

Scilab Textbook Companion for  
Principles Of Fluid Mechanics  
by M. K. Natarajan <sup>1</sup>

Created by  
Sarvan Dharavath  
Chemical Engineering  
Others  
IIT Bombay  
College Teacher  
NA

Cross-Checked by  
Mukul R. Kulkarni

December 10, 2013

<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:** Principles Of Fluid Mechanics

**Author:** M. K. Natarajan

**Publisher:** Oxford and IBH Publishing Co.

**Edition:** 3

**Year:** 1984

**ISBN:** 8120408888

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 Basic Concepts	8
2 Fluid Statics	12
3 Conservation Principle of Mass	22
4 Conservation Principle of Momentum	24
5 Conservation Principle of Energy	29
6 Dimensional Analysis and Similitude	44
7 In compressible Flow through Conduits	47
8 Uniform Open Channel Flow	61
9 Potential Flow	68
10 The Boundary Layer	70
11 Forces on Immersed Bodies	73

# List of Scilab Codes

Exa 1.1	Example 1	8
Exa 1.2	Example 2	8
Exa 1.3	Example 3	9
Exa 1.4	Example 4	10
Exa 1.6	Example 5	10
Exa 1.7	Example 6	10
Exa 2.1	Example 1	12
Exa 2.2	Example 2	12
Exa 2.3	Example 3	13
Exa 2.4	Example 4	14
Exa 2.5	Example 5	14
Exa 2.6	Example 6	14
Exa 2.7	Example 7	15
Exa 2.8	Example 8	15
Exa 2.9	Example 9	16
Exa 2.10	Example 10	16
Exa 2.11	Example 11	17
Exa 2.12	Example 12	17
Exa 2.13	Example 13	18
Exa 2.14	Example 14	18
Exa 2.15	Example 15	19
Exa 2.16	Example 16	19
Exa 2.17	Example 17	20
Exa 2.18	Example 18	20
Exa 2.19	Example 19	21
Exa 3.3	Example 1	22
Exa 3.4	Example 2	22
Exa 4.1	Example 1	24

Exa 4.3	Example 2	24
Exa 4.4	Example 3	25
Exa 4.5	Example 4	26
Exa 4.6	Example 5	26
Exa 4.7	Example 6	27
Exa 4.8	Example 7	28
Exa 5.1	Example 1	29
Exa 5.2	Example 2	29
Exa 5.3	Example 3	30
Exa 5.4	Example 4	31
Exa 5.6	Example 5	31
Exa 5.7	Example 6	32
Exa 5.8	Example 7	32
Exa 5.9	Example 8	33
Exa 5.10	Example 9	33
Exa 5.11	Example 10	34
Exa 5.12	Example 11	34
Exa 5.13	Example 12	35
Exa 5.15	Example 15	36
Exa 5.16	Example 16	36
Exa 5.17	Example 17	37
Exa 5.18	Example 18	37
Exa 5.19	Example 19	38
Exa 5.20	Example 20	38
Exa 5.21	Example 21	39
Exa 5.22	Example 22	39
Exa 5.23	Example 23	40
Exa 5.24	Example 24	40
Exa 5.25	Example 25	41
Exa 5.26	Example 26	41
Exa 5.27	Example 27	42
Exa 5.28	Example 28	42
Exa 6.11	Example 11	44
Exa 6.12	Example 12	45
Exa 6.14	Example 14	45
Exa 6.16	Example 16	46
Exa 6.17	Example 17	46
Exa 7.1	Example 1	47

Exa 7.2	Example 2	48
Exa 7.3	Example 3	48
Exa 7.4	Example 4	49
Exa 7.5	Example 5	50
Exa 7.6	Example 6	50
Exa 7.7	Example 7	51
Exa 7.8	Example 8	51
Exa 7.9	Example 9	52
Exa 7.10	Example 10	52
Exa 7.11	Example 11	53
Exa 7.12	Example 12	54
Exa 7.13	Example 13	54
Exa 7.14	Example 14	55
Exa 7.15	Example 15	55
Exa 7.16	Example 16	56
Exa 7.17	Example 17	57
Exa 7.18	Example 18	57
Exa 7.19	Example 19	57
Exa 7.20	Example 20	58
Exa 7.21	Example 21	59
Exa 7.22	Example 22	60
Exa 8.1	Example 1	61
Exa 8.2	Example 2	61
Exa 8.3	Example 3	62
Exa 8.4	Example 4	63
Exa 8.5	Example 5	63
Exa 8.6	Example 6	64
Exa 8.7	Example 7	64
Exa 8.8	Example 8	65
Exa 8.9	Example 9	65
Exa 8.10	Example 10	66
Exa 8.11	Example 11	67
Exa 9.4	Example 4	68
Exa 9.14	Example 14	68
Exa 9.15	Example 15	69
Exa 10.1	Example 1	70
Exa 10.2	Example 2	70
Exa 10.5	Example 5	71

Exa 10.6	Example 6	.....	72
Exa 11.1	Example 1	.....	73
Exa 11.2	Example 2	.....	73
Exa 11.3	Example 3	.....	74
Exa 11.4	Example 4	.....	75
Exa 11.5	Example 5	.....	75
Exa 11.6	Example 6	.....	75



# Chapter 1

## Basic Concepts

Scilab code Exa 1.1 Example 1

```
1 clc
2 // Initialization of variables
3 weight = 9800 //Kg
4 g=9.81 //m/s^2
5 a=2 //m/s^2
6 // calculations
7 m=weight/g
8 Wm=m*a
9 // results
10 printf("Density on earth =%.2f Kg/m^3",m)
11 printf("\\n Weight on moon = %.2f N",Wm)
12 printf("\\n Density on moon remains unchanged and is
    equal to %.2f Kg/m^3",m)
```

---

Scilab code Exa 1.2 Example 2

```
1 clc
2 // Initialization of variables
```

```

3 w=150 //N
4 theta=30 //degrees
5 l=0.8 //m
6 b=0.8 //m
7 dy=0.12 //cm
8 v=20 //cm/s
9 //calculations
10 Tau=w*sind(theta) /(l*b)
11 rd=v/dy
12 vis=Tau/rd
13 //results
14 printf("Viscosity of the fluid = %.2f N s/m^2",vis)

```

---

### Scilab code Exa 1.3 Example 3

```

1 clc
2 clear
3 //Initialization of variables
4 vis=2.5/10 //N s/m^2
5 D=15 //cm
6 N=180
7 dy=0.0001 //m
8 l=0.15 //m
9 b=0.25 //m
10 r=0.152 //m
11 //calculations
12 dv=%pi *D*N/60/100
13 Tau=vis*dv/dy
14 Tor=Tau*%pi*l*b*r/2
15 P=Tor*2*%pi*N/60
16 //results
17 printf("Power required = %d W",P)
18 disp("The answer is a bit different due to rounding
      off error in textbook.")

```

---

#### Scilab code Exa 1.4 Example 4

```
1 clc
2 // Initialization of variables
3 w=1 //rad/s
4 T=0.4 //N/m^2
5 // calculations
6 mu=T/tan(w)
7 // results
8 printf("Viscosity = %.2f N s/m^2",mu)
```

---

#### Scilab code Exa 1.6 Example 5

```
1 clc
2 // Initialization of variables
3 d=0.05*10^-3 //m
4 T=72*10^-3 //N/m
5 P=101 //kN/m^2
6 // calculations
7 Pi=P*1000 + 2*T/(d/2)
8 // results
9 printf("Pressure = %.2f kN/m^2",Pi/1000)
```

---

#### Scilab code Exa 1.7 Example 6

```
1 clc
2 // Initialization of variables
3 gam=981 //dyn/cm^2
4 sigma=72 //dyn/cm
```

```
5 theta=0 //degrees
6 d=0.5 //cm
7 depth=90 //cm
8 //calculations
9 h=4*sigma*cosd(theta) /(gam*d)
10 Td=depth-h
11 //results
12 printf("True depth = %.3f cm",Td)
```

