^2;
43 E2=round(E2*1000)/1000;
44 E1=round(E1*1000)/1000;
45 ahm=(D2u+Dnd)/2;//av. hydraulic mean
46 afv=.5*(q/D2u+q/Dnd);//av. fluid velocity
47 i=(afv*0.021/ahm^(2/3))^2;
48 l=(E2-E1)/(su-i+0.0002);

```

```
47 disp(1,"distance in (m) of occurrence of jump by not
    so accurate method:")
48 //part5
49 rho=998;
50 Eu=Dnu+q^2/2/g/Dnu^2;
51 Eu=round(Eu*1000)/1000;
52 P=rho*g*q*(Eu-E1);
53 disp(P/1000,"power loss in hydraulic jump per unit
    width in (kW):")
```

---

# Chapter 6

## pumping of liquids

Scilab code Exa 6.1.1 cavitation and its avoidance in suction pipes

```
1
2 //example 6.1
3 clc; funcprot(0);
4 //exapple 6.1
5 // Initialization of Variable
6 atp=100.2*1000;
7 g=9.81;
8 rho_w=996;
9 rho_toluene=867;
10 vap_pre_toluene=4.535*1000;
11 viscosity_toluene=5.26/10000;
12 //calculation
13 m=(atp-vap_pre_toluene)/rho_toluene/g;
14 disp(m,"Max. height of toluene supported by atm.
    pressure (in m):");
15 //part(1)
16 hopw=0.650;//head of pump in terms of water
17 hopt=hopw*rho_w/rho_toluene;//head of pump in terms
    of toluene
18 Q=1.8*10^-3;//flow in m^3/s
19 d=2.3*10^-2;//diameter of pipe
```

```

20 pi=3.14127;
21 //u=4*Q/pi/d^2
22 //substituting this for reynolds no.
23 Re=4*Q*rho_toluene/pi/d/viscosity_toluene;//reynolds
    no.
24 disp(Re ,”reynolds no :”);
25 phi=0.0396*Re^-0.25;
26 //since both LHS and RHS are function of x(max. ht.
    ab. toluene)
27 //we define a new variable to solve the eqn
28 //y=(atp/rho_toluene/g)-(vap_pre_toluene/rho_toluene
    /g)-(4*phi*16*Q^2*x/pi^2/d^5/g)-hopt;
29 //y=x
30 //these are two equations
31 b=[0;((atp/rho_toluene/g)-(vap_pre_toluene/
    rho_toluene/g)-hopt)];
32 A=[1 -1;1 4*phi*16*Q^2/pi^2/d^5/g];
33 x=A\b;
34 disp(x(2,1), ”the maximum height above toluene in
    the tank the pump can be located without risk
    while flow rate is 1.80dm^3/s (in m):”);
35 //solution of part(2)
36 l=9//length
37 u=sqrt(((atp/rho_toluene/g)-(vap_pre_toluene/
    rho_toluene/g)-hopt-l)*d*g/4/phi/l);//fluid
    velocity in pipes
38 Q=pi*d^2*u/4;
39 disp(Q,”Maximum delivery rate if pump is located 9m
    above toluene tank(in m^3/s)”)
40 //solution of part(3)
41 //clubing d together we get
42 Q=1.8/1000;
43 a=(atp/rho_toluene/g)-(vap_pre_toluene/rho_toluene/g
    )-hopt-l;
44 b=a*pi^2*g/4/9/16/Q^2/0.0396/(4*Q*rho_toluene/pi/
    viscosity_toluene)^-0.25;
45 d=(1/b)^(1/4.75);
46 disp(d , ”minimum smooth diameter of suction pipe

```

which will have flow rate as (1.8 dm<sup>3</sup>/s) for pump kept at 9 m high (in m):”);

---

### Scilab code Exa 6.1.2 specific speed of a centrifugal pump

```
1
2 //example 6.2
3 clc; funcprot(0);
4 //exapple 6.2
5 // Initialization of Variable
6 Q1=24.8/1000;//flow in pump 1
7 d1=11.8/100;//diameter of impeller 1
8 H1=14.7//head of pump 1
9 N1=1450//frequency of motor 1
10 Q2=48/1000//flow in pump 2
11 //calculation
12 H2=1.15*H1;//head of pump 2
13 specific_speed=N1*Q1^0.5/H1^0.75;
14 N2=specific_speed*H2^0.75/Q2^0.5;//frequency of
    motor 2
15 disp(N2 ,”frequency of motor 2 in rpm”);
16 d2=sqrt(N2^2*H1/H2/N1^2/d1^2);
17 disp(1/d2 , ”diametr of impeller 2 (in m)”);
```

---

### Scilab code Exa 6.1.3 theoritical and effective characteristic of centrifugal pump

```
1
2 //example 6.3
3 clc; funcprot(0);
4 clf()
5 //exapple 6.3
6 // Initialization of Variable
7 Q=[0 0.01 0.02 0.03 0.04 0.05];//discharge
```

```

8 effi_hyd=[65.4 71 71.9 67.7 57.5 39.2];
9 effi_over=[0 36.1 56.0 61.0 54.1 37.0];
10 H_sys=[0 0 0 0 0 0]
11 d=0.114;//diameter of pipe
12 d_o=0.096;//diameter of impeller
13 h=8.75;//elevation
14 g=9.81;//acc. of gravity
15 rho=999;//denisity of water
16 l=60;//length of pipe
17 theta=0.611;//angle in radians
18 B=0.0125;//width of blades
19 pi=3.1412
20 mu=1.109/1000;//viscosity of water
21 omega=2*pi*1750/60;
22 // calculation
23 for i=1:6
24     if i==1 then
25         H_sys(i)=h;
26     else
27
28         H_sys(i)=h+8*Q(i)^2/pi^2/d^4/g*(1+8*1*0.0396/d
                *(4*rho*Q(i)/pi/d/mu)^-0.25);
29     end,
30 end;
31 H_theor=omega^2*d_o^2/g-omega*Q/2/pi/g/B/tan(theta);
32 //disp(H_sys"head of system (in m)");
33 //disp(H_theor);
34 for i=1:6
35     H_eff(i)=effi_hyd(i)*H_theor(i)/100;
36 end
37 //disp(H_eff);
38 plot(Q,effi_hyd, 'r—d');
39 plot(Q,effi_over, 'g');
40 plot(Q,H_eff,'k');
41 plot(Q,H_theor);
42 plot(Q,H_sys, 'c-');
43 title('system characteritics');
44 ylabel('Head(m) or Efficiency (%)');

```

```

45 xlabel('volumetric flow rate(m^3/s)');
46 //calculation of power
47 //at intersecting point using datatrip b/w H_sys &
   H_eff
48 Q=0.0336
49 effi_over=59.9
50 H_eff=13.10
51 P=H_eff*rho*g*Q/effi_over/10;
52 disp(P ,”Power required to pump fluid at this rate(
   in KW):”)

```

---

Scilab code Exa 6.1.4 flow rate when cetrifugal pumps operate singly and in parall

```

1  clc; funcprot(0);
2  clf()
3  //exapple 6.4
4  // Initialization of Variable
5  //each is increased by five units to make each
   compatible for graph plotting
6  mu=1.109/1000;
7  Q=[0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
   0.1]; //flow rate
8  HeffA=[20.63 19.99 17.80 14.46 10.33 5.71 0 0 0 0 0
   ]; //Heff of pump A
9  HeffB=[18 17 14.95 11.90 8.10 3.90 0 0 0 0 0]; //Heff
   of pump B
10 alpha=1;
11 h=10.4;
12 d=0.14;
13 l=98;
14 pi=3.1412;
15 g=9.81;
16 rho=999;
17 for i=1:11
18     if i==1 then

```

