Scilab Textbook Companion for Introduction To Fluid Mechanics by R. W. Fox And A. T. McDonald[1]

Created by
Eswar Prasad
B.Tech
Mechanical Engineering
National Institute of Technology
College Teacher
Prof. Shivraj Deshmukh
Cross-Checked by
Prof. Shivraj Deshmukh

December 5, 2013

Book Description

Title: Introduction To Fluid Mechanics

Author: R. W. Fox And A. T. McDonald

Publisher: Wiley India, New Delhi

Edition: 5

Year: 2001

ISBN: 9971-51-355-2

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

Chapter 1

Introduction

check Appendix AP 79 for dependency:

```
1_01.sci
```

Scilab code Exa 1.01 Head addition

```scilab
1 //Heat addition//
2 filename=pathname+filesep()+'1.01-data.sci'
3 exec(filename)
4 //Heat added during the process(in kJ):
5 Q12=m*cp*(T2-T1)
6 printf("\n\nRESULTS\n\n")
7 printf("\n\nHeat added during the process: %f kJ\n\n
    ",Q12/1000)
```

check Appendix AP 78 for dependency:

```
1_02.sci
```

Scilab code Exa 1.02 Head addition

```
1  //speed and actual speed//
2  pathname=get_absolute_file_path('1.02.sce')
3  filename=pathname+filesep()+'1.02-data.sci'
4  exec(filename)
5  //Speed at which the ball hits the ground(in m/sec):
6  V=sqrt(m*g/k*(1-%e^(2*k/m*(-y0))))
7  //Terminal speed(in m/sec):
8  Vt=sqrt(m*g/k)
9  //Ratio of actual speed to the terminal speed:
10 r=V/Vt;
11 printf("\n\nRESULTS\n\n")
12 printf("\n\nSpeed at which the ball hits he ground:
      %f m/sec\n\n",V)
13 printf("\n\nRatio of actual speed to the terminal
      speed: %f\n\n",r)
```

Chapter 2

Fundamental Concepts

check Appendix AP 77 for dependency:

    2_02.sci

Scilab code Exa 2.02 Viscosity and stress

```
1  //Viscosity and stress//
2  pathname=get_absolute_file_path('2.02.sce')
3  filename=pathname+filesep()+'2.02-data.sci'
4  exec(filename)
5
6  //Viscosity in units of lbf-s/ft^2:
7  u1=u/100/454/32.2*30.5
8  //Kinematic viscosity (in m/sec^2):
9  v=u1/SG/d*(0.305)^2
10 //Shear stress on the upper plate(lbf/ft^2):
11 tu=u1*U/D*1000
12 //Shear stress on the lower plate(in Pa)
13 tl=tu*4.45/0.305^2
14 printf("\n\nRESULTS\n\n")
15 printf("\n\nViscosity in units of lbf-s/ft^2: %1.8f
        $lbf-s/ft^2\n\n",u1)
```

```
16  printf("\n\nKinematic viscosity: %1.8f m/sec^2\n\n",
        v)
17  printf("\n\nShear stres on the upeer plate: %f lbf/
        ft^2\n\n",tu)
18  printf("\n\nSear stress on the lower plate: %f Pa\n\
        n",tl)
```

Chapter 3

Fluid Statics

check Appendix for dependency:

```
3_01.sci
```

Scilab code Exa 3.01 liquid level

```
1  //liquid level//
2  pathname=get_absolute_file_path('3.01.sce')
3  filename=pathname+filesep()+'3.01-data.sci'
4  exec(filename)
5  //Tube diameter(in mm):
6  D=1:25;
7  D1=D/1000
8  [m n]=size(D1)
9  for i=1:n
10 //Change in liquid level for water(in mm):
11 dhw(i)=4*STw*cosd(thetaw)/dw/g/D1(i);
12 //Change in liquid level for mercury(in mm):
13 dhm(i)=4*STm*cosd(thetam)/dm/g/D1(i);
14 end;
15
16 //Plotting tube daimeter and water level:
17 plot(D1*1000,dhw,'-o')
```

```
18  // Plotting  tube  daimeter  and  mercury  level :
19  plot ( D1 *1000 , dhm , '−∗ ')
20  legend ([ ' Water ' ; ' Mercury ' ]) ;
21  xtitle ( ' Liquid  level  vs  Tube  diameter ' , ' Liquid  level
        ( in  mm) ' , ' Tube  diameter ( in  mm) ')
```

check Appendix AP 75 for dependency:

```
3_03.sci
```

Scilab code Exa 3.03 pressure difference

```
1  // pressure  difference //
2  pathname = get_absolute_file_path ( ' 3.03. sce ')
3  filename = pathname + filesep () + ' 3.03 − data . sci '
4  exec ( filename )
5  // Pressure  difference ( in  lbf / in ^2 ) :
6  dp = g * d *( - d1 + SGm * d2 - SGo * d3 + SGm * d4 + d5 ) /12/144
7  printf ( " \n\nRESULTS\n\n ")
8  printf ( " \n\nPressure  difference  between  A  and  B:  %f
        lbf / in ^2\n\n " , dp )
```