---

# Chapter 2

## Fluid Statics

Scilab code Exa 2.1 Example 1

```
1 clc
2 // Initialization of variables
3 h1=1.5 //m
4 h2=2 //m
5 g1=800 //kg/m^3
6 g2=1000 //kg/m^3
7 g=9.81
8 // calculations
9 P=h1*g*g1 + h2*g*g2
10 // results
11 printf(" Pressure at the bottom of the vessel = %.2f
    kN/m^2" ,P/1000)
```

---

Scilab code Exa 2.2 Example 2

```
1 clc
2 // Initialization of variables
3 depth=8000 //m
```

```

4 sw=10.06 //kN/m^3
5 BM=2.05*10^9 //N/m^2
6 //calculations
7 g=sw*10^3 /(1- sw*10^3 *depth/BM)
8 Ph=2.3*BM*log10(BM/(BM-depth*9.81*1025))
9 //results
10 printf(" Specific weight = %.2 f kN/m^2",g/1000)
11 printf("\n Pressure at depth h = %.2 f MN/m^2",Ph
    /10^6)

```

---

### Scilab code Exa 2.3 Example 3

```

1 clc
2 //Initialization of variables
3 Patm=101.3/9.81 //m of water
4 x1=0.45 //m
5 x2=0.3 //m
6 s1=920 //Kg/m^3
7 s2=13600 //Kg/m^3
8 g=9.81 //m/s^2
9 //calculations
10 Pa=s1*x1*g + s2*x2*g
11 Pa2=Pa/(1000*g)
12 Pa3=Pa/(s2)
13 //results
14 printf(" Pressure at A = %.1 f kPa",Pa/1000)
15 printf("\n Pressure at A = %.3 f m of water",Pa2)
16 printf("\n Pressure at A = %.3 f m of mercury",Pa3)
17 printf("\n Pressure at A = %.3 f m of water absolute"
    ,Pa/1000 +101.3)
18 printf("\n Pressure at A = %.3 f m of mercury",Pa2
    +10.3)

```

---

#### Scilab code Exa 2.4 Example 4

```
1 clc
2 // Initialization of variables
3 sg=1.25
4 d=0.5 //m
5 d2=13.5*10^-2 //m
6 sw=9.81 //kN/m^2
7 // calculations
8 s1=sg*sw
9 sm=13.6*sw
10 Pa=s1*d - sm*d2
11 // results
12 printf(" Pressure at A = %.2f kN/m^2 vacuum ",Pa)
```

---

#### Scilab code Exa 2.5 Example 5

```
1 clc
2 // Initialization of variables
3 s1=0.85
4 s2=13.6
5 z1=30
6 z2=15
7 z3=20
8 z4=35
9 z5=60
10 // calculations
11 dHa=s1*(z1+z5+z3-z4) +s2*z4 -z3+s2*z2-s1*(z1+z2)
12 Pd=1000*9.81*dHa/100
13 // results
14 printf(" Pressure difference = %.2f kN/m^2",Pd/1000)
```

---

#### Scilab code Exa 2.6 Example 6

```

1  clc
2  //Initialization of variables
3  P=450 //kN/m2
4  alt=2000 //m
5  r=610 //mm of mercury
6  //calculations
7  Pat=760-r
8  Pat2=Pat*13.6*9.81*10-3
9  Pg=Pat2+P
10 //results
11 printf("Gauge reading = %.2f kN/m2",Pg)

```

---

#### Scilab code Exa 2.7 Example 7

```

1  clc
2  //Initialization of variables
3  g=9.81 //kN/m2
4  hc=16.25 //m
5  l=1.5 //m
6  b=2.5 //m
7  f=0.3
8  Pi=50 //kN
9  //calculations
10 P=g*hc*l*b
11 Preq=Pi+f*P
12 //results
13 printf("Force required to lift the gate = %.2f kN",
        Preq)

```

---

#### Scilab code Exa 2.8 Example 8

```

1  clc
2  //Initialization of variables

```



```

3 a=6 //m
4 b=8 //m
5 //calculations
6 Ixy=9/32 *b^4 /4
7 xp= Ixy/(2/3 *b *1/2 *a*b)
8 ICG=1/36 *a*b^3
9 yp=2/3*b + ICG/(2/3 *b* 1/2 *a*b )
10 //results
11 printf("The coordinates of centre of pressure are (%
    .2f ,%d)",xp,yp)

```

---

#### Scilab code Exa 2.9 Example 9

```

1 clc
2 //Initialization of variables
3 z=1.2 //m
4 y=1 //m
5 //calculations
6 hp=0.6 + 1/12 *y*z^3 /(0.6*y*z)
7 //results
8 printf("Position of hinge = %.1f m",hp)

```

---

#### Scilab code Exa 2.10 Example 10

```

1 clc
2 //Initialization of variables
3 r=0.75 //m
4 gam=8 //kN/m^3
5 //calculations
6 hp=3*%pi*r/16
7 P=gam*2/3 *r^3
8 //results
9 printf("Total pressure location = %.3f m",hp)

```

```
10 printf("\n Total pressure = %.2f kN",P)
```

---

### Scilab code Exa 2.11 Example 11

```
1 clc
2 // Initialization of variables
3 l=3 //m
4 b=2 //m
5 h1=0.75 //m
6 h2=1 //m
7 sg=0.9
8 // calculations
9 IP=sg*9.81*h2
10 F1=0.5*IP*h2
11 F2=IP*h1
12 F3=0.5*(9.81*h1)*h1
13 F=l*(F1+F2+F3)
14 ybar= (F1*(h1+ 1/3) + F2* h1/2 + F3* h1/3)/(F1+F2+F3
        )
15 // results
16 printf(" Total force = %.2f kN",F)
17 printf("\n Location = %.3f m from the base",ybar)
```

---

### Scilab code Exa 2.12 Example 12

```
1 clc
2 // Initialization of variables
3 g=1000*9.81 //kg/m^3
4 hc=20 //m
5 Ax=40*1 //m^2
6 y1=0 //m
7 y2=40 //m
8 // calculations
```

```

9 Fx=g*hc*Ax
10 function [f] =fy(y)
11     f=(12*y)^(1/3)
12 endfunction
13 Fy=intg(y1,y2,fy)
14 Fy=g*Fy(1)
15 F=sqrt(Fx^2 +Fy^2)
16 //results
17 printf("Net force = %d kN",F/1000)
18 //The answer is a bit different due to rounding off
    error in the textbook

```

---

#### Scilab code Exa 2.13 Example 13

```

1 clc
2 // Initialization of variables
3 g=9.81 //kN/m^2
4 hc=1 //m
5 l=3 //m
6 b=0.5 //m
7 // calculations
8 Ax=l*b //m^2
9 Fx=g*hc*Ax
10 Fz=g*(0.5* %pi/4 *b^2)*l
11 F=sqrt(Fx^2 + Fz^2)
12 theta=atand(Fz/Fx)
13 //results
14 printf("Magintude of resultant force = %.2f kN",F)
15 printf("\n Directiononn of the resultant force = %.1f
    deg",theta)

```

---

#### Scilab code Exa 2.14 Example 14

```

1  clc
2  // Initialization of variables
3  r1=920 //kg/m^3
4  r2=1030 //kg/m^3
5  // calculations
6  VtbyV2=r2/r1
7  V1byV2=VtbyV2-1
8  V1byVt=1/(1+1/V1byV2)
9  // results
10 printf(" fraction = %.3f ",V1byVt)

```

---

**Scilab code Exa 2.15** Example 15

```

1  clc
2  // Initialization of variables
3  d=3 //m
4  rh1=1.19 //kg/m^3
5  rh2=0.17 //kg/m^3
6  g=9.81 //m/s^2
7  // calculations
8  pay=(rh1-rh2)*g*pi/6 *d^3
9  // results
10 printf(" Pay load = %.2f N",pay)

```

---

**Scilab code Exa 2.16** Example 16

```

1  clc
2  // Initialization of variables
3  x=poly(0,"x")
4  // calculations
5  y=6*x^2 -6*x+1
6  z=roots(y)
7  // results

```

```
8 printf("For stability , s must be greater than %.2f
    and less than %.2f and must be less than 1",z(1),
    z(2))
```

---

#### Scilab code Exa 2.17 Example 17

```
1 clc
2 // Initialization of variables
3 ax=1.5 //m/s^2
4 g=9.81 //m/s^2
5 // calculations
6 alpha=atand(ax/g)
7 // results
8 printf("The interface is inclined at %.2f degrees
    with the horizontal",alpha)
```