```

19     H_sys(i)=h;
20     else
21
22     H_sys(i)=h+8*Q(i)^2/pi^2/d^4/g*(1+8*1*0.0396/d
        *(4*rho*Q(i)/pi/d/mu)^-0.25);
23 end,
24 end;
25 //H_sys is head of the system
26 disp(H_sys, "the head of system in terms of height
        of water :");
27 plot(Q,H_sys, 'r—d');
28 plot(Q,HeffA, '-c');
29 plot(Q,HeffB);
30 //at intersecting point using datatrip b/w H_sys &
        amp; H_effA
31 disp(0.03339, "the flow rate at which H_sys takes
        over HeffA");

```

---

### Scilab code Exa 6.1.5 pumping with a reciprocating pump

```

1
2 //example 6.5
3 clc; funcprot(0);
4 //exapple 6.5
5 // Initialization of Variable
6 rho=1000;
7 dc=.15;
8 l=7.8;
9 g=9.81;
10 pi=3.1428;
11 atp=105.4*1000;
12 vap_pre=10.85*1000;
13 sl=.22;
14 dp=0.045;
15 h=4.6;

```

```

16 //("x(t)=s1/2*cos(2*pi*N*t)" "the function of
    displacement");
17 //”since we have to maximize the acceleration double
    derivate the terms”);
18 //since double derivation have the term cos(kt)
19 //finding it maxima
20 t=linspace(0,5,100);
21 k=1;
22 function [m,v]= maximacheckerforcosine()
23 h=0.00001;
24 a=0.00;
25 for i=1:400
26     if (cos(a+h)-cos(a-h))/2*h==0 & cos(i-1)>0 then
27 break;
28 else
29     a=0.01+a;
30 end
31 break;
32 end
33 m=i-1;
34 v=cos(i-1);
35 endfunction;
36 [a, b]= maximacheckerforcosine();
37 disp(a,”time t when the acceleration will be maximum
    (s)”);
38 //double derivative will result in a square of value
    of N
39 //lets consider its coefficient all will be devoid
    of N^2
40 k=s1/2*(2*pi)^2//accn max of piston
41 kp=k*1/4*pi*dc^2/1*4/pi/dp^2;//accn coeff. ofsuction
    pipe
42 f=1/4*pi*dp^2*1*rho*kp;//force exerted by piston
43 p=f/1*4/pi/dp^2;//pressure exerted by piston
44 //calculation
45 o=atp-h*rho*g-vap_pre;
46 //constant term of quadratic eqn
47 y=poly([o 0 -p], 'N', 'coeff')

```

```

48 a=roots(y);
49 disp(abs(a(1,1)), "Maximum frequency of oscillation
    if cavitation o be avoided(in Hz)");

```

---

### Scilab code Exa 6.1.6 pumping with a air lift pump

```

1
2 //example 6.6
3 clc; funcprot(0);
4 //exapple 6.6
5 // Initialization of Variable
6 rhos=1830;//density of acid
7 atp=104.2*1000;//atmospheric pressure
8 temp=11+273;//temp in kelvin
9 M=28.8/1000;//molar mass of air
10 R=8.314;//universal gas constant
11 g=9.81;//acceleration of gravity
12 pi=3.14;
13 d=2.45;//diameter of tank
14 l=10.5;//length of tank
15 h_s=1.65;//height of surface of acid from below
16 effi=0.93//efficiency
17 //calculation
18 mliq=pi*d^2*l*rhos/4;
19 h_atm=atp/rhos/g;//height conversion of atp
20 h_r=4.3-1.65;//height difference
21 mair=g*h_r*mliq*M/(effi*R*temp*log(h_atm/(h_atm+h_s)
    ));//mass of air
22 disp(mair,"mass of air required to lift the
    sulphuric acid tank");
23 disp("The negative sign indicates air is expanding &
    work done is magnitude of value in kg:");
24 m=abs(mair/mliq);
25 disp(m, "The mass of air required for per kilo of
    acid transferred:");

```



# Chapter 7

## Flow Through Packed Beds

Scilab code Exa 7.1.1 determination of particle size and specific surface area for

```
1
2
3 //exapple 7.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 mu=1.83/1000;
7 rhom=1.355*10000;//density mercury
8 K=5;
9 g=9.81;
10 d=2.5/100;
11 pi=3.14;
12 thik=2.73/100;
13 rho=3100;//density of particles
14 Q=250/(12*60+54)/10^6;
15 //calculation
16 A=pi*d^2/4;
17 Vb=A*thik;//volume of bed
18 Vp=25.4/rho/1000;//volume of particles
19 e=1-Vp/Vb;
20 u=Q/A;
21 delP=12.5/100*rhom*g;
```

```

22 S=sqrt(e^3*delP/K/u/thik/mu/(1-e)^2);
23 S=round(S/1000)*1000;
24 d=6/S;
25 disp(d*10^6,"average particle diameter in (x10^-6m)"
);
26 A=pi*d^2/1000/(4/3*pi*d^3/8*rho);
27 disp(A*10^4,"surface area per gram of cement (cm^2):
")

```

---

### Scilab code Exa 7.1.2 rate of flow through packed bed

```

1
2
3 //exapple 7.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 mu=2.5/1000;
7 rho=897;
8 g=9.81;
9 pi=3.1414;
10 K=5.1;
11 l=6.35/1000;
12 d=1;
13 hei=24.5+0.65;
14 len=24.5;
15 dc=2.65;//dia of column
16 thik=0.76/1000;
17 Vs=pi*d^2/4*l-pi*l/4*(d-2*thik)^2;//volume of each
ring
18 n=3.023*10^6;
19 e=1-Vs*n;
20 e=round(e*1000)/1000;
21 Surfacearea=pi*d*l+2*pi*d^2/4+pi*(d-2*thik)*l-2*pi*(
d-2*thik)^2/4;
22 S=Surfacearea/Vs;

```

```

23 S=round(S);
24 delP=hei*g*rho;
25 delP=round(delP/100)*100;
26 u=e^3*delP/K/S^2/mu/(1-e)^2/len;
27 Q=pi*dc^2/4*u;
28 disp(Q,"initial volumetric flow rate in (m^3/s):")

```

---

Scilab code Exa 7.1.3 determination of pressure drop to drive fluid through a pack

```

1
2
3 //exapple 7.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 dr=2; //dia of column
7 mu=2.02/10^5;
8 rho=998;
9 K=5.1;
10 g=9.81;
11 Q=10000/3600;
12 l=50.8/1000;
13 d=1;
14 n=5790;
15 len=18;
16 thik=6.35/1000;
17 pi=3.1414;
18 //part1
19 //calculation
20 CA=pi*dr^2/4; //cross sectional area
21 u=Q/CA;
22 Vs=pi*d^2/4*l-pi*l/4*(d-2*thik)^2; //volume of each
    ring
23 e=1-Vs*n;
24 Surfacearea=pi*d*l+2*pi*d^2/4+pi*(d-2*thik)*l-2*pi*(
    d-2*thik)^2/4;

```

```
25 S=Surfacearea/Vs;  
26 S=round(S*10)/10;  
27 delP=K*S^2/e^3*mu*len*u*(1-e)^2;  
28 delh=delP/rho/g;  
29 disp(delh*100,"pressure drop in terms of (cm of H20)  
    ")
```

---

# Chapter 8

## Filtration

Scilab code Exa 8.1.1 constant rate of filtration in a plate and frame filter process