check Appendix AP 74 for dependency:

```
3_04.sci
```

Scilab code Exa 3.04 temperature and pressure

```
1  // temperature  and  pressure //
2  pathname = get_absolute_file_path ( ' 3.04. sce ')
3  filename = pathname + filesep () + ' 3.04 − data . sci '
4  exec ( filename )
5  // Assuming  temperature  varies  linearly  with  altitude
        :
```

```
 6  //Temperature gradient(in F/ft):
 7  m=(T1-T2)/(z2-z1)
 8  //Value of g/(m*R):
 9  v=g/m/R/32.2
10  //Pressure at Vail Pass(in inches of Hg):
11  p12=p1*((T2+460)/(T1+460))^v
12  //Percentage change in density:
13  pc1=(p12/p1*(T1+460)/(T2+460)-1)*100
14  //Assuming density is constant:
15  //Pressure at Vail Pass(in inches of Hg):
16  p22=p1*(1-(g*(z2-z1)/(R*32.2)/(T1+460)))
17  //Percentage change in density:
18  pc2=0;
19  //Assuming temperature is constant:
20  //Pressure at Vail Pass(in inches of Hg):
21  p32=p1*%e^(-g*(z2-z1)/(R*32.2)/(T2+460))
22  //Percentage change in density:
23  pc3=(p32/p1*(T1+460)/(T1+460)-1)*100
24  //For an adiabatic atmosphere:
25  p42=p1*((62+460)/(80+460))^(k/(k-1))
26  //Percentage change in density:
27  pc4=(p42/p1*(T1+460)/(T2+460)-1)*100
28  printf("\n\nRESULTS\n\n")
29  printf("\n\n1) If temperature varies linearly with
         altitude\n\n")
30  printf("\n\n\tAtmospheric pressure at Vail Pass: %f
        inches of Hg\n\n",p12)
31  printf("\n\n\tPercentage change in density wrt
        Denver: %f percent\n\n",pc1)
32  printf("\n\n2) If density is constant\n\n")
33  printf("\n\n\tAtmospheric pressure at Vail Pass: %f
        inches of Hg\n\n",p22)
34  printf("\n\n\tPercentage change in density wrt
        Denver: %f percent\n\n",pc2)
35  printf("\n\n3) If temperature is constant\n\n")
36  printf("\n\n\tAtmospheric pressure at Vail Pass: %f
        inches of Hg\n\n",p32)
```

```
37  printf("\n\n\tPercentage  change  in  density  wrt
        Denver :  %f  percent\n\n",pc3)
38  printf("\n\n4)  For  an  adiabatic  atmosphere\n\n")
39  printf("\n\n\tAtmospheric  pressure  at  Vail  Pass :  %f
        inches  of  Hg\n\n",p42)
40  printf("\n\n\tPercentage  change  in  density  wrt
        Denver :  %f  percent\n\n",pc4)
```

check Appendix for dependency:

    3_05.sci

Scilab code Exa 3.05 force and pressure

```
1  //force  and  pressure//
2  pathname=get_absolute_file_path('3.05.sce')
3  filename=pathname+filesep()+'3.05-data.sci'
4  exec(filename)
5  //Net  force  on  the  gate(in  kN):
6  Fr=d*g*w*(D*L+L^2/2*sind(theta))
7  //Centre  of  pressure :
8  //Calculation  for  y  coordinate :
9      yc=D/sind(theta)+L/2
10     //Area(in  m^2):
11     A=L*w
12     //Moment  of  inertia  of  rectangular  gate(in  m^4):
13     Ixx=w*L^3/12
14     //y  coordinate(in  m):
15     y=yc+Ixx/A/yc
16  //Calculation  for  x  coordinate :
17     Ixy=0
18     xc=w/2
19     //x  coordinate(in  m):
20     x=xc+Ixy/A/xc
21  printf("\n\nRESULTS\n\n")
```

```
22  printf("\n\nNet force on the gate: %f kN\n\n",Fr
        /1000)
23  printf("\n\nCoordinate of centre of pressure:(%0.1f,
        %0.1f)\n\n",x,y)
```

check Appendix AP 72 for dependency:

```
3_06.sci
```

Scilab code Exa 3.06 force

```
1  //force//
2  pathname=get_absolute_file_path('3.06.sce')
3  filename=pathname+filesep()+'3.06-data.sci'
4  exec(filename)
5  //Force required to keep the door shut(in lbf):
6  function y=f(z),y=b/L*p0*z+d*b/L*(L*z-z^2),
        endfunction
7  Ft=intg(0,L,f)
8  printf("\n\nRESULTS\n\n")
9  printf("\n\nForce requiredto kep the door shut: %.1f
        lbf\n\n",Ft)
```

check Appendix AP 71 for dependency:

```
3_07.sci
```

Scilab code Exa 3.07 force at equilibrium

```
1  //force at equilibrium//
2  pathname=get_absolute_file_path('3.07.sce')
3  filename=pathname+filesep()+'3.07-data.sci'
4  exec(filename)
5  //Horizontal component of resultant force(in kN):
```

```
6  Frh=0.5*d*g*w*D^2
7  //Line of action of Frh(in m):
8  y1=0.5*D+w*D^3/12/(0.5*D)/(w*D)
9  //Vertical component of resultant force(in kN):
10 function y=q(x), y=d*g*w*(D-sqrt(a*x)),endfunction
11 Frv=intg(0,D^2/a,q)
12 //Line of acion of Frv(in m):
13 function k=f(x), k=d*g*w/Frv*x*(D-sqrt(a*x)),
       endfunction
14 xa=intg(0,D^2/a,f)
15 //Force required to keep the gate in equilibrium(in
       kN):
16 Fa=1/l*(xa*Frv+(D-y1)*Frh)
17 printf("\n\nRESULTS\n\n")
18 printf("\n\nForce required to keep the gate at
       equilibrium: %f kN\n\n",Fa/1000)
```

Chapter 4

Basic Equations in Integral form for a Control Volume

check Appendix AP 70 for dependency:

    4_01.sci

Scilab code Exa 4.01 Velocity

```
1  //Velocity//
2  pathname=get_absolute_file_path('4.01.sce')
3  filename=pathname+filesep()+'4.01-data.sci'
4  exec(filename)
5  //If I=integral of(pV.dA):
6  //For system: Ics=IA1+IA2+IA3+IA4.
7  //For area 1
8  IA1=-d*V1*A1
9  //For area 3: IA2=d*V3*A3=m3
10 IA3=m3
11 //For area 4: IA4=-d*V4*A4=-d*Q4
12 IA4=-d*Q4
13 //For area 2:
14 IA2=-IA1-IA3-IA4
15 //Velocity at section 2(in ft/sec):
```

```
16  V2 = IA2 / d / A2
17  //V2 is in the negative y direction
18  printf("\n\nRESULTS\n\n")
19  printf("\n\nVelocity at section 2: -%.0fj ft/sec\n\n
        ",V2)
```

check Appendix AP 69 for dependency:

```
4_02.sci
```

Scilab code Exa 4.02 Mass flow

```
1   //Mass flow//
2   pathname=get_absolute_file_path('4.02.sce')
3   filename=pathname+filesep()+'4.02-data.sci'
4   exec(filename)
5   //If I=integral of(pV.dA):
6   //For system: ICS=Iab+Ibc+Icd+Ida
7   //But ICS=0
8
9   //For Aab:
10  function p=f(y),p=-d*U*w*y^0,endfunction
11  IAab=intg(0,t,f)
12
13  //For Acd:
14  function q=g(y),q=d*U*w*(2*y/t-(y/t)^2),endfunction
15  IAcd=intg(0,t,g)
16
17  //Mass flow rate across surface bc(in kg/sec):
18  mbc=(-IAab-IAcd)/1000
19  printf("\n\nRESULTS\n\n")
20  printf("\n\nMass flow rate across surface bc: %.4f
        kg/sec\n\n",mbc)
```