---

#### Scilab code Exa 2.18 Example 18

```
1 clc
2 // Initialization of variables
3 d=10 //cm
4 h=25 //cm
5 hw=15 //cm
6 g=9.81 //m/s^2
7 // calculations
8 z=d^2 *d*2/d^2
9 w=sqrt(z*2*g/(d/2)^2 *100)
10 N=w/(2*%pi) *60
11 // results
12 printf("Speed of rotation = %d rpm",N)
```

---

Scilab code Exa 2.19 Example 19

```
1  clc
2  // Initialization of variables
3  dia=1 //m
4  h=3 //m
5  rho=1000 //kg/m^3
6  N=80 //rpm
7  g=9.81 //m/s^2
8  // calculation
9  w=2*%pi*N/60
10 function y = fun(r)
11     y=0.5*rho*w^2 *r^3 *2*%pi
12 endfunction
13 vec=intg(0,dia/2,fun)
14 Pt=vec(1) + %pi/4 *dia^2 *(h-dia)*rho*g
15 // results
16 printf("Total pressure on base = %.2f kN",Pt/1000)
```

---

# Chapter 3

## Conservation Principle of Mass

Scilab code Exa 3.3 Example 1

```
1 clc
2 // Initialization of variables
3 d1=60 //cm
4 V1=45 //cm/s
5 d2=90 //cm
6 // calculations
7 V2=V1*d1^2 /d2^2
8 Q=%pi/4 *d1^2 *V1 *10^-6
9 // results
10 printf(" Velocity at point 2 = %d cm/s",V2)
11 printf(" \n FLOW rate = %.4f m^3/s",Q)
```

---

Scilab code Exa 3.4 Example 2

```
1 clc
2 // Initialization of variables
3 dn1=4 //cm
4 v1=300 //cm/s
```

```
5 dn2=2.5 //cm
6 //calculations
7 v2=v1*dn1/dn2
8 //results
9 printf("Velocity = %.1f m/s",v2/100)
```

---



## Chapter 4

# Conservation Principle of Momentum

Scilab code Exa 4.1 Example 1

```
1 clc
2 // Initialization of variables
3 Q=0.2 //m^3/s
4 v=30 //m/s
5 angle=120 //degrees
6 rho=1000 //kg/m^3
7 // calculations
8 Rx=rho*Q*(v-v*cosd(angle))
9 Ry=rho*Q*v*sind(angle)
10 R=sqrt(Rx^2 +Ry^2)
11 // results
12 printf("Resultant force = %.2 f kN",R/1000)
```

---

Scilab code Exa 4.3 Example 2

```
1 clc
```

```

2 // Initialization of variables
3 angle =45 //degrees
4 p1=150*10^3 //N/m^2
5 Q=0.5 //m^3/s
6 d1=60 //cm
7 d2=30 //cm
8 rho=1000 //kg/m^3
9 g=9.81 //m/s^2
10 // calculations
11 V1=Q/(%pi/4 *(d1/100)^2)
12 V2=V1*(d1/d2)^2
13 P2=rho*g*(p1/(rho*g) + V1^2 /(2*g) -V2^2 /(2*g))
14 Rx=p1*%pi/4*(d1/100)^2 - P2*%pi/4 *(d2/100)^2 *cosd(
    angle) -rho*Q*(V2*cosd(angle) -V1)
15 Ry=P2*%pi/4 *(d2/100)^2 *sind(angle) + rho*Q*(V2*
    sind(angle))
16 R=sqrt(Rx^2 + Ry^2)
17 // results
18 printf("resultant force = %.2 f kN",R/1000)

```

---

#### Scilab code Exa 4.4 Example 3

```

1 clc
2 // Initialization of variables
3 Q=20*10^3 //cc/s
4 depth=4 //m
5 d=5 //cm
6 g=9.81 //m/s^2
7 rho=10^3 //kg/m^3
8 // calculations
9 V1= Q/(%pi/4 *d^2) /100
10 V2= sqrt(2*g*(V1^2/(2*g) + depth))
11 W=rho*Q*(V2-V1)/10^6
12 // results
13 printf("weight of water = %d N",W)

```

---

**Scilab code Exa 4.5** Example 4

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 V=50 //m/s
6 u=20 //m/s
7 A=6/10^4 //m^2
8 angle=180 //degrees
9 // calculations
10 Vr=V-u
11 rq=rho*A*Vr
12 Rx=-rq*(Vr*cosd(angle) - Vr)
13 Rx2=-rho*A*V*(Vr*cosd(angle) -Vr)
14 power=Rx2*u
15 // results
16 printf("Force exerted on fluid = %d N",Rx)
17 printf(" \n Force transferred in case 2 = %d N",Rx2)
18 printf(" \n Power transferred in case 2 = %d kW",
        power/1000)
```

---

**Scilab code Exa 4.6** Example 5

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 Vr=10 //m/s
6 u=8.5 //m/s
7 A=250/10^4 //m^2
```

```

8 // calculations
9 V=Vr-u
10 Q=A*Vr
11 R=rho*Q*V
12 P=R*u
13 eth=1/(1+ V/(2*u))
14 // results
15 printf("Power required = %.3f kW",P/1000)
16 printf("\n Efficiency of jet propulsion = %.2f
    percent",eth*100)

```

---

#### Scilab code Exa 4.7 Example 6

```

1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 v1=20 //m/s
6 v2=5 //m/s
7 r1=50/100 //cm
8 r2=30/100 //cm
9 a1=20 //degrees
10 a2=80 //degrees
11 N=300 //rpm
12 Q=5 //m^3/s
13 // calculations
14 u1=%pi*2*r1*N/60
15 u2=%pi*2*r2*N/60
16 T=rho*Q*(r1*v1*cosd(a1) - r2*v2*cosd(a2))
17 H=1/g *(u1*v1*cosd(a1) - u2*v2*cosd(a2))
18 power=rho*g*Q*H
19 // results
20 printf("torque = %d N m",T)
21 printf("\n Heat = %.1f m",H)
22 printf("\n Power = %d kW",power/10^3)

```

23 //The answers given in textbook are a bit different  
due to rounding off error

---

#### Scilab code Exa 4.8 Example 7

```
1  clc
2  //Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3
5  d1=0.05 //m
6  d2=0.3 //m
7  N=1800 //rpm
8  Q=0.425/60 //m^3/s
9  //calculations
10 u1=%pi*d1*N/60
11 u2=%pi*d2*N/60
12 T=rho*Q*(d2*u2 - d1*u1)/2
13 //results
14 printf("Torque supplied = %.1f Nm",T)
```

---

# Chapter 5

## Conservation Principle of Energy

Scilab code Exa 5.1 Example 1

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 z2=0 //m
6 z1=8 //m
7 V2=5 //m/s
8 V1=3 //m/s
9 // calculations
10 Hs=(z2-z1) + (V2^2 -V1^2)/(2*g)
11 // results
12 printf("Work done by fluid = %.3f J/N",Hs)
```

---

Scilab code Exa 5.2 Example 2

```
1 clc
```

```

2 clear
3 // Initialization of variables
4 g=9.81 //m/s^2
5 rho=10^3 //kg/m^3
6 P1=80*10^3 //N/m^2
7 P2=12*10^6 + 101300 //N/m^2
8 Hq=-400 //J/N
9 //calculations
10 g1=g*rho
11 Hs= -Hq+ (P2-P1)/(g1)
12 //results
13 printf("Energy added by pump = %d J/N",Hs)
14 disp("The answer given in textbook is wrong. Please
    verify using a calculator")

```

---

### Scilab code Exa 5.3 Example 3

```

1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 d1=15 //cm
6 d2=10 //cm
7 V1=2.4 //m/s
8 P1=450*10^3 //N/m^2
9 rho2=900 //kg/m^3
10 //calculations
11 V2=d1^2 /d2^2 *V1
12 P2=g*rho2*(P1/(rho2*g) + V1^2 /(2*g) - V2^2 /(2*g))
13 Q=%pi/4*(d2/100)^2 *V2
14 //results
15 printf("Pressure at 2 = %.2f kN/m^2",P2/1000)
16 printf("\n Flow rate = %.4f m^3/s",Q)
17 //The answer given in textbook is wrong. Please
    verify it.

```

---

**Scilab code Exa 5.4** Example 4

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 z=10 //m
6 // calculations
7 PE=g*rho*%pi*z^2 /2
8 // results
9 printf("Work obtained = %.2e J",PE)
```