```
1
2
3 //exapple 8.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 //part1
7 a=78/1000; //dV/dt
8 rho=998; //density of water
9 rhoc=2230; //density of china clay
10 rhod=1324; //density of cowdung cake
11 mu=1.003/1000;
12 P2=3.23*1000; //pressure after 2 min.
13 P5=6.53*1000; //pressure after 5 min.
14 t=30*60;
15 b=[P2;P5];
16 A=[a^2*120 a;a^2*300 a];
17 x=A\b;
18 P=x(1,1)*a^2*t+x(2,1)*a;
19 disp(P/1000,"pressure drop at t=30min in (kN/m^2):")
20 //part2
21 J=0.0278; //mass fraction
```

```

22 l=1.25;
23 b1=0.7;
24 A1=l*b1*17*2; //area of filtering
25 V=a*30*60; //volume of filtrate
26 e=1-rhod/rhoc;
27 nu=J*rho/((1-J)*(1-e)*rhoc-J*e*rho);
28 l1=nu*V/A1;
29 disp(l1,"the thickness of filtercake formed after 30
      min in (m):")
30 //part3
31 r=x(1,1)/mu/nu*A1^2;
32 L=x(2,1)*A1/r/mu;
33 disp(L,"thickness of cake required in (m):");
34 //part 4
35 S=sqrt(r*e^3/5/(1-e)^2);
36 d=6/S;
37 disp(d*10^6,"average particle diameter in(10^-6m):")

```

---

#### Scilab code Exa 8.1.2 Constant rate and pressure drop filtration

```

1
2
3 //exapple 8.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 P1=5.34*1000; //pressure after 3 min.
7 P2=9.31*1000; //pressure after 8 min.
8 a=240/1000000; //dV/dt
9 P3=15*10^3; //final pressure
10 //calculation
11 b=[P1;P2];
12 A=[a^2*180 a;a^2*480 a];
13 x=A\b;
14 //part1
15 t=(P3-x(2,1)*a)/x(1,1)/a^2;

```

```

16 disp(t,"time at which the required pressure drop
    have taken place in (s):");
17 //part 2
18 V1=a*t;
19 disp(V1,"volume of filtrate in (m^3):");
20 //part 3
21 V2=0.75;
22 t2=t+x(1,1)/2/P3*(V2^2-V1^2)+x(2,1)/P3*(V2-V1);
23 disp(t2,"the time required to collect 750dm^3 of
    filtrate in (s):");
24 //part 4
25 P4=12*10^3;
26 a=P4/(x(1,1)*V2+x(2,1));
27 t=10/1000/a;
28 disp(t,"time required to pass 10dm^3 volume in (s):"
    )

```

---

Scilab code Exa 8.1.3 determination of characteristic of filtration system

```

1
2
3 //exapple 8.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 a=16/1000;//dV/dt
7 J=0.0876;//mass fraction
8 rho=999;//density of water
9 rhoc=3470;//density of slurry
10 mu=1.12/1000;
11 rhos=1922;//density of dry filter cake
12 t1=3*60;
13 t2=8*60;
14 V1=33.8/1000;//volume at t1
15 V2=33.8/1000+23.25/1000;//volume at t2
16 P=12*1000;//pressure difference

```

```

17 Ap=70^2/10000*2*9;
18 As=650/10000;
19 // calculation
20 b=[t1;t2]
21 A=[V1^2/2/P V1/P;V2^2/2/P V2/P];
22 x=A\b;
23 K1p=x(1,1)*As^2/Ap^2;
24 K2p=x(2,1)*As/Ap;
25 P2=15*1000;//final pressure drop
26 t=(P2-K2p*a)/K1p/a^2;//time for filterate
27 V=a*t;//volume of filterate
28 e=1-rhos/rhoc;
29 nu=J*rho/((1-J)*(1-e)*rhoc-J*e*rho);
30 l=(11-1)/200;
31 Vf=Ap*l/nu;
32 tf=t+K1p/2/P2*(Vf^2-V^2)+K2p/P2*(Vf-V);
33 r=K1p/mu/nu*Ap^2;
34 L=K2p*Ap/r/mu;
35 disp(L,"the thickness of filter which has resistance
    equal to resistance of filter medium in (m):")

```

---

Scilab code Exa 8.1.4 constant pressure drop filtration of suspension which gives :

```

1
2
3 //exapple 8.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 t1=3*60;//time 3min
7 t2=12*60;//time 12min
8 t3=5*60;//time 5min
9 P=45*1000;//pressure at t1&t2
10 P2=85*1000;//pres. at t3
11 a=1.86;//area
12 mu=1.29/1000;

```

```

13 c=11.8;
14 V1=5.21/1000; //volume at t1
15 V2=17.84/1000; //volume at t2
16 V3=10.57/1000; //volume at t3
17 //calculation
18 b=[t1;t2];
19 A=[mu*c/2/a^2/P*V1^2 V1/P;mu*c/2/a^2/P*V2^2 V2/P];
20 x=A\b;
21 r45=x(1,1);
22 r85=(t3-x(2,1)*V3/P2)*2*a^2*P2/V3^2/mu/c;
23 n=log(r45/r85)/log(45/85);
24 rbar=r45/(1-n)/(45*1000)^n;
25 r78=rbar*(1-n)*(78*1000)^n;
26 //part1
27 //polynomial in V as ax^2+bx+c1=0
28 c1=90*60; //time at 90
29 Pt=78*1000; //Pt=pressure at time t=90
30 r78=round(r78/10^12)*10^12;
31 a1=r78*mu/a^2/Pt*c/2;
32 b=x(2,1)/Pt;
33 y=poly([-c1 b a1], 'V1', 'coeff');
34 V1=roots(y);
35 disp(V1(2), "Volume at P=90kPa in (m^3):");
36 //part2
37 Pt=45*1000;
38 c1=90*60;
39 a1=r45*mu/a^2/Pt*c/2;
40 b=x(2,1)/Pt;
41 y=poly([-c1 b a1], 'V1', 'coeff');
42 V1=roots(y);
43 disp(V1(2), "Volume at p=45kPa in (m^3):");

```

---

Scilab code Exa 8.1.5 filtration on a rotatory drum filter

```

2
3 //exapple 8.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 t=60*0.3/0.5;//time of 1 revollution
7 d=34/1000000;
8 S=6/d;
9 e=0.415;
10 J=0.154;
11 P=34.8*1000;
12 mu=1.17/1000;
13 L=2.35/1000;
14 rho=999;//density of water
15 rhos=4430;//density of barium carbonate
16 //calculation
17 //part1
18 nu=J*rho/((1-J)*(1-e)*rhos-J*e*rho);
19 r=5*S^2*(1-e)^2/e^3;
20 //quadratic in l
21 //in the form of ax^2+bx+c=0
22 c=-t;
23 b=r*mu*L/nu/P;
24 a=r*mu/2/nu/P;
25 y=poly([c b a], 'l', 'coeff');
26 l=roots(y);
27 disp(l(2),"thickness of filter cake in (m):");
28 //part2
29 d=1.2;
30 l1=2.6;
31 pi=3.1428;
32 u=pi*d*0.5/60;
33 Q=u*l1*l(2);
34 mnet=Q*(1-e)*rhos+Q*e*rho;
35 disp(mnet,"rate at which wet cake will be scrapped
    in (kg/s):");
36 //part3
37 md=Q*(1-e)*rhos;//rate at which solid scrapped from
    the drum

```

```

38 r=md/0.154;
39 disp(r*3600,"rate of which slurry is treated is (kg/
    h):")

```

---

### Scilab code Exa 8.1.6 filtration of centrifugal filter