check Appendix AP 68 for dependency:

```
4_03.sci
```

20

Scilab code Exa 4.03 density

```
1  //density//
2  pathname=get_absolute_file_path('4.03.sce')
3  filename=pathname+filesep()+'4.03-data.sci'
4  exec(filename)
5  //Rate of change of air density in tank(in (kg/m^3)/
      s):
6  r=-d*v*A/V/10^6
7  printf("\n\nRESULTS\n\n")
8  printf("\n\nRate of change of air density in tank: %
      .3f kg/m^3\n\n",r)
9  printf("\n\nThe density decreases as is indicated by
       the negative sign\n\n")
```

check Appendix for dependency:

```
4_04.sci
```

Scilab code Exa 4.04 Horizontal force

```
1  //Horizontal force//
2  pathname=get_absolute_file_path('4.04.sce')
3  filename=pathname+filesep()+'4.04-data.sci'
4  exec(filename)
5  //1) Control Volume selected so that area of left
      surface is equal to the area of the right surface
6  u1=15;
7  //Force of support on control volum(in kN):
8  function y=f(A),y=-u1*d*V,endfunction
9  Rx1=intg(0,0.01,f)
10 //Horizontal force on support(in kN):
```

```
11  Kx=-Rx1
12  //2) Control volumes are selected do that the area
       of the left and right surfaces are equial to the
       area of the plate
13
14  function z=g(A),z=-u1*d*V,endfunction
15  Fsx=intg(0,0.01,g)
16  //Net force on plate:Fx=0=-Bx-pa*Ap+Rx
17  //                    Rx=pa*Ap+Bx
18  //From the above, it is obtained that:
19  Rx2=-2.25
20  //Horizontal force on support(in kN):
21  Kx2=-Rx2
22  printf("\n\nRESULTS\n\n")
23  printf("\n\nHorizontal force on support: %.3f kN\n\n
       ",Kx/1000)
```

check Appendix for dependency:

```
4_05.sci
```

Scilab code Exa 4.05 Scale

```
1  //Scale//
2  pathname=get_absolute_file_path('4.05.sce')
3  filename=pathname+filesep()+'4.05-data.sci'
4  exec(filename)
5  //Weight of water in the tank(in lbf):
6  d1=62.4;
7  WH2O=d1*A*h
8  v=-5;
9  //Total body force in negative y direction(lbf):
10 function y=f(A),y=-v*d2*V1,endfunction
11 F=intg(0,A1,f)
12 //Force of scale on control volume(in kN):
13 Ry=W+WH2O-F
```

```
14  printf("\n\nRESULTS\n\n")
15  printf("\n\nScale Reading: %.3f lbf\n\n",Ry)
```

check Appendix AP 65 for dependency:

```
4_06.sci
```

Scilab code Exa 4.06 force exerted per unt

```
1  //force exerted per unt//
2  pathname=get_absolute_file_path('4.06.sce')
3  filename=pathname+filesep()+'4.06-data.sci'
4  exec(filename)
5  //X-component of reaction force per unit width of
       the gate(in N/m):
6  Rxw=(d*(V2^2*D2-V1^2*D1))-(d*g/2*(D1^2-D2^2))
7  //Horizontal force exerted per unt width on the gate
       (in N/m):
8  Kxw=-Rxw
9  printf("\n\nRESULTS\n\n")
10 printf("\n\nHorizontal force exerted per unt width
       on the gate: %.3f kN/m\n\n",Kxw/1000)
```

check Appendix AP 64 for dependency:

```
4_07.sci
```

Scilab code Exa 4.07 Force to hold

```
1  //Force to hold//
2  pathname=get_absolute_file_path('4.07.sce')
3  filename=pathname+filesep()+'4.07-data.sci'
4  exec(filename)
5  //Velocity at section 1(in m/sec):
```

```
6  V1 = V2 * A2 / A1
7  // Gauge pressure ( in kPa ) :
8  p1g = p1 - patm
9  u1 = V1 ; u2 = - V2 ;
10 // Reaction force component in the x direction ( in N ) :
11 Rx = - p1g * A1 - u1 * d * V1 * A1
12 // Reaction force component in the y direction ( in N ) :
13 Ry = u2 * d * V2 * A2
14 printf ( " \n\nRESULTS\n\n " )
15 printf ( " \n\nForce to hold elbow acting to the left :
      %.3 f kN\n\n " , Rx /1000)
16 printf ( " \n\nForce to hold elbow acting downwards : %
      .3 f N\n\n " , Ry )
```

check Appendix AP 63 for dependency:

    4_08.sci

Scilab code Exa 4.08 Tension

```
1 // Tension //
2 filename = pathname + filesep () + '4.08 - data . sci '
3 exec ( filename )
4 // Tension required to pull the belt ( in lbf ) :
5 T = Vbelt * m /32.2
6 printf ( " \n\nRESULTS\n\n " )
7 printf ( " \n\nTension required to pull the belt : %.3 f
      lbf \n\n " , T )
```

check Appendix AP 62 for dependency:

    4_09.sci

Scilab code Exa 4.09 pressure required

```
1  //pressure required//
2  pathname=get_absolute_file_path('4.09.sce')
3  filename=pathname+filesep()+'4.09-data.sci'
4  exec(filename)
5  //Minimum gauge pressure required(in lbf/in^2):
6  pg=8/%pi^2*d/D1^4*Q^2*((D1/D2)^4-1)*144
7  printf("\n\nRESULTS\n\n")
8  printf("Minimum gauge pressure required: %.3f lbf/in
       ^2",pg)
```

check Appendix AP 61 for dependency:

```
4_10.sci
```

Scilab code Exa 4.10 Net force

```
1  //Net force//
2  pathname=get_absolute_file_path('4.10.sce')
3  filename=pathname+filesep()+'4.10-data.sci'
4  exec(filename)
5  u1=V-U
6  u2=(V-U)*cosd(theta)
7  v2=(V-U)*sind(theta)
8  V1=V-U
9  V2=V1
10 //X component of moment equation(in N):
11 function y=f(A),y=u1*-(d*V1),endfunction
12 function z=g(A),z=u2*d*V2,endfunction
13 Rx=intg(0,A,f)+intg(0,A,g)
14
15 //Y component of moment equation(in N):
16 function a=h(A),a=v2*d*V1,endfunction
17 Ry=intg(0,A,h) //This is after neglecting weight of
       vane and the water.
18 printf("\n\nRESULTS\n\n")
19 printf("\n\nNet force on the vane: %.3f i+%.2f j kN\
       n\n",Rx/1000,Ry/1000)
```