---

**Scilab code Exa 5.6** Example 5

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 d1=7.5 //cm
6 d2=3 //cm
7 P1=300+101.3 //kPa
8 P2=25 //kPa
9 // calculations
10 V1=sqrt(2*g/ ((d1/d2)^4 -1) *(P1*10^3 /(rho*g) -P2
    *10^3 /(rho*g)))
11 Q=%pi/4 *(d1/100)^2 *V1
12 // results
13 printf("Max discharge = %.4f m^3/s",Q)
14 //The answer given in textbook is wrong. Please use
    a calculator to verify
```

---



### Scilab code Exa 5.7 Example 6

```
1  clc
2  // Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3
5  z1=1.2 //m
6  z2=4 //m
7  d=5 //cm
8  // calculations
9  Va=sqrt(2*g*(z2-z1))
10 Q=%pi/4 *(d/100)^2 *Va
11 Pc= - z2*rho*g
12 P=25*10^3 //Pa
13 Zab=(101325 - P)/rho/g
14 // results
15 printf("rate of discharge = %.4f m^3/s",Q)
16 printf("\n Pressure at C = %.2f kPa",Pc/1000)
17 printf("\n Max. permissible length = %.2f m",Zab)
```

---

### Scilab code Exa 5.8 Example 7

```
1  clc
2  // Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3
5  Q=0.09 //m^3/s
6  d1=0.12 //m
7  d2=0.2 //m
8  P1=80 //kN/m^2
9  P2=120 //kN/m^2
10 // calculations
```

```

11 V1=Q/(%pi/4 *d1^2)
12 TE1 = P1*10^3 /(rho*g) + V1^2 /(2*g)
13 V2= d1^2 /d2^2 *V1
14 TE2= P2*10^3 /(rho*g) + V2^2 /(2*g)
15 //results
16 if TE1>TE2 then
17     printf("Flow is from section 1 to section 2")
18 else
19     printf("Flow is from section 2 to section 1")
20 end

```

---

#### Scilab code Exa 5.9 Example 8

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 Q=0.012 //m^3/s
6 z=10 //m
7 d=0.075 //m
8 //calculations
9 Vb=Q/(%pi/4 *d^2)
10 Hm=z+ Vb^2 /(2*g)
11 P=Hm*rho*g*Q
12 //results
13 printf("Power required = %.3f kW",P/1000)

```

---

#### Scilab code Exa 5.10 Example 9

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=860 //kg/m^3

```

```

5 P1=20 *10^3 //Pa
6 P2=50*10^3 //Pa
7 z=2.8 //m
8 d1=0.1 //m
9 //calculations
10 V1=sqrt(2*g*(P2/(rho*g) -z - P1/(rho*g)))
11 Q=%pi/4 *d1^2 *V1
12 //results
13 printf("rate of flow = %.4f m^3/s",Q)

```

---

#### Scilab code Exa 5.11 Example 10

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 Cv=0.92
6 P=210*10^3 //Pa
7 d=0.05 //m
8 ret=1.5 //m/s^2
9 //calculations
10 H=P/(g*rho)
11 Va=Cv*(2*g*H)
12 h=Cv^2 *H
13 h2= Cv^2 *2*g*H/(2*(g+ret))
14 //results
15 printf("The height to which the jet will rise is %.2
    f m",h)
16 printf("\n In case 2., height = %.2f m",h2)

```

---

#### Scilab code Exa 5.12 Example 11

```

1 clc

```

```

2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 h=4 //m
6 d=0.03 //m
7 Qa=3.8/1000 //m^3/s
8 x=2.5 //m
9 y=0.41 //m
10 //calculations
11 Qth = %pi/4 *d^2 *sqrt(2*g*h)
12 Cd=Qa/Qth
13 Cv=sqrt(x^2 /(4*y*h))
14 Cc=Cd/Cv
15 //results
16 printf("Cd = %.2 f", Cd)
17 printf("\n Cv = %.3 f", Cv)
18 printf("\n Cc= %.2 f", Cc)

```

---

### Scilab code Exa 5.13 Example 12

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 rho2=13.6*10^3 //kg/m^3
6 d1=3.2 //m
7 d2=0.6 //m
8 //calculations
9 z1=d1*rho/rho2
10 head= d2+z1
11 V=sqrt(2*g*head)
12 //results
13 printf("Efflux velocity = %.2 f m/s", V)
14 //The answer is a bit different due to rounding off
    error.

```

---

**Scilab code Exa 5.15** Example 15

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 Cd=0.6
6 d=0.04 //m
7 h2=2.5 //m
8 // calculations
9 function y=fun(h)
10     y=1/(Cd*%pi/4 *d^2 *sqrt(2*g)) *(4/sqrt(h) +
        sqrt(64-h^2))
11 endfunction
12 t=intg(0,h2,fun)
13 tmin=31.1
14 // results
15 printf("Time required = %.1f min",tmin)
```

---

**Scilab code Exa 5.16** Example 16

```
1 clc
2 // Initialization of variables
3 g=981 //cm/s^2
4 Cd=0.6
5 Q=1200
6 d=3 //cm
7 l=30 //cm
8 b=30 //cm
9 dh=5 //cm
10 h1=9 //cm
```

```

11 //calculations
12 function y =fun1(h)
13     y= l*b/(Q - Cd*%pi/4 *d^2 *sqrt(2*g*h))
14 endfunction
15 t=intg(h1,h1+dh,fun1)
16 t=126
17 //results
18 printf("Time required = %d sec",t)

```

---

#### Scilab code Exa 5.17 Example 17

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 pst=25.2*10^3 //Pa
6 h=2.5 //m
7 //calculations
8 v=sqrt(2/rho *(pst - g*rho*h))
9 //results
10 printf("velocity = %.2f m/s",v)

```

---

#### Scilab code Exa 5.18 Example 18

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 vel=800*10^3 /3600
6 sm=13.57
7 sl2=12.2
8 //calculations
9 sl=s12/(g*rho)

```

```

10 y=vel^2 /(2*g*(sm/sl -1))
11 //results
12 printf("length of manometer = %d cm",y*100)

```

---

#### Scilab code Exa 5.19 Example 19

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 h=3.5 //m
6 //calculations
7 v=sqrt(2*g*h)
8 //results
9 printf("Speed necessary = %.1 f m/s",v)

```

---

#### Scilab code Exa 5.20 Example 20

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 sm=13.6
6 s=1
7 Q=1 //m^3/s
8 d2=0.25 //m
9 d1=0.5 //m
10 nu=1e-6
11 //calculations
12 RN=Q*d1/(%pi/4 *d1^2 *nu)
13 Cv=0.98
14 yd= Q^2 *(1-d2^4 /d1^4)/(Cv^2 *%pi/4 *d2^2 *2*g)
15 y=yd/(sm/s -1)

```

```
16 //results
17 printf("Mercury manometer reading = %.2f cm",y*100)
```

---

#### Scilab code Exa 5.21 Example 21

```
1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 sm=13.6
6 s=1
7 y=0.12 //m
8 Cv=0.984
9 d1=0.05 //m
10 d2=0.1 //m
11 nu=1e-6
12 //calculations
13 Q=Cv*%pi/4 *d1^2 *sqrt(2*g) /sqrt(1- (d1/d2)^4) *
    sqrt(y*(sm/s -1))
14 V1=Q/(%pi/4 *d2^2)
15 R=V1*d1/nu
16 //results
17 printf("Since , reynolds number is in required value ,
    Flow rate = %.4f m^3/s",Q)
```

---

#### Scilab code Exa 5.22 Example 22

```
1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 P1=150*10^3 //Pa
6 d0=3 //cm
```



```

7 d1=6 //cm
8 Cv=0.98
9 Cc=0.62
10 //calculations
11 P1g=P1/(g*rho)
12 Ar= (d0/d1)^4
13 A0=%pi/4 *(d0/100)^2
14 Q= Cv*Cc*A0 *sqrt(2*g) /sqrt(1- Cc^2 *Ar) *sqrt(P1g)
15 //results
16 printf("Discharge = %.2f lps",Q*10^3)