```

1
2
3 //exapple 8.6
4 clc; funcprot(0);
5 // Initialization of Variable
6 mu=0.224;
7 rho=1328;
8 K=5;
9 b=3*.5;//radius
10 h=2.5;
11 pi=3.1428;
12 x=2.1*.5;
13 rhos=1581;//density of sucrose
14 e=0.435;//void ratio
15 J=0.097;//mass fraction
16 m=3500;//mass flowing
17 a=85/10^6;//side length
18 L=48/1000;//thickness
19 omega=2*pi*325/60;
20 //calculation
21 bi=b^2-m/pi/h/(1-e)/rhos;//inner radius
22 bi=sqrt(bi);
23 bi=round(bi*1000)/1000;
24 nu=J*rho/((1-J)*(1-e)*rhos-J*e*rho);
25 S=6/a;
26 r=5*S^2*(1-e)^2/e^3;
27 t=((b^2-bi^2)*(1+2*L/b)+2*bi^2*log(bi/b))/(2*nu*rho*
    omega^2/r/mu*(b^2-x^2));
28 disp(t,"time taken to collect sucrose crystal in (s)

```

```
        :");  
29 //part2  
30 vl=pi*(b^2-bi^2)*h*e;  
31 vs=pi*(b^2-bi^2)*h/nu-vl;  
32 disp(vs,"volume of liquid separated as filtrate i (  
    m^3):");
```

---

# Chapter 9

## Forces on bodies Immersed in fluids

Scilab code Exa 9.1.1 drag forces and coefficient

```
1
2
3 //exapple 9.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1.2;
7 mu=1.85/100000;
8 pi=3.1428;
9 d=3;
10 v=50*1000/3600;
11 //calculation part 1
12 Re=d*rho*v/mu;
13 //from chart of drag coeff. vs Re
14 Cd=0.2; //coeff. of drag
15 Ad=pi*d^2/4; //projected area
16 Fd=Ad*Cd*rho*v^2/2;
17 disp(Fd , "The drag force on sphere in N");
18 //part 2
19 v=2;
```

```

20 l=0.25;
21 Re=l*v*rho/mu;
22 zi=4*pi*(l^3*3/4/pi)^(2/3)/6/l^2; //sphericity
23 //using graph
24 Cd=2;
25 Ad=l^2;
26 Fd=Ad*Cd*rho*v^2/2;
27 disp(Fd , "The drag force on cube in N");

```

---

### Scilab code Exa 9.1.2 lift force and lift coefficient

```

1
2
3 //exapple 9.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1.2;
7 mu=1.85/100000;
8 pi=3.1428;
9 g=9.81;
10 d=1.38;
11 t=0.1; //thickness
12 v=30*1000/3600;
13 T=26.2; //Tension
14 m=0.51 //mass
15 theta=60*pi/180;
16 //calculation
17 Fd=T*cos(theta);
18 disp(Fd,"Drag force in N:");
19 A=pi*d^2/4;
20 Ad=A*cos(theta); //area component to drag
21 Cd=2*Fd/Ad/rho/v^2; //coeff of drag
22 disp(Cd , "The drag coefficient:")
23 Fg=m*g; //force of gravity
24 Fb=rho*pi*d^2/4*t*g; //buoyant force

```

```

25 F1=Fg-Fb+T*sin(theta);
26 disp(F1 , "The lift force in N :");
27 A1=A*sin(theta);
28 C1=2*F1/A1/rho/v^2;
29 disp(C1 , "The coefficient of lift:")

```

---

### Scilab code Exa 9.1.3 Particle diameter and terminal settling velocity

```

1
2
3 //exapple 9.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 rhog=1200;//density of glycerol
7 mu=1.45;
8 pi=3.1428;
9 g=9.81;
10 rhos=2280;//density of sphere
11 v=0.04;//terminal velocity;
12 a=2*mu*g*(rhos-rhog)/v^3/3/rhog^2;//a=Cd/2/Re
13 //using graph of Cd/2/Re vs Re
14 Re=0.32;
15 d=Re*mu/v/rhog;
16 disp(d , "Diameter of sphere in (m):");

```

---

### Scilab code Exa 9.1.4 terminal settling velocity of sphere

```

1 //exapple 9.4
2 clc; funcprot(0);
3 // Initialization of Variable
4 rhoa=1.218;//density of air
5 mu=1.73/100000;
6 pi=3.1428;

```

```

7 g=9.81;
8 rhog = 1200;
9 rhop=2280; //density of polythene
10 d=0.0034; //diameter
11 a=4*d^3*(rhop-rhoa)*rhoa*g/3/mu^2; //a=Cd*Re^2
12 //using graph of Cd*Re^2 vs Re
13 Re=2200;
14 v=Re*mu/d/rhog;
15 disp(v , "The terminal vrlocity in (m/s)");

```

---

#### Scilab code Exa 9.1.5 effect of shape on drag force

```

1
2
3 //exapple 9.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 pi=3.1428;
7 rho=825;
8 mu=1.21;
9 g=9.81;
10 l=0.02;
11 de=0.02; //dia exterior
12 di=0.012; //dia interior
13 //calculation
14 //part 1
15 zi=pi*(6*(pi*de^2/4-pi*di^2/4)*l/pi)^(2/3)/(pi*l*(di
    +de)+2*pi*(de^2/4-di^2/4));
16 disp(zi, "sphericity of Raschig ring is:");
17 //part 2
18 u=0.04;
19 ds=0.003 //diameter of each sphere
20 zi=pi*(6*pi*ds^3/pi)^(2/3)/6/pi/ds^2; //sphericity
21 disp(zi, "sphericity of given object is:");
22 Ap=4*ds^2-4*3/4*(ds^2-pi*ds^2/4); //projected area

```

```

23 dp=sqrt(4*Ap/pi); //projected dia
24 Re=dp*u*rho/mu;
25 disp(Re, "Reynolds no. for the object:");
26 //using graph b/w Re and zi and Cd
27 Cd=105; //coeff. of drag
28 Fd=Ap*Cd*u^2*rho/2;
29 disp(Fd, "The drag force on object in (N):")

```

---

### Scilab code Exa 9.1.6 estimation of hindered settling velocity

```

1
2
3 //exapple 9.6
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=998; //density of water
7 mu=1.25/1000; //viscosity of water
8 w=100; //mass of water
9 pi=3.1428;
10 g=9.81;
11 rhog=2280; //density of glass
12 wg=60; //mass of glass
13 d=45*10^-6; //diameter of glass sphere
14 //claculation
15 rhom=(w+wg)/(w/rho+wg/rhog); //density of mixure
16 e=w/rho/(w/rho+wg/rhog); //volume fraction of watter
17 //using charts
18 zi=exp(-4.19*(1-e));
19
20 K=d*(g*rho*(rhog-rho)*zi^2/mu^2)^(1/3); //stoke's law
    coeff.
21 disp(K);
22 if K<3.3 then
23     disp("settling occurs in stoke-s law range");
24     U=g*d^2*e*zi*(rhog-rhom)/18/mu;

```

```

25     disp(U,"settling velocity in m/s:")
26 else
27     disp("settling does not occurs in stoke-s law
        range");
28 end

```

---

Scilab code Exa 9.1.7 acceleration of settling particle in gravitational feild

```

1
2
3 //exapple 9.7
4 clc; funcprot(0);
5 // Initialization of Variable
6 rhog=1200;//density of glycerol
7 mu=1.45;//viscosity of glycerol
8 pi=3.1428;
9 g=9.81;
10 rhos=2280;//density of sphere
11 d=8/1000;
12 s=0;
13 uf=0.8*0.026;
14 //calculation
15 function [a]=intre()
16     u=linspace(0,uf,1000);
17     for i=1:1000
18         y=((pi/6*d^3*rhos*g-pi*d^3/6*rhog*g-0.5*pi*d
            ^2/4*24*mu/d/rhog*rhog*u(i))/pi*6/d^3/
            rhos)^(-1)*uf/1000;
19         s=s+y;
20     end
21     a=s;
22 endfunction
23 [t]=intre();
24 disp(t,"Time taken by particle to reach 80% of its
        velocity in (s):");

```



# Chapter 10

## Sedimentation and Classification

Scilab code Exa 10.1.1 determination of settling velocity from a single batch sedi