25

check Appendix AP 60 for dependency:

```
4_11.sci
```

Scilab code Exa 4.11 PLOTTING

```
1  //PLOTTING//
2  pathname=get_absolute_file_path('4.11.sce')
3  filename=pathname+filesep()+'4.11-data.sci'
4  exec(filename)
5  //Evaluating the value of Vb:
6  Vb=V*(1-cosd(theta))*d*A/M
7  //Value of U/V for various values of t
8  t=0:20;
9  [m n]=size(t)
10 for i=1:n
11    U_V(i)=Vb*t(i)/(1+Vb*t(i));
12 end
13
14 //Plotting U/V vs t:
15 plot(t,U_V)
16 xtitle('U/V vs t','t (in sec)','U/V')
```

check Appendix AP 59 for dependency:

```
4_12.sci
```

Scilab code Exa 4.12 Velocity of rocket

```
1  //Velocity of rocket//
2  pathname=get_absolute_file_path('4.12.sce')
3  filename=pathname+filesep()+'4.12-data.sci'
4  exec(filename)
```

```
5  //Acceleration of rocket at t=0(in m/sec^2):
6  Ve*me/M0-g
7  //Velocity of rocket at t=10 (in m/sec):
8  function y=f(t),y=Ve*me/(M0-me*t)-g,endfunction
9  Vcv=intg(0,t,f)
10 printf("\n\nRESULTS\n\n")
11 printf("\n\nVelocity of rocket at t=10: %.1f m/sec\n
        \n",Vcv)
```

check Appendix AP 58 for dependency:

```
4_14.sci
```

Scilab code Exa 4.14 Relative speed and friction

```
1  //Relative speed and friction//
2  pathname=get_absolute_file_path('4.14.sce')
3  filename=pathname+filesep()+'4.14-data.sci'
4  exec(filename)
5  //Area of jet(in mm^2):
6  Ajet=%pi/4*D^2
7  //Jet speed relative to the nozzle(in m/sec):
8  Vrel=Q/2/Ajet*10^6/60/1000
9  //Value of w*R in m/sec:
10 wR=w*R*2*%pi/60/1000
11 //Friction torque at pivot(in N-m):
12 Tf=R*(Vrel*cosd(alpha)-wR)*d*Q/1000/60/1000
13 printf("\n\nRESULTS\n\n")
14 printf("\n\nJet speed relative to each nozzle: %.2f
        m/sec\n\n",Vrel)
15 printf("\n\nFriction torque at pivot: %.5f N-m\n\n",
        Tf)
```

check Appendix AP 57 for dependency:

```
4_16.sci
```

Scilab code Exa 4.16 Rate of heat

```
1  //Rate of heat//
2  pathname=get_absolute_file_path('4.16.sce')
3  filename=pathname+filesep()+'4.16-data.sci'
4  exec(filename)
5  //Velocity at exit(in ft/sec):
6  V2=m*R*(T2+460)/A2/p2/144
7  //As power input is to CV, Ws=-600
8  //Rate of heat transfer(in Btu/sec):
9  Q=Ws*550/778+m*cp*(T2-T1)+m*V2^2/2/32.2/778
10 printf("\n\nRESULTS\n\n")
11 printf("\n\nRate of heat transfer: %.3f Btu/sec\n\n"
       ,Q)
```

check Appendix AP 56 for dependency:

```
4_17.sci
```

Scilab code Exa 4.17 Mass flow rate

```
1  //Mass flow rate//
2  pathname=get_absolute_file_path('4.17.sce')
3  filename=pathname+filesep()+'4.17-data.sci'
4  exec(filename)
5  //Density of tank(in kg/m^3):
6  d=(p1+patm)/R/T
7  //Mass flow rate of air into the tank(in kg/sec):
8  m=d*V*cv*r/R/T*1000
9  printf("\n\nRESULTS\n\n")
10 printf("\n\nMass flow rate of air into the tank: %.3
       f g/sec\n\n",m)
```

Chapter 5

Introducton to Differential Analysis of Fluid Motion

check Appendix for dependency:

```
5_02.sci
```

Scilab code Exa 5.02 Rate of change

```scilab
1 //Rate of change//
2 pathname=get_absolute_file_path('5.02.sce')
3 filename=pathname+filesep()+'5.02-data.sci'
4 exec(filename)
5 //Rate of change of density with time(in kg/m^3-s):
6 r=-d*V/L
7 printf("\n\nRESULTS\n\n")
8 printf("\n\nRate of change of density with time: %.1
     f kg/m^3-s\n\n",r)
```

check Appendix for dependency:

```
5_07.sci
```

Scilab code Exa 5.07 angular and rotation

```
1  //angular and rotation//
2  pathname=get_absolute_file_path('5.07.sce')
3  filename=pathname+filesep()+'5.07-data.sci'
4  exec(filename)
5  //At point b, u=3 mm/sec
6  u=3;
7  //Displacemet of b(in mm):
8  xb=u*t
9  //Rate of angular deformation(in s^-1):
10 def=U/h
11 //Rate of rotation(in s^-1):
12 rot=-0.5*U/h
13 printf("\n\nRSULTS\n\n")
14 printf("\n\nRate of angular deformation: %.1f /sec\n
      \n",def)
15 printf("\n\nRate of rotation: %.1f /sec\n\n",rot)
```