```

---

#### Scilab code Exa 5.23 Example 23

```

1 clc
2 clear
3 //Initialization of variables
4 g=9.81 //m/s^2
5 rho=10^3 //kg/m^3
6 Cd=0.6
7 L=3 //m
8 H=0.4 //m
9 V0=[0 0.24 0.275]
10 //calculations
11 Q= Cd*2/3 *sqrt(2*g) *(L-0.2*H) *((H+ V0.^2 ./ (2*g)
    ).^(3/2) - (V0.^2 ./ (2*g)).^(3/2))
12 //results
13 H=max(Q)
14 printf("Flow rate = %.3f m^3/s",H)

```

---

#### Scilab code Exa 5.24 Example 24

```

1 clc
2 //Initialization of variables

```

```

3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 d=0.5 //m
6 vel=1 //m/s
7 depth=1.2 //m
8 Cd=0.62
9 //calculations
10 H=(d*3/(2*Cd))^(2/3)
11 hw=depth-H
12 //results
13 printf("height of weir plate = %.2f m",hw)
14 //The answer given in textbook is wrong please use a
    caclculator.

```

---

**Scilab code Exa 5.25** Example 25

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 Q=0.1*100^2 /(24*3600) //m^3/s
6 Cd=0.61
7 theta=60 //degrees
8 Hd=Q/(Cd*8/15 *sqrt(2*g) *tand(theta/2))
9 H=Hd^(2/5)
10 //results
11 printf("apex of weir must be set %.1f cm below the
    free surface",H*100)
12 //The answer in the textbook is wrong. Please verify
    it

```

---

**Scilab code Exa 5.26** Example 26

```

1  clc
2  // Initialization of variables
3  Q1=0.93
4  Q2=0.4
5  H1=0.7
6  H2=0.5
7  // calculations
8  n=log(Q1/Q2) /log(H1/H2)
9  // results
10 printf("Shape n = %.1f . hence shape of weir is
        triangular",n)

```

---

**Scilab code Exa 5.27** Example 27

```

1  clc
2  // Initialization of variables
3  g=981 //cm/s^2
4  H=20 //cm
5  err=3/100
6  // calculations
7  dH=err/2.5 *H
8  v0=sqrt(2*g*dH)
9  // results
10 printf("Required velocity = %.2f cm/s",v0)
11 //The answer is a bit different due to rounding off
    error

```

---

**Scilab code Exa 5.28** Example 28

```

1  clc
2  // Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3

```

```
5 Q=12000
6 f=30
7 t1=0.5
8 t2=1.2
9 //calculations
10 function y= fun2(h)
11     y=Q/f *(1/h^(3/2))
12 endfunction
13 t=intg(t1,t2,fun2)
14 //results
15 printf("Time = %d sec",t)
16 //The answer is a bit different due to rounding off
    error
```

---

# Chapter 6

## Dimensional Analysis and Similitude

Scilab code Exa 6.11 Example 11

```
1  clc
2  // Initialization of variables
3  dw=1000 //kg/m^3
4  muw=0.001 //N s /m^2
5  da=1.225 //kg/m^3
6  mua=18*10^-6 //N s /m^2
7  lr=1/10
8  // calculations
9  dr=da/dw
10 mur=mua/muw
11 vr=mur/dr
12 velocity= vr/lr
13 discharge =lr*vr
14 pressure = mur^2 /(dr*lr^2)
15 force = mur^2 /dr
16 // results
17 printf("Scale ratio for velocity = %d ",velocity)
18 printf("\\nScale ratio for discharge = %.2f ",
    discharge)
```

```
19 printf("\nScale ratio for pressure = %.1f ",pressure
    )
20 printf("\nScale ratio for force = %.3f ",force)
```

---

#### Scilab code Exa 6.12 Example 12

```
1 clc
2 // Initialization of variables
3 dr=1000
4 mur=100
5 lr=1/10
6 dpm=60
7 // calculations
8 Vr=mur/dr/lr
9 dpr=dr*Vr^2
10 dpp=dpm/dpr
11 // results
12 printf("Pressure drop in prototype = %d N/m^2",dpp
    *10^3)
```

---

#### Scilab code Exa 6.14 Example 14

```
1 clc
2 // Initialization of variables
3 lr=1/25
4 Tp=6 //sec
5 dr=1/1.025
6 Fm=70 //N
7 // calculations
8 Tr=lr^(0.5)
9 Tm=Tr*Tp
10 Fr=dr*lr^3
11 Fp=Fm/Fr
```

```
12 //results
13 printf("Wave period = %.1 f sec",Tm)
14 printf("Force = %.3 f kN",Fp/1000)
```

---

**Scilab code Exa 6.16** Example 16

```
1 clc
2 //Initialization of variables
3 lr=1/10
4 Vp=10 //knots
5 Fm=12 //N
6 //calculations
7 Vm=Vp*sqrt(lr)
8 Fp=Fm/lr^3
9 //results
10 printf("force = %.1 f kN",Fp/1000)
```

---

**Scilab code Exa 6.17** Example 17

```
1 clc
2 //Initialization of variables
3 lr=1/7200
4 //calculations
5 Tr=60/(12*3600)
6 yr=(lr/Tr)^2
7 //results
8 printf("vertical scale to be adopted is 1 in %d",1/yr)
```

---

# Chapter 7

## In compressible Flow through Conduits

Scilab code Exa 7.1 Example 1

```
1  clc
2  // Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3
5  h1=4 //m
6  muw=0.001 //Ns/m^2
7  l=1.5 //m
8  B=0.15/1000 //m
9  len=11.2 //m
10 // calculations
11 P1=g*rho*h1
12 V=P1*B^2 /(12*muw*l)
13 A=B*len
14 Q=A*V
15 Q=7112.4
16 tau= B/2 *(P1)/l
17 // results
18 printf("Average velocity through the crack = %.3f m/
    s",V)
```



```
19 printf("\n rate of leakage = %.1f l/hr",Q)
20 printf("\n Shear stress = %.3f N/m^2",tau)
```

---

### Scilab code Exa 7.2 Example 2

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=1200 //kg/m^3
5 mu=0.005 //Ns/m^2
6 d=0.006 //m
7 Re=2000
8 V=0.15 //m/s
9 // calculations
10 Vc=Re*mu/(d*rho)
11 Vr=V/Vc
12 T0=8*mu*V/d
13 // results
14 printf("Shear stress = %d N/m^2",T0)
```

---

### Scilab code Exa 7.3 Example 3

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 Q=0.45/(60*1000) //m^3/s
6 d=0.003 //m
7 depth=0.95 //m
8 alpha=2
9 len=1.25 //m
10 // calculations
11 A=%pi/4 *d^2
```

```

12 V=Q/A
13 nu= (depth - alpha*V^2 /(2*g))*g*d^2 /(32*V*len)
14 Re=V*d/nu
15 //results
16 if Re<2000 then
17     printf("Flow is laminar")
18 else
19     printf("Flow is not laminar")
20 end

```

---

#### Scilab code Exa 7.4 Example 4

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=787 //kg/m^3
5 Q=90*10^-3 //m^3/hr
6 d=0.015 //m
7 k=0.0045*10^-2 //m
8 nu=1.6e-6
9 l=5 //m
10 //calculations
11 V=Q/(60*%pi/4 *d^2)
12 Rn=V*d/nu
13 e=k/d
14 disp("From moody diagram , f=0.028")
15 f=0.028
16 hl=f*l/d *V^2 /(2*g)
17 Power=rho*g*Q/60 *hl
18 //result
19 printf("Head loss = %.2 f m" ,hl)
20 printf("\n power required =%.3 f kW" ,Power/1000)