```
1
2
3 //example 10.1
4 clc; funcprot(0);
5 clf()
6 //exapple 10.1
7 // Initialization of Variable
8 t=[0 0.5 1 2 3 4 5 6 7 8 9 10]; //time
9 h=[1.10 1.03 .96 .82 .68 .54 .42 .35 .31 .28 .27
    .27];
10 C1=[0 0 0 0 0 0 0 0 0 0 0];
11 m=0.05;
12 V=1/1000; //volume
13 //calculations
14 Co=m/V; //concentration at t=0
15 v(1)=(h(1)-h(2))/(t(2)-t(1));
16 C1(1)=Co;
17 for i=2:11
18
19         v(i)=(h(i-1)-h(i+1))/(t(i+1)-t(i-1)); //slope
           or settling velocity
```

```

20         Cl(i)=Co*h(1)/(h(i)+v(i)*t(i));
21
22
23 end
24 plot(t,h,'r—d');
25 clf();
26 plot(Cl,v,'r->');
27 xtitle("Concentration vs Settling veocity" , "
        Concentration(kg/m^3)" , "Settling velocity (m/h)
        ");

```

---

Scilab code Exa 10.1.2 Minimum area required for a continuous thickener

```

1
2
3 //example 10.2
4 clc; funcprot(0);
5 clf()
6 //exapple 10.2
7 // Initialization of Variable
8 t=[0 0.5 1 2 3 4 5 6 7 8 9 10]; //time
9 h=[1.10 1.03 .96 .82 .68 .54 .42 .35 .31 .28 .27
    .27];
10 Cl=50:5:100;
11 U=[19.53 17.71 16.20 14.92 13.82 12.87 12.04 11.31
    10.65 9.55]; //mass ratio of liquid to solid
12 v=[0.139 0.115 0.098 0.083 0.071 0.062 0.055 0.049
    0.043 0.034]; //terminal velocity
13 //above value taken from graph given with ques.
14 C=130; //conc. of solids
15 Q=0.06; //slurry rate
16 Cmax=130 //maximum solid conc.
17 rhos=2300; //density of solid
18 rho=998; //density of water
19 V=rho*(1/C-1/rhos);

```

```

20 F=Q*C1(1)*3600;
21 for i=1:10
22 A(i)=F*(U(i)-V)/rho/v(i);
23 end
24 plot(v,A,'r-');
25 xtitle("", "Settling Velocity(m/h)", "Area(m^2)")
26 //maxima finding using datatraveller in the graph
27 disp(A,"the area for each settling velocity");
28 disp("1005 m^2 is the maximum area found out from
the plot");
29 Qu=Q-F/3600/Cmax;
30 disp(Qu, "Volumetric flow rate of clarified water in
(m^3/s):")

```

---

Scilab code Exa 10.1.3 classification of materials on basis of settling velocities

```

1
2
3 //example 10.3
4 clc; funcprot(0);
5 //exapple 10.3
6 // Initialization of Variable
7 rho1=2600;//density lighter
8 rho2=5100;//density heavier
9 pd1=0.000015:0.000010:0.000095;//particle diameter
lighter
10 pd2=0.000025:0.00001:0.000095;//particle diameter
heavier
11 wp1=[0 22 35 47 59 68 75 81 100];//weight
distribution lighter
12 wp2=[0 21 33.5 48 57.5 67 75 100];//weight
distribution heavier
13 rho=998.6;//density water
14 mu=1.03/1000;//viscosity water
15 g=9.81;

```

```

16 u=0.004;//velocity of water
17 d=95/1000000;//particle diameter maximum
18 //calculation
19 //part 1
20 Re=d*u*rho/mu;
21 d1=sqrt(18*mu*u/g/(rho1-rho));
22 d2=sqrt(18*mu*u/g/(rho2-rho));
23 function [a]=inter(d,f,g,b);//interpolation linear
24     for i=1:b
25         if d<=f(i+1)& d>f(i) then
26             break
27         else
28             continue
29         end
30         break
31     end
32     a=(d-f(i))/(f(i+1)-f(i))*(g(i+1)-g(i))+g(i);
33 endfunction
34 [a]=inter(d1,pd1,wp1,9);
35 [b]=inter(d2,pd2,wp2,8);
36 v2=1/(1+5)*100-b/100*1/(1+5)*100;
37 v1=5/(1+5)*100-a/100*5/(1+5)*100;
38 pl2=(v2)/(v2+v1);
39 disp(pl2, "The fraction of heavy ore remained in
    bottom");
40 //part 2
41 rho=1500;
42 mu=6.25/10000;
43 a=log10(2*d^3*rho*g*(rho1-rho)*3*mu^2);//log10(Re
    ^2(R/rho/mu^2))
44 //using value from chart(graph)
45 Re=10^0.2136;
46 u=Re*mu/rho/d;
47 d2=sqrt(18*mu*u/g/(rho1-rho));
48 [b]=inter(d2,pd2,wp2,8);
49 disp(100-b+3.5,"The percentage of heavy ore left in
    this case");
50 //part 3

```

```

51 a=0.75//% of heavy ore in overhead product
52 s=100*5/6/(100*5/6+0.75*100/6);
53 disp(s,"the fraction of light ore in overhead
    product:");
54 //part 4
55 da=pd2(1);
56 db=pd1(9);
57 rho=(da^2*rho2-db^2*rho1)/(-db^2+da^2);
58 disp(rho,"The minimum density required to separate
    2 ores in kg/m^3:")

```

---

Scilab code Exa 10.1.4 density variation of settling suspension

```

1
2
3 //example 10.4
4 clc; funcprot(0);
5 //exapple 10.4
6 // Initialization of Variable
7 rho=998;
8 w0=40;//density of slurry
9 mu=1.01/1000;
10 g=9.81;
11 rho1=2660;//density quartz
12 h=0.25;
13 t=18.5*60;
14 mp=[5 11.8 20.2 24.2 28.5 37.6 61.8];
15 d=[30.2 21.4 17.4 16.2 15.2 12.3 8.8]/1000000;
16 u=h/t;
17 d1=sqrt(18*mu*u/g/(rho1-rho));
18 function [a]=inter(d,f,g,b);//interpolation linear
19     for i=1:b
20         if d>f(i+1)& d<=f(i) then
21             break
22             else

```

```

23             continue
24             end
25             break
26         end
27
28         a=-(d-f(i+1))/(f(i)-f(i+1))*(g(i+1)-g(i))+g(i+1)
           ;
29     endfunction
30 [a]=inter(d1,d,mp,6);
31 phi=1-a/100;
32 rhot=phi*(rho1-rho)/rho1*w0+rho;
33 disp(rhot,"the density of suspension at depth 25cm
       in kg/m^3 is")

```

---

Scilab code Exa 10.1.5 determination of particle size distribution using a sedimentation method

```

1 //example 10.5
2 clc; funcprot(0);
3 clf()
4 //example 10.5
5 // Initialization of Variable
6 g = 9.81;
7 t=[0 45 135 495 1875 6900 66600 86400]; //time
8 m=[0.1911 0.1586 0.1388 0.1109 0.0805 0.0568 0.0372
    0.0359]; //mass total
9 rho1=3100; //density of cement
10 mu=1.2/1000; //viscosity of disperant liquid
11 rho=790; //density of disperant liquid
12 h=0.2;
13 V=10;
14 s=0;
15 d(1)=100/1000000; //assumed value
16 for i=1:7
17     d(i+1)=sqrt(18*mu*h/g/t(i+1)/(rho1-rho)); //dia
           of particles

```