check Appendix AP 53 for dependency:

```
5_08.sci
```

Scilab code Exa 5.08 Rates and area

```
1  //Rates and area//
2  pathname=get_absolute_file_path('5.08.sce')
3  filename=pathname+filesep()+'5.08-data.sci'
4  exec(filename)
5  //Value of T:
6  T=log(3/2)/A
7  x0=1:2;
8  y0=1:2;
9  for i=1:2
10    for j=1:2
11    //For X coordinate:
```

```
12    X(i)(j)=x0(i)*%e^(A*T)
13    //For Y coordinate:
14    Y(i)(j)=y0(j)*%e^(-A*T)
15    end
16  end
17  plot(X,Y)
18  //Rates of linear deformation in X direction:
19  Ax=0.3;
20  //Rate of linear deformation in the y direction:
21  Ay=-0.3;
22  //Rate of volume dilation(s^-1):
23  v=A-A
24  //Area of abcd:
25  A1=1;
26  //Area of a'b'c'd':
27  A2=(3-3/2)*(4/3-2/3)
28  printf("\n\nRESULTS\n\n")
29  printf("\n\nRates of linear deformation in X and Y
          direction: %.1f /s , %.1f /s\n\n",Ax,Ay)
30  printf("\n\nRate of volume dilation: %.0f /sec\n\n",
          v)
31  printf("\n\nArea of abcd and a,b,c,d:%.1f m^2, %.1f
          m^\n\n",A1,A2)
```

check Appendix AP 52 for dependency:

```
5_09.sci
```

Scilab code Exa 5.09 Volume flow rate

```
1  //Volume flow rate//
2  pathname=get_absolute_file_path('5.09.sce')
3  filename=pathname+filesep()+'5.09-data.sci'
4  exec(filename)
5  //Volume flow rate(in m^3/sec):
6  Q=d*g*sind(theta)*b*(h/1000)^3*1000/u/3
```

```
7  printf("RESULTS")
8  printf("\n\nVolume flow rate: %.4f m^3/sec\n\n",Q)
```

Chapter 6

Incompressible Inviscid Flow

check Appendix AP 51 for dependency:

```
6_01.sci
```

Scilab code Exa 6.01 Volume flow rate

```scilab
1  //Volume flow rate//
2  pathname=get_absolute_file_path('06.01.sce')
3  filename=pathname+filesep()+'06.01-data.sci'
4  exec(filename)
5  //Velocity of flow(in m/sec):
6  V=sqrt(dw/log((r+w)/r)*g/da*p/1000)
7  //Volume flow rate(in m^3/sec):
8  Q=V*(d*w)
9  printf("\n\nRESULTS\n\n")
10 printf("\n\nVolume flow rate: %.3f m^3/sec\n\n",Q)
```

check Appendix AP 50 for dependency:

```
6_02.sci
```

33

Scilab code Exa 6.02 Velocity of flow

```
1 //Velocity of flow//
2 pathname=get_absolute_file_path('06.02.sce')
3 filename=pathname+filesep()+'06.02-data.sci'
4 exec(filename)
5 //Velocity of flow(in m/sec):
6 V=sqrt(2*dw*g*p/1000*SG/da)
7 printf("\n\nRESULTS\n\n")
8 printf("\n\nVelocity of flow: %.3f m/sec\n\n",V)
```

check Appendix AP 49 for dependency:

```
6_03.sci
```

Scilab code Exa 6.03 prssure required

```
1 //prssure required//
2 filename=pathname+filesep()+'06.03-data.sci'
3 exec(filename)
4 //Velocity of flwat the inlet(in m/sec):
5 V1=Ae/Ai*V2
6 //Gauge pressure required at the inlet(in kPa):
7 p=0.5*da*(V2^2-V1^2)
8 printf("\n\nRESULTS\n\n")
9 printf("\n\nGauge prssure required at the inlet: %.3
    f kPa\n\n",p/1000)
```

check Appendix AP 48 for dependency:

```
6_04.sci
```

Scilab code Exa 6.04 Speed and pressure

34

```
1 //Speed and pressure//
2 pathname=get_absolute_file_path('06.04.sce')
3 filename=pathname+filesep()+'06.04-data.sci'
4 exec(filename)
5 //Speed of water at exit(in m/sec):
6 V2=sqrt(2*g*z)
7 //Pressure at point A in the flow(kPa):
8 pA=p1+d*g*(0-l)-0.5*d*V2^2
9 printf("\n\nRESULTS\n\n")
10 printf("\n\nSpeed of water at exit: %.3f m/sec\n\n",
       V2)
11 printf("\n\nPressure at point A in the flow: %3f kPa
       \n\n",pA/1000)
```

check Appendix AP 47 for dependency:

```
6_05.sci
```

Scilab code Exa 6.05 flow

```
1 //flow//
2 pathname=get_absolute_file_path('06.05.sce')
3 filename=pathname+filesep()+'06.05-data.sci'
4 exec(filename)
5 //Velocity of flow at the exit(in ft/sec):
6 V2=sqrt(2*g*(Du-Dd/12))
7 //Volume flow rate/width(ft^2/sec):
8 Q=V2*Dd/12
9 printf("\n\nRESULTS\n\n")
10 printf("\n\nVelocity of flow at the exit: %.3f ft/
       sec\n\n",V2)
11 printf("\n\nVolume flow rate/width: %.3f ft^2/sec\n\
       n",Q)
```

check Appendix AP 46 for dependency:

```
6_06.sci
```

Scilab code Exa 6.06 pressure

```
1  //pressure//
2  pathname=get_absolute_file_path('06.06.sce')
3  filename=pathname+filesep()+'06.06-data.sci'
4  exec(filename)
5  //Pressure of air at 1000 m(in N/m^2):
6  p=P1*pa
7  //Density of air at 1000m(in kg/m^3):
8  d=D1*da
9  //Stagnation pressure at A(in kPa):
10 p0A=p+0.5*d*(V*1000/3600)^2
11 //Static pressure at B(in kPa):
12 pB=p+d/2*((V*1000/3600)^2-Vb^2)
13 printf("\n\nRESULTS\n\n")
14 printf("\n\nStagnation pressure at A: %.3f kPa\n\n",
      p0A/1000)
15 printf("\n\nStatic pressure at B: %.3f kPa\n\n",pB
      /1000)
```