```

---

### Scilab code Exa 7.5 Example 5

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=870 //kg/m^3
5 Q=2*10^-3 //m^3/s
6 d=0.03 //m
7 mu=5*10^-4
8 l=50 //m
9 // calculations
10 V=Q/(%pi/4 *d^2)
11 RN=rho*V*d/mu
12 f=0.017
13 hl=f*l/d *V^2/(2*g)
14 Ploss=rho*g*hl
15 // results
16 printf("Loss of pressure = %.1f kN/m^2",Ploss/1000)
17 //The answers are a bit different due to rounding
    off error in textbook
```

---

### Scilab code Exa 7.6 Example 6

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=813 //kg/m^3
5 Q=0.007 //m^3/hr
6 d=0.01 //m
7 mu=0.002 //Ns/m^2
8 l=30 //m
9 // calculations
10 V=Q/(60*%pi/4*d^2)
11 RN=V*d*rho/mu
12 f=0.316/RN^(0.25)
```

```
13 h=(1+f*l/d)*V^2 /(2*g)
14 //result
15 printf("Height required = %.2 f m",h)
```

---

#### Scilab code Exa 7.7 Example 7

```
1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 h1=0.02
6 d=1.2 //m
7 l=1 //m
8 k=0.5 *10^-2 //m
9 //calculations
10 v2f=h1*(2*g*d)/l
11 e=k/d
12 f=0.028
13 V=sqrt(v2f/f)
14 Q=%pi/4 *d^2 *V
15 //results
16 printf("Rate of flow = %.2 f m^3/s",Q)
```

---

#### Scilab code Exa 7.8 Example 8

```
1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 e=0.03*10^-2 //m
6 l=3000 //m
7 Q=300*10^-3 //m^3/s
8 nu=10^-5 //m^2/s
```

```

9 h1=24 //m
10 //calculations
11 d5f= 1*Q/(%pi/4) * Q/(%pi/4) /(h1*2*g)
12 f=0.022
13 d=(d5f*f)^(1/5)
14 //results
15 printf("Size of the required pipe = %d cm",d*100)

```

---

#### Scilab code Exa 7.9 Example 9

```

1 clc
2 //Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 d=0.3 //m
6 per=25/100
7 Q=0.1 //m^3/s
8 k0=0.025*10^-2 //m
9 nu=0.000001
10 year=10
11 //calculations
12 V=Q/(%pi/4 *d^2)
13 RN=V*d/nu
14 e1=k0/d
15 f1=0.019
16 f2=(1+per)*f1
17 e2=0.002
18 k2=e2*d
19 rate = (k2-k0)*100/year
20 //results
21 printf("Rate of increase =%.4 f cm/year",rate)

```

---

#### Scilab code Exa 7.10 Example 10

```

1  clc
2  // Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3
5  l=1 //m
6  b=0.3 //m
7  Q=4.2 //m^3/s
8  // calculations
9  A=l*b
10 R=A/(2*(1+b))
11 d5=1.62/24.15
12 d=d5^(1/5)
13 Pr=2*(1+b)/(pi*d)
14 // results
15 printf("The rectangular cross section will cost %.2
    f times that of a circular cross section",Pr)

```

---

#### Scilab code Exa 7.11 Example 11

```

1  clc
2  // Initialization of variables
3  g=9.81 //m/s^2
4  rho=10^3 //kg/m^3
5  d1=2.5*10^-2 //m
6  d2=7.2*10^-2 //m
7  Q=100*10^-3 //m^3/hr
8  // calculations
9  V1=Q/(60*pi/4*d1^2)
10 V2=(d1/d2)^2 *V1
11 dp= -(V2^2 -V1^2 + (V1-V2)^2)/(2*g)
12 Pdiff=dp*g*rho
13 // results
14 printf("pressure difference = %.2f kN/m^2",Pdiff
    /1000)
15 //The answers are a bit different due to rounding

```

off error

---

### Scilab code Exa 7.12 Example 12

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 d2=30/100 //cm
6 d1=60/100 //cm
7 Pu=105 //kN/m^2
8 Pd=75 //kN/m^2
9 Cc=0.65
10 // calculations
11 V22=(2*g/(1 - (d2/d1)^4 + (1/Cc -1)^2)) *(Pu-Pd)
    *10^3 /(rho*g)
12 V2=sqrt(V22)
13 Q=%pi/4 *V2 *d2^2
14 h1=(1/Cc -1)^2 *V2^2 /(2*g)
15 // results
16 printf("Flow rate = %.3f m^3/s",Q)
17 printf("\n Head loss = %.3f m",h1)
```

---

### Scilab code Exa 7.13 Example 13

```
1 clc
2 // Initialization of variables
3 g=9.81 //m/s^2
4 rho=10^3 //kg/m^3
5 d=9 //m
6 dia=0.3 //m
7 // calculations
8 V302= 2*g*d/(0.5 + 20 + 2.53+101+0.66+41.47+2.07)
```

```
9 V30=sqrt(V302)
10 Q=%pi/4 *dia^2 *V30
11 //results
12 printf("Flow rate = %.3f m^3/s",Q)
```

---

#### Scilab code Exa 7.14 Example 14

```
1 clc
2 clear
3 //Initialization of variables
4 h=6 //m
5 rho=930 //kg/m^3
6 Q=3/60 //m^3/s
7 d=0.15 //m
8 L=20 //m
9 mu=0.006
10 g=9.81 //m/s^2
11 //calculations
12 V=Q/(%pi/4 *d^2)
13 RN=V*d*rho/mu
14 f=0.316/RN^0.25
15 hl=f*L/d *V^2 /(2*g)
16 Hp=h+hl
17 gam=rho*g
18 W=gam*Q
19 Power= W*Hp
20 //results
21 printf("Power required = %.3f kW",Power/1000)
```

---

#### Scilab code Exa 7.15 Example 15

```
1 clc
2 //Initialization of variables
```



```

3 d=0.02 //m
4 d2=1.2 //m
5 f=0.01
6 L=250
7 ken=0.5
8 g=9.81
9 h1=8 //m
10 h2=4 //m
11 // calculations
12 V2=2*g/(1+ken+ f*L/d)
13 V=sqrt(V2)
14 Q=%pi/4 *d^2 *V
15 function t=time(h)
16     t=-%pi/4 *d2^2 /Q /sqrt(h)
17 endfunction
18 ti=intg(h1,h2,time)
19 // results
20 printf("Time required = %d sec",ti)

```

---

#### Scilab code Exa 7.16 Example 16

```

1 clc
2 // Initialization of variables
3 d1=0.1 //m
4 d2=0.05 //m
5 l1=20 //m
6 l2=20 //m
7 f=0.02
8 // calculations
9 K1=(f*l2/d2 *(d1/d2)^4 - f*l1/d1)
10 // results
11 printf("Loss coefficient = %d ",K1)

```

---

Scilab code Exa 7.17 Example 17

```
1 clc
2 clear
3 // Initialization of variables
4 g=9.81
5 rati=1.265
6 // calculations
7 percent = (rati-1)*100
8 // results
9 printf("Increase in discharge = %.1f",percent)
```

---

Scilab code Exa 7.18 Example 18

```
1 clc
2 // Initialization of variables
3 Q=0.6 //m^3/s
4 l1=1200 //m
5 l2=800 //m
6 d1=0.3 //m
7 // calculations
8 V1=1.02 //m/s
9 d5= d1*l2*4^2 *Q^2 /(l1*%pi^2 *V1^2)
10 d=d5^(1/5)
11 // results
12 printf("diameter of the single pipe = %.2f m",d)
```

---

Scilab code Exa 7.19 Example 19

```
1 clc
2 // Initialization of variables
3 g=9.81
4 Q=0.18 //m^3/s
```

```

5 d3=0.3 //m
6 f=0.032
7 L3=360 //m
8 z=25.5 //m
9 z2=30 //m
10 L2=450 //m
11 d2=0.45 //m
12 L1=950 //m
13 d1=0.45 //m
14 zn=18 //m
15 rho=1000
16 // calculations
17 V3=Q/(%pi/4 *d3^2)
18 h13=f*L3/d3 *(V3^2 /(2*g))
19 Z2=z+h13
20 h12=Z2-z2
21 V2= sqrt(2*g*d2*h12/(f*L2))
22 Q2=%pi/4 *d2^2 *V2
23 V1=V2+ (d3/d2)^2 *V3
24 h11=f*L1/d1*V1^2 /(2*g)
25 Hp= h11+ Z2-zn
26 gam=rho*g
27 P=gam*Hp
28 // results
29 printf("Discharge into the reservoir = %.3f m^3/s",
        Q2)
30 printf("\n Pressure maintained by the pump = %.2f kN
        /m^2",P/1000)

```

---

#### Scilab code Exa 7.20 Example 20