```

18     mc(i+1)=m(i+1)-0.2/100*V;//mass of cement
19     s=s+mc(i+1);
20 end
21 mc(1)=m(1)-0.2*V/100;
22 s=s+mc(1);
23 mp(1)=100;
24 for i=1:7
25     mp(i+1)=mc(i+1)/mc(1)*100;//mass percent below
        size
26 end
27 plot(mp,d);
28 xtitle("", "%undersize", " Particle Size(m)");
29 u=h/t(2);
30 Re=d(2)*u*rho/mu;
31 if Re<2 then
32     disp("since Re<2 for 81% of particles so
        settlement occurs mainly by stoke-s law")
33 end

```

---

Scilab code Exa 10.1.6 determination of particle size distribution of a suspended

```

1
2
3 //example 10.6
4 clc; funcprot(0);
5 //exapple 10.6
6 clf()
7 // Initialization of Variable
8 rho=998;
9 rho1=2398;//density of ore
10 mu=1.01/1000;
11 g=9.81;
12 h=25/100;
13 t=[114 150 185 276 338 396 456 582 714 960];
14 m=[0.1429 0.2010 0.2500 0.3564 0.4208 0.4781 0.5354

```

```

    0.6139 0.6563 0.7277];
15 for i=1:10
16 ms=0.0573+m(10); //total mass setteled
17 d(i)=sqrt(18*mu*h/g/(rho1-rho)/t(i));
18 P(i)=m(i)/ms*100; //mass percent of sample
19 end
20 plot(t,P);
21 xtitle("","Settling time (s)","mass percent in (%)")
    ;
22 disp(P,d,"& its percentage mass distribution
    respectively" ,"the particle size distribution in
    (m)" );
23 for i=2:9
24     del(i)=(P(i+1)-P(i-1))/(t(i+1)-t(i-1)); //
        slope
25     W(i)=P(i)-t(i)*del(i);
26     W(1)=P(1)-P(1);
27
28 end
29 W(10)=P(10)-t(10)*0.025;
30 disp("mass% and diameter(m) respectively with serial
    no:")
31 for i=4:10
32     disp(i-4);
33     disp("mass% is")
34     disp(" for diameter in(m) of",W(i));
35     disp(d(i));
36
37 end

```

---

Scilab code Exa 10.1.7 decanting of homogeneous suspension to obtain particle size

```

1
2
3 //example 10.7

```

```

4  clc; funcprot(0);
5  //example 10.7
6  // Initialization of Variable
7  rho=1002;//density of disperant
8  rho1=2240;//density of kaolin
9  mu=1.01/1000;//viscosity
10 g=9.81;
11 t=600;
12 h2=0.2;
13 h1=0.4;
14 dg=15*10-6;//particle size to be removed
15 //calculations
16 //part 1
17 d=sqrt(18*mu*h2/g/(rho1-rho)/t);
18 x=dg/d;
19 f=h2/h1*(1-x2);//fraction separated after first
    decanting
20 g=f*(1-f);
21 disp(g,"fraction of particles separated after second
    decanting");
22 disp(f+g,"total fraction of particles separated
    after decanting")
23 //part 2
24 h=(1-20/40*(1-x2))6;
25 disp(h,"fraction of particles separated after sixth
    decanting");

```

---

# Chapter 11

## Fluidisation

Scilab code Exa 11.1.1 particulate and aggregative fluidisation

```
1
2
3 //exapple 11.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 pi=3.1428;
7 d=0.3/1000;
8 mu=2.21/100000;
9 rho=106.2; //density under operating condition
10 u=2.1/100;
11 rhos=2600; //density of particles
12 l=3.25;
13 g=9.81;
14 dt=0.95 //fluidising diameter
15 //part 1
16 //calculation
17 a=u^2/d/g*d*rho*u/mu*(rhos-rho)/rho*l/dt;
18 if a>100 then
19     disp(a,"Bubbling fluidisation will occur as
20         value is ")
21 end
```

```

21 //part 2
22 Q=2.04/100000;
23 rhos=2510;
24 rho=800;
25 mu=2.85/1000;
26 l=4.01;
27 dt=0.63;
28 d=0.1/1000;
29 u=Q*4/pi/dt^2;
30 a=u^2/d/g*d*rho*u/mu*(rhos-rho)/rho*l/dt;
31 if a<100*10^-4 then//compare as value of a is much
    less than 100
32     disp(a,"fluidisation occur in smooth mode as
        value is:");
33 end

```

---

Scilab code Exa 11.1.2 calculation of minimum flow rates

```

1
2
3 //exapple 11.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 d=50/1000000;
7 rhos=1850;//density of particle
8 rho=880;//density of hydrocarbon
9 mu=2.75/1000;//viscosity of hydrocarbon
10 e=0.45;//void fraction coeff.
11 g=9.81;
12 h=1.37;//flow depth
13 c=5.5/1000;//c=1/K
14 //calculation
15 //part 1
16 u=c*e^3*d^2*g*(rhos-rho)/mu/(1-e);
17 disp(u,"The superficial linear flow rate in (m/s):")

```

```

18 //part 2
19 u=d^2*g*(rhos-rho)/18/mu;
20 disp(u,"Terminal Settling Velocity in (m/s):");
21 Re=d*u*rho/mu;
22 if Re<2 then
23     disp("Stoke law assumption is sustained with
           this velocity")
24 end
25 //part 3
26 P=g*(rhos-rho)*h*(1-e);
27 disp(P,"Pressure drop across fluidised bed in (N/m
       ^2):");

```

---

#### Scilab code Exa 11.1.3 calculation of flow rates in fluidised beds

```

1
2
3
4 //exapple 11.3
5 clc; funcprot(0);
6 // Initialization of Variable
7 g=9.81;
8 rhos=1980;//density of ore
9 rho=1.218;//density of air
10 e=0.4;
11 mu=1.73/10^5;
12 s=0;
13 wp=[0 .08 .20 .40 .60 .80 .90 1.00]);//weight percent
14 d=[0.4 0.5 0.56 0.62 0.68 0.76 0.84 0.94]/1000;
15 //part 1
16 for i=1:7
17     dav(i)=d(i+1)/2+d(i)/2;//average dia
18     mf(i)=wp(i+1)-wp(i);//mass fraction
19     a(i)=mf(i)/dav(i);
20     s=s+a(i);

```

```

21 end
22 db=1/s; //d bar
23 //quadratic coeff. ax^2 +bx +c=0
24 c=-(rhos-rho)*g;
25 b=150*(1-e)/e^3/db^2*mu;
26 a=1.75*rho/e^3/db;
27 y=poly([c b a], 'U', 'coeff');
28 U=roots(y);
29 disp(abs(U(2)), "the linear air flow rate in (m/s):"
    );
30 //part 2
31 d=0.4/1000;
32 a=2*d^3/3/mu^2*rho*(rhos-rho)*g;
33 a=log10(a);
34 disp(a, "log10(Re^2/rho/U^2*R)=");
35 //using chart
36 Re=10^1.853;
37 u=Re*mu/rho/d;
38 disp(u, "speed required for smallest particle in (m/
    s):")

```

---

Scilab code Exa 11.1.4 estimation of vessel diameters and height for fluidisation

```

1
2
3 //exapple 11.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 U=2.032/10^4;
7 pi=3.1428;
8 rho=852;
9 g=9.81;
10 mu=1.92/1000;
11 mf=125/3600; //mass flow rate
12 //calculation

```