check Appendix AP 45 for dependency:

```
6_08.sci
```

Scilab code Exa 6.08 temperature

```
1  //temperature//
2  pathname=get_absolute_file_path('06.08.sce')
3  filename=pathname+filesep()+'06.08-data.sci'
4  exec(filename)
5  //Velocity of flow at exit(in ft/sec):
6  V4=sqrt(2*g*(z3-0))
```

```
7  //Mass flow rate of water(in slug/sec):
8  m=d*V4*A4/144
9  //Rise in temperature between points 1 and 2(in R):
10 T=Q*3413/3600/m/32.2
11 printf("\n\nRESULTS\n\n")
12 printf("\n\nRise in temperature between points 1 and
        2: %.3f R\n\n",T)
```

check Appendix AP 44 for dependency:

```
6_09.sci
```

Scilab code Exa 6.09 Streamline flow

```
1  //Streamline flow//
2  pathname=get_absolute_file_path('06.09.sce')
3  filename=pathname+filesep()+'06.09-data.sci'
4  exec(filename)
5  t=0:5
6  //Value of sqrt(2gh):
7  x=sqrt(2*g*h)
8  //Value of 1/2L*sqrt(2gh):
9  y=1/2/L*x
10 [m n]=size(t)
11 i=1:n;
12 //Velocity(in m/sec):
13 V2=x*tanh(y*t(i))
14 plot(t,V2);
15 xtitle('Streamline flow from 1 to 2','Time(in s)','
     V2(in m/sec)')
```

Chapter 7

Dimensional Analysis and Simlitude

check Appendix AP 43 for dependency:

```
7_04.sci
```

Scilab code Exa 7.04 speed and force

```
1  //speed and force//
2  pathname=get_absolute_file_path('7.04.sce')
3  filename=pathname+filesep()+'7.04-data.sci'
4  exec(filename)
5  //Velocity of prototype in ft/sec
6  Vp1=Vp*6080/3600
7  //Reynolds number of prototype:
8  Rep=Vp1*Dp/vp
9  //Rep=Rem
10 //Therefore:
11 Rem=Rep;
12 //Velocity of air for wind tunnel(in ft/sec):
13 Vm=Rem*vm/(Dm/12)
14 //Drag force on prototype(in lbf):
15 Fp=Fm*(dp/dm)*(Vp1/Vm)^2*(Dp/(Dm/12))^2
```

```
16  printf("\n\nRESULTS\n\n")
17  printf("\n\nTest speed in air: %.3f ft/sec\n\n",Vm)
18  printf("\n\nDrag force on prototype: %.3f lbf\n\n",
        Fp)
```

check Appendix AP 42 for dependency:

7_05.sci

Scilab code Exa 7.05 speed force and power

```
1  //speed force and power//
2  pathname=get_absolute_file_path('7.05.sce')
3  filename=pathname+filesep()+'7.05-data.sci'
4  exec(filename)
5  //Width of the model(in m):
6  wm=S*wp*0.3048
7  //Area of model (in m^2):
8  Am=S^2*Ap*0.305^2
9  [m n]=size(V)
10 i=1:n
11 //Aerodynamic drag coefficient():
12 Cd=2.*Fd(i)/d./(V(i))^2/0.0305
13 //Reynolds number:
14 Re=V(i)*wm/v
15 plot(Re,Cd);
16 a=gca()
17 a.data_bounds=[100000,0.4;500000,0.6]
18 xtitle('Aerodynamic drag coefficient vs drag force',
        'Reynolds number','Model Drag Coeff.')
19 //It is seen that drag coefficient becomes constant
        at CD=0.46above Re=4*10^5 at which speed of air
        is 40m/s
20 CDc=0.46;
21 Va=40;
22 //Drag force (in N):
```

```
23  FDp=CDc/2*d*(Vp*5/18)^2*Ap*0.305^2
24  //Power required to pull prototype at 100 kmph(in W)
25  Pp=FDp*Vp*5/18
26  printf("\n\nRESULTS\n\n")
27  printf("\n\nSpeed above which Cd is constant: %.3f m
        /sec\n\n",Va)
28  printf("\n\nDrag Force: %.3f kN\n\n",FDp/1000)
29  printf("\n\nPower required to pull prototype at 100
        kmph: %.3f kW\n\n",Pp/1000)
```

check Appendix AP 41 for dependency:

```
7_06.sci
```

Scilab code Exa 7.06 power and speed

```
1  //power and speed//
2  pathname=get_absolute_file_path('7.06.sce')
3  filename=pathname+filesep()+'7.06-data.sci'
4  exec(filename)
5  //The same pump is used for both the conditions.
       Hence:
6  D2=D1;
7  //The same water is used for both the conditions.
       Hence:
8  d2=d1;
9  //Flow rate at condition 2(in gpm):
10  Q2=Q1*N2/N1*(D2/D1)^3
11  //Head at condition 1(in ft):
12  H1=(N1*sqrt(Q1)/Nscu1)^(4/3)
13  //Head at condition 1(in ft):
14  H2=H1*(N2/N1)^2*(D2/D1)^2
15  //Pump output power at condition 1(in hp):
16  P1=d1*g*Q1*H1/7.48/60/550
17  //Pump output power at condition 2(in hp):
18  P2=P1*(d2/d1)*(N2/N1)^3*(D2/D1)^5
```

```
19  //Required input power(in hp):
20  Pin=P2/Effp
21  //Specific speed at condition 2:
22  Nscu2=N2*sqrt(Q2)/H2^(3/4)
23  printf("\n\nRESULTS\n\n\n")
24  printf("\n\nVolume flow rate at condition 2: %.3f
        gpm\n\n\n",Q2)
25  printf("\n\nHead at condition: %.3f ft\n\n\n",H2)
26  printf("\n\nPump output power at condition: %.3f hp\
        n\n\n",P2)
27  printf("\n\nRequired input power: %.3f hp\n\n\n",Pin
        )
28  printf("\n\nSpecific speed at condition 2: %.3f\n\n\
        n",Nscu2)
```