```

1 clc
2 // Initialization of variables
3 h=[1 2 1.9 1.96]
4 z1=10 //m

```

```

5 z2=5 //m
6 z3=7.5 //m
7 f=0.04
8 l1=100 //m
9 l2=50 //m
10 l3=70 //m
11 d1=0.1 //m
12 d2=0.075 //m
13 d3=0.06 //m
14 g=9.81 //m/s^2
15 //calculations
16 Q1=sqrt(d1^5 *(%pi/4)^2 *2*g/(f*l1)) .*sqrt(z1-h)
17 Q2=sqrt(d2^5 *(%pi/4)^2 *2*g/(f*l2)) .*sqrt(h+z2)
18 Q3=sqrt(d3^5 *(%pi/4)^2 *2*g/(f*l3)) .*sqrt(h+z3)
19 len=length(h)
20 for i=1:len
21     Q=Q2(i)+Q3(i)
22     if (Q1(i) == Q) then
23         break;
24     end
25 end
26 printf("height h = %.2 f m",h(i))
27 printf("\nDischarge in BC Q2 = %.2 f lps",Q2(i)*1000)
28 printf("\nDischarge in BD Q3 = %.2 f lps",Q3(i)*1000)

```

---

### Scilab code Exa 7.21 Example 21

```

1 clc
2 //Initialization of variables
3 e=0.8
4 output=400 //kW
5 H=150 //m
6 rho=1000
7 g=9.81
8 f=0.028

```

```

9 l=1250 //m
10 //calculations
11 gam=rho*g
12 inpu=output/e
13 Q=inpu*10^3 /(2/3 *gam*H)
14 hl=1/3 *H
15 d5= f*l*Q^2 /(2*g* %pi/4 * %pi/4 *hl)
16 d=d5^(1/5)
17 //results
18 printf("Smallest diameter of pen stock = %d cm",d
        *100)

```

---

#### Scilab code Exa 7.22 Example 22

```

1 clc
2 //Initialization of variables
3 f=0.04
4 H=30 //m
5 l=200 //m
6 d=0.075 //m
7 g=9.81
8 rho=1000
9 gam=rho*g
10 //calculations
11 h=2/3 *H
12 vj=sqrt(2*g*h)
13 hl= 1/3 *H
14 V= sqrt(hl*d*2*g/(f*l))
15 dj= d*(sqrt(V/vj))
16 Power= 2/3 *gam*%pi/4 *d^2 *V*H
17 //results
18 printf("Size of nozzle = %.1f cm",dj*100)
19 printf("\n Max power = %.2f kW",Power/1000)

```

---

# Chapter 8

## Uniform Open Channel Flow

Scilab code Exa 8.1 Example 1

```
1  clc
2  // Initialization of variables
3  b=4 //m
4  y=1.2 //m
5  sf=0.001
6  n=0.012
7  gam=9.81*1000
8  // calculations
9  A=b*y
10 R=A/(b+ 2*y)
11 Q=1/n *A*R^(2/3) *sf^(1/2)
12 T=gam*R*sf
13 // results
14 printf(" Discharge = %.3f m^3/s",Q)
15 printf("\n bed shear = %.2f N/m^2",T)
16 //The answer in textbook is wrong for discharge.
    Please use a calculator.
```

---

Scilab code Exa 8.2 Example 2

```

1  clc
2  // Initialization of variables
3  b=6 //m
4  y=2 //m
5  sf=0.005
6  slope = 2
7  gam=9.81*1000
8  Q=65 //m^3/s
9  // calculations
10 A=(b+ 2*y)*slope
11 P=b+ 2*y*sqrt(slope^2 +1)
12 R=A/P
13 V=Q/A
14 n=R^(2/3) *sf^(1/2) /V
15 // results
16 printf("Value of mannings coefficient = %.3f",n)

```

---

### Scilab code Exa 8.3 Example 3

```

1  clc
2  // Initialization of variables
3  b=3 //m
4  y=1 //m
5  sf=0.005
6  n=0.028
7  gam=9.81*1000
8  Q=0.25 //m^3/s
9  slope=1.5
10 // calculations
11 A= 0.5 *b*y
12 P=2*sqrt(1 + (slope)^2)
13 R=A/P
14 yx= Q*n/(slope * R^(2/3) *sf^(1/2))
15 y= yx^(3/8)
16 // results

```

```
17 printf("depth = %.2f m",y)
```

---

#### Scilab code Exa 8.4 Example 4

```
1 clc
2 //Initialization of variables
3 sf=0.0064
4 n=0.015
5 Q=6 //m^3/s
6 gam=9.81*1000
7 //calculations
8 AR= n*Q/sqrt(sf)
9 disp("On trial and error , ")
10 y=0.385 //m
11 printf("normal depth = %.3f m",y)
```

---

#### Scilab code Exa 8.5 Example 5

```
1 clc
2 //Initialization of variables\
3 sf=0.00007
4 n=0.013
5 gam=9.81*1000
6 V=0.45 //m/s
7 Q=1.4 //m^3/s
8 //calculations
9 by=Q/V
10 x=poly(0,"x")
11 y=roots(x^2 -2.66*x +1.55)
12 b=by ./y
13 //results
14 printf("y = ")
15 disp( y )
```



```
16 printf("corresponding b=")
17 disp(b)
```

---

#### Scilab code Exa 8.6 Example 6

```
1 clc
2 // Initialization of variables
3 sf=0.0016
4 n=0.02
5 Q=0.84 //m^3/s
6 gam=9.81*1000
7 // calculations
8 y53= Q*n/sqrt(sf)
9 y=y53^(3/5)
10 // results
11 printf("depth of flow = %.2f m",y)
```

---

#### Scilab code Exa 8.7 Example 7

```
1 clc
2 // Initialization of variables
3 n=0.015
4 Q=1.3 //m^3/s
5 V=0.6 //m/s
6 gam=9.81*1000
7 // calculations
8 alpha=60 //degrees
9 A=0.5 *(1/2)^2 *(180-alpha)/180 *%pi -(1/4)^2 *tand(
    alpha)
10 A=0.206
11 P=0.5*(180-alpha)/180 *%pi
12 R=A/P
13 d2=V*n/(R^(2/3))
```

```

14 d8= Q*n*4*4^(2/3) /%pi
15 d=sqrt(d8/d2)
16 sf= (d2/d^(2/3))^2
17 //results
18 printf("Diameter = %.2f m",d)
19 printf("\n slope = %.5f ",sf)
20 //The answer given in textbook is wrong. please
    check

```

---

#### Scilab code Exa 8.8 Example 8

```

1  clc
2  //Initialization of variables
3  b=0.5 //m
4  y=0.35 //m
5  sf=0.001
6  nc=0.016
7  gam=9.81*1000
8  Q=0.15 //m^3/s
9  //calculations
10 A=b*y
11 P= b+ 2*y
12 R=A/P
13 ng=1/Q *A*R^(2/3) *sf^(1/2)
14 n= (b*nc^(3/2) + 2*y*ng^(3/2))^(2/3) /(P^(2/3))
15 Q2=1/n *A*R^(2/3) *sf^(1/2)
16 //results
17 printf("flow in case 2 = %.3f m^3/s",Q2)

```

---

#### Scilab code Exa 8.9 Example 9

```

1  clc
2  //Initialization of variables

```

```

3 b1=8 //m
4 b2=5 //m
5 y=5 //m
6 b5=15 //m
7 b3=3 //m
8 b4=3 //m
9 y2=2 //m
10 y3=3 //m
11 n1=0.025
12 n2=0.035
13 sf=0.0008
14 //calculations
15 A= (b1+b2)*y
16 P= b1+ sqrt(b2^2 +y^2) + sqrt(b3^2 +b4^2)
17 R=A/P
18 Q1=1/n1 *A*R^(2/3) *sf^(1/2)
19 A2 = b5*y2 - 0.5*y2*y2 + 0.5*y3*y2
20 P2= b5 + sqrt(b4^2 + y3^2)
21 R2=A2/P2
22 Q2= 1/n2 *A2*R2^(2/3) *sf^(1/2)
23 Q=Q1+Q2
24 //results
25 printf("Total discharge = %.1f m^3/s",Q)

```

---

### Scilab code Exa 8.10 Example 10