```

13 //part 1
14 G=U*rho;
15 A=mf/G;
16 d=sqrt(4*A/pi);
17 disp(d, "the diameter of vessel will be in(m):");
18 //part 2
19 A=0.201;
20 e=0.43;
21 ms=102;//mass of solids
22 rhos=1500;//density of solid
23 L=ms/rhos/A;
24 Lmf=L/(1-e);
25 disp(Lmf, "depth of bed in (m):")
26 //part 3
27 d1=0.2/1000;
28 U=2*5.5/10^3*e^3*d1^2*(rhos-rho)*g/mu/(1-e);
29 //now euating for e
30 //a=e^3/(1-e)
31 a=U/5.5*10^3/(d1^2*(rhos-rho)*g/mu);
32 y=poly([-a a 0 1], 'e', "coeff");
33 e2=roots(y);
34 L=Lmf*(1-e)/(1-e2(3));
35 disp(L, "depth of fluidised bed under operating
    condition in (m):")

```

---

#### Scilab code Exa 11.1.5 power required for pumping in fluidised beds

```

1
2
3 //exapple 11.5
4 clc; funcprot(0);
5 // Initialization of Variable
6 g=9.81;
7 pi=3.1428;
8 r=0.51;

```

```

 9 e=0.48; //void ratio
10 rhos=2280; //density of glass
11 rho=1.204; //density of air
12 U=0.015; //velocity of water entering bed
13 L=7.32;
14 gam=1.4; //gamma
15 neta=0.7 //efficiency
16 P4=1.013*10^5;
17 P1=P4;
18 v1=1/1.204; //volume 1
19 //calculation
20 P3=P4+g*(rhos-rho)*(1-e)*L;
21 P2=P3+0.1*85090;
22 v2=(P1*v1^gam/P2)^(1/gam); //vlume 2
23 W=1/neta*gam/(gam-1)*(P2*v2-P1*v1); //work done
24 v3=P2*v2/P3; //volume 3
25 M=U*pi*r^2/v3; //mass flow rate
26 P=M*W;
27 disp(P,"The power supplies to the blower in (W):");

```

---

### Scilab code Exa 11.1.6 wall effect in fluidised beds

```

1
2
3 //exapple 11.6
4 clc; funcprot(0);
5 // Initialization of Variable
6 dt=12.7/1000;
7 d=1.8/1000;
8 Q=2.306/10^6;
9 pi=3.1428;
10 //calculation
11 //part 1
12 Sc=4/dt;
13 S=6/d;

```

```

14 f=(1+0.5*Sc/S)^2;
15 U=Q*4/pi/dt^2; // velocity
16 Ua=f*U; // actual velocity
17 disp(Ua,"minimum fluidising velocity found using
    smaller glass column in (m/s):")
18 //part 2
19 dt=1.5;
20 Sc=4/dt;
21 f=(1+0.5*Sc/S)^2;
22 Ua=f*U; // actual velocity
23 disp(Ua,"fluidising velocity found using larger
    glass column in (m/s):")

```

---

Scilab code Exa 11.1.7 effect of particle size on the ratio of terminal velocity

```

1
2
3 //exapple 11.7
4 clc; funcprot(0);
5 // Initialization of Variable
6 e=0.4; //incipent to fluidisation
7 //calculation
8 //part 1
9 disp("for Re<500");
10 disp("the ratio of terminal velocity & minimum
    fluidising velocity is");
11 a=3.1*1.75/e^3;
12 disp(sqrt(a));
13 //part 2
14 disp("for Re>500");
15 disp("the ratio of terminal velocity & minimum
    fluidising velocity is");
16 a=150*(1-e)/18/e^3;
17 disp(a);

```

---

# Chapter 12

## Pneumatic Conveying

Scilab code Exa 12.1.1 flow pattern in pneumatic conveying

```
1
2
3 //example 12.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1.22;
7 pi=3.1428;
8 rhos=518;
9 rhoav=321;
10 mu=1.73/10^5;
11 g=9.81;
12 d=0.65/1000;
13 d2=25.5/100;//dia of duct
14 ms=22.7/60;//mass flow rate
15 //calculation
16 e=(rhos-rhoav)/(rhos-rho);
17 //coeff of quadratic eqn in U
18 //a*x^2+b*x+c=0
19 c=-(1-e)*(rhos-rho)*g;
20 b=150*(1-e)^2*mu/d^2/e^3;
21 a=1.75*(1-e)*rho/d/e^3;
```

```

22 y=poly([c b a], 'U', 'coeff');
23 U=roots(y);
24 Us=ms*4/pi/d2^2/rhos; //superficial speed
25 Ua=e/e*(U(2)/e+Us/(1-e));
26 disp(Ua,"the actual linear flow rate through duct in
      (m/s):")

```

---

Scilab code Exa 12.1.2 prediction of choking velocity and choking voidage

```

1
2
3 //example 12.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1.22; //density of air
7 pi=3.1428;
8 rhos=910; //density of polyethene
9 d=3.4/1000; //dia of particles
10 mu=1.73/10^5;
11 g=9.81;
12 dt=3.54/100; //dia of duct
13 //calculation
14 a=2*d^3*rho*g*(rhos-rho)/3/mu^2;
15 disp(a,"R/rho/U^2*(Re^2)=");
16 //using Chart
17 Re=2*10^3;
18 U=mu*Re/d/rho;
19 b=U/(g*dt)^.5;
20 if b>0.35 then
21     disp("choking can occur of this pipe system");
22 else
23     disp("choking can not occur of this pipe system"
24         );
25 end
26 //part 2

```

```

26 Uc=15; //actual gas velocity
27 e=((Uc-U)^2/2/g/dt/100+1)^(1/-4.7);
28 Usc=(Uc-U)*(1-e); //superficial speed of solid
29 Cmax=Usc*rhos*pi*dt^2/4;
30 disp(Cmax,"the maximum carrying capacity of
    polythene particles in (kg/s)");

```

---

Scilab code Exa 12.1.3 prediction of pressure drop in horizontal pneumatic transport

```

1
2
3 //example 12.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1.22; //density of air
7 pi=3.1428;
8 rhos=1400; //density of coal
9 mu=1.73/10^5;
10 g=9.81;
11 U=25;
12 Ut=2.80;
13 l=50;
14 ms=1.2; //mass flow rate
15 mg=ms/10; //mass flow of gas
16 //calculation
17 Qs=ms/rhos; //flow of solid
18 Qg=mg/rho; //flow of gas
19 us=U-Ut; //actual linear velocity
20 A=Qg/U;
21 Us=Qs/A; //solid velocity
22 e=(us-Us)/us;
23 d=sqrt(4*A/pi);
24 function [y ]= fround(x,n)
25 // fround(x,n)
26 // Round the floating point numbers x to n decimal

```

```

    places
27 // x may be a vector or matrix// n is the integer
    number of places to round to
28 y=round(x*10^n)/10^n;
29 endfunction
30 [d]=fround(d,4);
31 Re=d*rho*U/mu;
32 //using moody's chart
33 phi=2.1/1000;//friction factor
34 P1=2*phi*U^2*1*rho/d*2;
35 f=0.05/us;
36 P2=2*1*f*(0.0098)*rhos*us^2/d;
37 P2=fround(P2/1000,1)*1000
38 delP=rho*e*U^2+rhos*(0.0098)*us^2+P1+P2;
39 //disp(delP,"the pressure difference in kN/m^2 ");
40 printf('The Pressure value in (kN/m^2) is %.1f',delP
    /1000);

```

---

Scilab code Exa 12.1.4 prediction of pressure drop in vertical pneumatic transport

```

1
2
3 //example 12.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=1.22;//density of air
7 pi=3.1428;
8 rhos=1090;//density of steel
9 mu=1.73/10^5;
10 g=9.81;
11 d=14.5/100;
12 Qg=0.4;
13 Qs=5000/3600/1090;
14 Ut=6.5;
15 ar=0.046/1000;//absolute roughness

```