Chapter 8

Internal Incompressible Viscous Flow

check Appendix AP 40 for dependency:

```
8_01.sci
```

Scilab code Exa 8.01 Leakage flow rate

```scilab
1  //Leakage flow rate//
2  pathname=get_absolute_file_path('8.01.sce')
3  filename=pathname+filesep()+'8.01-data.sci'
4  exec(filename)
5  //Leakage flow rate (in mm^3/sec):
6  Q=%pi/12*D*a^3*(p1-p2)*10^3/u/L
7  //Velocity of flow(in m/sec):
8  V=Q/%pi/D/a/1000
9  //Specific gravity of SAE 10W oil:
10 SG=0.92;
11 //Reynolds Number:
12 Re=SG*dw*V*a/u/1000
13 //As Re<1400, flow is laminar.
14 printf("\n\nRESULTS\n\n")
15 printf("\n\nLeakage flow rate: %.3f mm^3/sec\n\n",Q)
```

check Appendix for dependency:

```
8_02.sci
```

Scilab code Exa 8.02 Torque and power

```
1  //Torque and power//
2  pathname=get_absolute_file_path('8.02.sce')
3  filename=pathname+filesep()+'8.02-data.sci'
4  exec(filename)
5  //Shear stres (in lbf/ft^2):
6  Tyx=u*N*2*%pi/60*D/2/(a/2)
7  //Torqe(in inches-lbf):
8  T=%pi/2*Tyx*D^2*L/144
9  //Power dissipated in the bearing(in hp):
10 P=T*N/60*2*%pi/12/550
11 //Reynolds number:
12 Re=SG*p*N*2*%pi/60*1.5*a/2/u/144
13 printf("\n\nRESULTS\n\n")
14 printf("\n\nTorque: %.3f inches-lbf\n\n",T)
15 printf("\n\nPower dissipated in the bearing: %.3f hp
       \n\n",P)
```

check Appendix for dependency:

```
8_04.sci
```

Scilab code Exa 8.04 Viscosity of fluid

```
1  //Viscosity of fluid//
2  pathname=get_absolute_file_path('8.04.sce')
3  filename=pathname+filesep()+'8.04-data.sci'
4  exec(filename)
```

```
5  //Viscosity of the liquid(in N-s/m^2):
6  u=%pi/128*p*1000*D^4/Q/L/1000
7  //Velocity(in m/sec)
8  V=Q/(%pi/4*D^2)/1000
9  //Reynolds number:
10 Re=d*V*D/u/1000
11 printf("\n\nRESULTS\n\n")
12 printf("\n\nViscosity of fluid %.3f N-s/m^2\n\n",u)
```

check Appendix AP 37 for dependency:

```
8_05.sci
```

Scilab code Exa 8.05 required

```
1  //required//
2  pathname=get_absolute_file_path('8.05.sce')
3  filename=pathname+filesep()+'8.05-data.sci'
4  exec(filename)
5  //Reservoir depth required to maintain flow(in m):
6  D1=8*Q^2/(%pi)^2/D^4/g*(f*L/D+K+1)
7  //Reynolds number:
8  Re=4*d*Q/((%pi)*u*D)
9  printf("\n\nRESULTS\n\n")
10 printf("\n\nReservoir depth required to maintain
       flow: %.3f m\n\n",D1)
```

check Appendix AP 36 for dependency:

```
8_06.sci
```

Scilab code Exa 8.06 Maximum and power

44

```
1  //Maximum and power//
2  pathname=get_absolute_file_path('8.06.sce')
3  filename=pathname+filesep()+'8.06-data.sci'
4  exec(filename)
5  //Velocity of flow(in ft/sec):
6  V=Q/24/3600/(%pi/4*(D/12)^2)*42/7.48
7  //Maximum spacing(in ft):
8  L=2/f*D/12*(p2-p1)/(SG*d)/V^2*144
9  //Power needed at each pump(in hp):
10 Win=1/Effp*V*%pi/4*(D/12)^2*(p2-p1)/550*144
11 printf("\n\nRESULTS\n\n")
12 printf("\n\nMaximum spacing: %.3f feet\n\n",L)
13 printf("\n\nPower needed at each pump: %.3f hp\n\n",
       Win)
```

check Appendix AP 35 for dependency:

8_07.sci

Scilab code Exa 8.07 Volume low

```
1  //Volume low//
2  pathname=get_absolute_file_path('8.07.sce')
3  filename=pathname+filesep()+'8.07-data.sci'
4  exec(filename)
5  //Velocity(in ft/sec):
6  V2=sqrt(2*g*l/(f*((L+l)/D*12+8)+1))
7  //Volume flow rate(in gpm):
8  Q=V2*%pi*(D/12)^2/4*7.48*60
9  printf("\n\nRESULTS\n\n")
10 printf("\n\nVolume low rate: %.3f\n\n",Q)
```

check Appendix AP 34 for dependency:

8_08.sci

Scilab code Exa 8.08 Minimum diameter

```
1  //Minimum diameter//
2  pathname=get_absolute_file_path('8.08.sce')
3  filename=pathname+filesep()+'8.08-data.sci'
4  exec(filename)
5  //Value of dPmax(in psi):
6  dPmax=p1-p2
7  //Q in cubic feet/sec:
8  Q1=1500/60/7.48;
9  //Initially assume diameter to be 4inches:
10 D=4;
11 //Reynolds number:
12 Re=4*Q1/%pi/v/D*12
13 //For this value,
14 f=0.012;
15  dP=8*f*L*p*Q1^2/(%pi)^2/D^5*1728;
16 while(dP>dPmax)
17     dP=8*f*L*p*Q1^2/(%pi)^2/D^5*1728;
18     if(dP<dPmax)
19     break
20     else
21     D=D+1;
22   end
23 end
24 printf("\n\nRESULTS\n\n")
25 printf("Minimum diameter that can be used:%.1f
     inches\n\n",D)
```

check Appendix for dependency:

```
8_09.sci
```

Scilab code Exa 8.09 Loss Coefficient

```
1  //Loss Coefficient//
```

```
2  pathname=get_absolute_file_path('8.09.sce')
3  filename=pathname+filesep()+'8.09-data.sci'
4  exec(filename)
5  //Average velocity (in ft/s):
6  V2=4/%pi*Q/D^2*144
7  //Reynolds number:
8  Re=V2*D/v/12
9  //For this value,
10 f=0.013;
11 //Power law exponent:
12 n=-1.7+1.8*log10(Re)
13 //Value of V/U:
14 v_u=2*n^2/(n+1)/(2*n+1)
15 //Value of alpha:
16 alpha=(1/v_u)^3*2*n^2/(3+n)/(3+2*n)
17 //Loss Coefficient for a square edged entrance:
18 K=2*g*h/V2^2-f*L/D*12-alpha;
19 printf("\n\nRESULTS\n\n")
20 printf("\n\nLoss Coefficient for a square edged
       entrance: %.3f \n\n",K)
```