```

1 clc
2 //Initialization of variables
3 Q=12 //m^3/s
4 n=0.023
5 A=2.472
6 b=0.472
7 sf=1/8000
8 //calculations
9 y8= Q*n/A *2^(2/3) /sf^(1/2)

```

```
10 y=y8^(3/8)
11 b2= b*y
12 //results
13 printf("depth = %.3f m",y)
14 printf("\n width = %.2f m",b2)
```

---

#### Scilab code Exa 8.11 Example 11

```
1 clc
2 //Initialization of variables
3 Q=30
4 V=1
5 //calculations
6 A=Q/V
7 y = sqrt(A/(sqrt(2) + 0.5))
8 b= (A- 0.5*y^2)/y
9 //results
10 printf("width = %.2f m",b)
11 printf("\n depth = %.2f m",y)
```

---

# Chapter 9

## Potential Flow

Scilab code Exa 9.4 Example 4

```
1 clc
2 // Initialization of variables
3 k=1.5
4 r=40 //cm
5 theta=45 //degrees
6 // calculations
7 vr= -2*k*r*cosd(2*theta)
8 vt= 2*k*r*sind(2*theta)
9 // results
10 printf("velocity in radial direction = %d cm/s",vr)
11 printf("\\n velcoity in angular direction = %d cm/s",
    vt)
```

---

Scilab code Exa 9.14 Example 14

```
1 clc
2 // Initialization of variables
3 T=4.5
```

```

4 a=0.6
5 u=5 //m/s
6 rho=1000 //kg/m^3
7 //calculations
8 sint=0.5*(1- T/(2*pi*a*u))
9 theta= asind(sint)
10 dp= 0.5*rho*u^2 *(1 - (2 + T/(2*pi*a*u))^2)
11 //results
12 printf("Angle = %.1f %.1f degrees",theta,180-theta)
13 printf("\n Min guage pressure = %.2f kN/m^2",dp
    /1000)
14 //The answer in textbook is wrong. please check

```

---

#### Scilab code Exa 9.15 Example 15

```

1 clc
2 //Initialization of variables
3 T=6*pi
4 r=1/3
5 //calculations
6 vab=T/(4*pi)
7 vba= T/(2*pi)
8 w=vab/r
9 //results
10 printf("rate of rotation = %.1f rad/s",w)
11 printf("\nspeed of A by B = %.1f m/s",vab)
12 printf("\nspeed of B by A = %.1f m/s",vba)

```

---

# Chapter 10

## The Boundary Layer

Scilab code Exa 10.1 Example 1

```
1 clc
2 // Initialization of variables
3 v=30 //m/s
4 nu=1.5e-5 //m^2/s
5 // calculations
6 Re=5*10^5
7 xc= Re*nu/v
8 // results
9 printf("Transistion region = %.2f m",xc)
```

---

Scilab code Exa 10.2 Example 2

```
1 clc
2 // Initialization of variables
3 u=2 //m/s
4 x=0.15 //m
5 nu=1.5e-5 //m^2/s
6 B=0.5 //m
```

```

7 rho=1.22 //kg/m^3
8 //calcualtions
9 Rx=u*x/nu
10 delta= 4.91*x/sqrt(Rx)
11 deltas=1.729*x/sqrt(Rx)
12 Cf=1.328/sqrt(Rx)
13 Ff=Cf*0.5*rho*u^2 *2*B*x
14 //results
15 printf("Boundary layer thickness = %.2f cm",delta
        *100)
16 printf("\n Displacement thickness = %.2f cm",deltas
        *100)
17 printf("\n Average drag coefficient = %.4f",Cf)
18 printf("\n Drag force = %.4f N",Ff)

```

---

#### Scilab code Exa 10.5 Example 5

```

1 clc
2 //Initialization of variables
3 U=172*1000/3600 //m/s
4 w=3 //m
5 h=3 //m
6 L=100 //m
7 nu=1.5e-5 //m^2/s
8 rho=1.22 //kg/m^3
9 //calculations
10 Rl=U*L/nu
11 Cf=0.074 /(Rl^(1/5))
12 Ff=Cf*0.5*rho*U^2 *w*h*L
13 power= Ff*U
14 //results
15 printf("power required = %.1f kW",power/1000)
16 //The answer is a bit different due to rounding off
    error

```

---



Scilab code Exa 10.6 Example 6

```
1  clc
2  // Initialization of variables
3  U=4000 //m/s
4  L=8 //m
5  nu=3600e-6 //m^2/s
6  rho=1000 //kg/m^3
7  b=5 //m
8  // calculations
9  Rl=U*L/nu
10 Cf= 0.074/Rl^(1/5) -1700/Rl
11 Ff=Cf*0.5*rho*(U/3600)^2 *L*b
12 // results
13 printf("Skin friction drag = %.2f N",Ff)
```

---

# Chapter 11

## Forces on Immersed Bodies

Scilab code Exa 11.1 Example 1

```
1 clc
2 // Initialization of variables
3 d=1.2 //m
4 w=1 //m
5 U=60*1000/3600 //m/s
6 nu=1.5e-5 //m^2/s
7 Cd=0.4
8 rho=1.22 //kg/m^3
9 // calculations
10 Rn=U*d/nu
11 A=d*w
12 Fd= Cd*0.5*rho*U^2 *A
13 M= 0.5*Fd
14 // results
15 printf(" Bending moment = %.2 f h^2 N m",M)
```

---

Scilab code Exa 11.2 Example 2

```

1  clc
2  // Initialization of variables
3  d=0.006 //m
4  U=0.01 //m/s
5  gaml=8000 //N/m^3
6  gams=7.9*10^3 *9.81
7  mu=13.9
8  // calculations
9  mu= d^2 /18 *(gams - gaml)/U
10 RN= U*d*(gaml/9.81) /mu
11 // results
12 printf("Viscosity of oil = %.1f Ns /m^2",mu)

```

---

### Scilab code Exa 11.3 Example 3

```

1
2  clc
3  // Initialization of variables
4  s=2.7
5  gamw=9810 //N/m^3
6  mu=0.001 //Ns/m^2
7  d=0.15*10^-3 //m
8  rho=1000 //kg/m^3
9  // calculations
10 gams=s*gamw
11 U= d^2 *(gams-gamw)/(18*mu)
12 RN= U*d*rho/mu
13 Cd = (1+ 3/16 *RN)^0.5 *(24/RN)
14 U22 = 4/3 *d*(gams-gamw) /(Cd*rho)
15 U2=sqrt(U22)
16 // results
17 printf("Settling velocity of sand in case 1 = %.2f
    m/s",U)
18 printf("\n Settling velocity of sand in case 2 = %.4
    f m/s",U2)

```

```
19 //The answer is a bit different due to rounding off
    error.
```

---

#### Scilab code Exa 11.4 Example 4

```
1 clc
2 //Initialization of variables
3 A=2 //m^2
4 U=100*1000/3600 //m/s
5 Cd=0.32
6 rho=1.24
7 //calculations
8 Fd= Cd*0.5*rho*U^2 *A
9 P= Fd*U
10 //results
11 printf("Power required = %.1f kW",P/1000)
```

---

#### Scilab code Exa 11.5 Example 5

```
1 clc
2 //Initialization of variables
3 ratio=0.15
4 //calculations
5 VU= (1/(1-ratio))^(1/3)
6 percent= (VU-1)*100
7 //results
8 printf("percent increase in speed = %.1f ",percent)
```

---

#### Scilab code Exa 11.6 Example 6

```
1 clc
2 //Initialization of variables
3 U=50*1000/3600 //m/s
4 cd1=0.34
5 cd2=1.33
6 //calculations
7 disp("On solving for both convex and concave
      surfaces ,")
8 w=18.26 //m/s
9 N=w/(2*%pi) *60
10 //results
11 printf("rotational speed = %.1f rpm",N)
```

---