```

16 l=18.5; //length
17 //calculation
18 function [y ]= fround(x,n)
19 // fround(x,n)
20 // Round the floating point numbers x to n decimal
    places
21 // x may be a vector or matrix// n is the integer
    number of places to round to
22 y=round(x*10^n)/10^n;
23 endfunction
24 Us=Qs/pi/d^2*4; //solid velocity
25 U=Qg/pi/d^2*4;
26 us=U-Ut; //actual linear velocity
27 e=1-Us/us;
28 e=fround(e,4);
29 Re=rho*U*d/mu;
30 rr=ar/d; //relative roughness
31 //using moody's diagram
32 phi=2.08/1000;
33 P1=2*phi*U^2*l*rho/d*2;
34 f=0.05/us;
35 P2=2*l*f*(1-e)*rhos*us^2/d;
36 P2=fround(P2/1000,2)*1000;
37 delP=rhos*(1-e)*us^2+rhos*(1-e)*g*l+P1+P2;
38 //disp(delP,"the pressure difference in kN/m^2 ");
39 printf('The Pressure value in (kN/m^2) is %.2f',delP
    /1000)

```

---

Scilab code Exa 12.1.5 density phase flow regime for pneumatic transport

```

1 //example 12.5
2 clc; funcprot(0);
3 // Initialization of Variables
4 l=25;
5 g = 9.81;

```

```
6 pi=%pi;
7 rhos=2690;//density of ore
8 emin=0.6;
9 emax=0.8;
10 //calculation
11 Pmax=rhos*(1-emin)*g*l;
12 disp(Pmax,"The maximum pressure drop in (N/m^2):");
13 Pmin=rhos*(1-emax)*g*l;
14 disp(Pmin,"The minimum pressure drop in (N/m^2):");
```

---

# Chapter 13

## Centrifugal Separation Operations

Scilab code Exa 13.1.1 Equations of centrifugal operations

```
1
2
3 //exapple 13.1
4 clc; funcprot(0);
5 // Initialization of Variable
6 rho=998;
7 g=9.81;
8 pi=3.1428;
9 omega=2*pi*1055/60; //angular rotation
10 r=2.55/100 //radius outer
11 ld=1.55/100; //liq. depth
12 l=10.25/100;
13 // calculation
14 //part1
15 a=r*omega^2/g;
16 disp(a,"ratio of cetrifugal force & gravitational
    force is:");
17 //part2
18 ri=r-ld; //radius internal
```

```

19 V=pi*(r^2-ri^2)*l;
20 sigma=(omega^2*V)/(g*log(r/ri));
21 disp(sigma,"equivalent to gravity settling tank of
    crossectional area of in (m^2):")

```

---

### Scilab code Exa 13.1.2 fluid pressure in tubular bowl centrifuge

```

1
2
3 //exapple 13.2
4 clc; funcprot(0);
5 // Initialization of Variable
6 sigma=55*10^6;//maximum stress
7 d=35.2/100;
8 rhos=8890;//density of bronze
9 rho=1105;//density of solution
10 t=80/1000;//thickness
11 tau=4.325/1000;
12 pi=3.1428;
13 //calculation
14 //part1
15 ri=d/2-t;//radius internal
16 function [y ]= fround(x,n)
17 // fround(x,n)
18 // Round the floating point numbers x to n decimal
    places
19 // x may be a vector or matrix// n is the integer
    number of places to round to
20 y=round(x*10^n)/10^n;
21 endfunction
22 omega=sqrt((sigma*tau*2/d)/(.5*rho*(d^2/4-ri^2)+rhos
    *tau*d/2));
23 N=60*omega/2/pi;
24 disp(N,"The maximum safe speed allowed in rpm:");
25 //part2

```

```

26 P=.5*rho*(d^2/4-ri^2)*omega^2;
27 P=fround(P/10^4,1)*10^4;
28 //disp(P," the power in N/m^2:");
29 printf('the power in N/m^2: %3.2e\n', P);
30 a=rho*omega^2*d/2;
31 a=fround(a/10^6,1)*10^6;
32 //disp(a," pressure gradient in radial direction in N
    /m^3:")
33 printf('pressure gradient in radial direction in N/m
    ^3: %3.2e\n', a);

```

---

### Scilab code Exa 13.1.3 particle size determination of fine particles

```

1
2
3 //exapple 13.3
4 clc; funcprot(0);
5 // Initialization of Variable
6 rhos=1425;//density of organic pigment
7 rho=998;//density of water
8 pi=3.1428;
9 omega=360*2*pi/60;
10 mu=1.25/1000;
11 t=360;
12 r=0.165+0.01;
13 ro=0.165;
14 //calculation
15 d=sqrt(18*mu*log(r/ro)/t/(rhos-rho)/omega^2);
16 printf('the minimum diameter in organic pigment in m
    : %3.1e\n', d);

```

---

### Scilab code Exa 13.1.4 flow rates in continuous centrifugal sedimentation

```

1
2
3 //exapple 13.4
4 clc; funcprot(0);
5 // Initialization of Variable
6 rhos=1455;//density of crystals
7 rho=998;//density of wliquid
8 g=9.81;
9 pi=3.1428;
10 mu=1.013/1000;
11 omega=2*pi*60000/60;
12 l=0.5;
13 d=2*10^-6;//dia of particles
14 r=50.5/1000;//radius
15 t=38.5/1000;//thickness of liquid
16 //calculation
17 ri=r-t;
18 V=pi*l*(r^2-ri^2);
19 Q=d^2*(rhos-rho)/18/mu*omega^2*V/log(r/ri);
20 disp(Q,"the maximum volumetric flow rate in (m^3/s):
    ")

```

---

Scilab code Exa 13.1.5 separation of two immiscible liquid by centrifugation

```

1
2
3 //exapple 13.5
4 clc; funcprot(0);
5 // Initialization of Variable
6 rhoc=867;//density of cream
7 rhom=1034;//density of skimmem milk
8 rm=78.2/1000;//radius of skimmed milk
9 rc=65.5/1000;//radius of cream
10 //calculation
11 r=sqrt((rhom*rm^2-rhoc*rc^2)/(rhom-rhoc));

```

```
12 disp(r,"distance of xis of rotation of cream milk  
    interface in (m):")
```

---

### Scilab code Exa 13.1.6 Cyclone Separators

```
1  
2  
3 //exapple 13.6  
4 clc; funcprot(0);  
5 // Initialization of Variable  
6 rho=1.210;//density of air  
7 mu=1.78/10^5;  
8 g=9.81;  
9 rhos=2655;//density of ore  
10 pi=3.1428;  
11 d=0.095;  
12 dp=2*10^-6//particle diameter  
13 dt=0.333;//dia of cyclone separator  
14 h=1.28;  
15 //calculation  
16 U=dp^2*g*(rhos-rho)/18/mu;  
17 Q=0.2*(pi*d^2/4)^2*d*g/U/pi/h/dt;  
18 disp(Q,"volumetric flow rate in(m^3/s):")
```

---

### Scilab code Exa 13.1.7 efficiency of cyclone separators

```
1  
2  
3 //exapple 13.6  
4 clc; funcprot(0);  
5 // Initialization of Variable  
6 b=4.46*10^4;  
7 c=1.98*10^4;
```

```
8 s=0;
9 function [a]=integrate()
10     s=0;
11     for i=1:10889
12         d=linspace(0,10000,10889);
13         y=(1-exp(-b*d(i))*c*(1-exp(-c*d(i))))*0.69;;
14         s=s+y;
15
16     end
17     a=y;
18 endfunction
19 a=integrate();
20 disp(a*100,"overall efficiency of cyclone separator
    in %");
```

---