check Appendix AP 32 for dependency:

```
8_10.sci
```

Scilab code Exa 8.10 Volume and increase

```
1  //Volume and increase//
2  pathname=get_absolute_file_path('8.10.sce')
3  filename=pathname+filesep()+'8.10-data.sci'
4  exec(filename)
5  //Velocity V1(in m/s):
6  V1=sqrt(2*g*z0/1.04)
7  //Volume flow rate(in m^3/sec):
8  Q=V1*%pi*D^2/4
9  Kdiff=1-1/A_R^2-Cp
```

```
10  //For 2nd case:
11  //Velocity(in m/s):
12  V1=sqrt(2*g*z0/0.59)
13  //Volume flow rate(in m^3/s):
14  Qd=V1*%pi*D^2/4
15  //Increase in discharge after addition of diffuser
       is:
16  dQ=(Qd-Q)/Q*100
17  printf("\n\nRESULTS\n\n")
18  printf("\n\nVolume flow rate in case1: %.3f m^3/sec\
       n\n",Q)
19  printf("\n\nVolume flow rate in case 2: %.3f m^3/sec
       \n\n",Qd)
20  printf("\n\nIncrease in discharge after addition of
       diffuser is: %.3f percent\n\n",dQ)
```

check Appendix AP 31 for dependency:

```
8_11.sci
```

Scilab code Exa 8.11 Diameter and head

```
1   //Diameter and head//
2   pathname=get_absolute_file_path('8.11.sce')
3   filename=pathname+filesep()+'8.11-data.sci'
4   exec(filename)
5
6   //Value of K*B^2:
7   K_B=Q/(%pi/4*D^2)*sqrt(0.5*d1/g/d2/h)
8   //Reynods number:
9   ReD1=4/%pi*Q/D/v
10  //By trial and error method, the value of beta is
       fixed at:
11  betta=0.66;
12  //K is then:
13  K=K_B/betta^2
```

48

```
14  //Diameter of orifice plate(in m):
15  Dt=betta*D
16  //Value of p3-p2(in N/m^2):
17  P1=d1*Q^2/(%pi/4*D^2)^2*(1/0.65/betta^2-1)
18  //Value of p1-p2(in N/m^2):
19  P2=d2*g*h
20  //Head loss between sections 1 and 3(in N-m/kg):
21  hLT=(P2-P1)/d1
22  //Expressing the permanent pressure as a fractio of
        the meter differential:
23  C=(P2-P1)/P2
24  printf("\n\n\nRESULTS\n\n")
25  printf("\n\nDiameter of the orifice: %.3f m\n\n",Dt)
26  printf("\n\nHead loss between secions 1 and 3: %.3f
        N-m/kg\n\n",hLT)
```

Chapter 9

External Incompressible Viscous Flow

check Appendix AP 30 for dependency:

```
9_01.sci
```

Scilab code Exa 9.01 static pressure

```
1 //static pressure//
2 pathname=get_absolute_file_path('9.01.sce')
3 filename=pathname+filesep()+'9.01-data.sci'
4 exec(filename)
5 //Change in static pressure between sections 1 and
     2:
6 C=(((L-2*d1)/(L-2*d2))^4-1)*100;
7 printf("\n\nRESULTS\n\n")
8 printf("\n\nChange in static pressure between the
     sections 1 and 2: %.3f percent \n\n",C)
```

check Appendix AP 29 for dependency:

```
9_04.sci
```

Scilab code Exa 9.04 Displacement thickness and stress

```
1  //Displacement thickness and stress//
2  pathname=get_absolute_file_path('9.04.sce')
3  filename=pathname+filesep()+'9.04-data.sci'
4  exec(filename)
5  //Reynolds number:
6  ReL=U*L/v
7  //FOR TURBULENT FLOW
8  //Disturbance thickness(in m):
9  dL1=0.382/ReL^0.2*L
10 //Displacement thickness(in m):
11 function y=f(n),y=dL1*(1-n^(1/7))
12 endfunction
13 dl1=intg(0,1,f)
14 //Skin friction coefficient:
15 Cf1=0.0594/ReL^0.2
16 //Wall shear stress(in N/m^2):
17 tw1=Cf1*0.5*d*U^2
18 //For LAMINAR FLOW:
19 //Disturbance thickness(in m)
20 dL2=5/sqrt(ReL)*L
21 //Displacement thickness(in m):
22 dl2=0.344*dL2
23 //Skin friction coefficient:
24 Cf2=0.664/sqrt(ReL)
25 //Wall shear stress(in N/m^2):
26 tw2=Cf2*0.5*d*U^2
27 //COMPARISON OF VALUES WITH LAMINAR FLOW
28 //Disturbance thickness
29 D=dL1/dL2
30 //Displacement thickness
31 DS=dl1/dl2
32 //Wall shear stress
33 WSS=tw1/tw2
34 printf("\n\nRESULTS\n\n")
35 printf("\n\nDisturbace thickness: %.3f m\n\n",dL1)
36 printf("\n\nDisplacement thickness: %.3f m\n\n",dl1)
```

```
37  printf("\n\nWall shear stress: %f N/m^2\n\n",tw1)
38  printf("\n\nCOMPARISON WIH LAMINAR FLOW\n\n\n")
39  printf("\n\n Disturbance thicknes: %.3f \n\n",D)
40  printf("\n\nDisplacement thickness: %.3f\n\n",DS)
41  printf("\n\nWall shear stress: %.3f \n\n",WSS)
```

check Appendix AP 28 for dependency:

```
9_05.sci
```

Scilab code Exa 9.05 force and power

```
1  //force and power//
2  pathname=get_absolute_file_path('9.05.sce')
3  filename=pathname+filesep()+'9.05-data.sci'
4  exec(filename)
5  //Speed in m/s:
6  U=s*6076*0.305/3600
7  //Reynolds number:
8  Re=U*L/v
9  //Drag coefficient:
10 Cd=0.455/log10(Re)^2.58-1610/Re
11 //Area(in m^2):
12 A=L*(W+D)
13 //Drag force(in N)
14 Fd=Cd*A*0.5*d*U^2
15 //Power required to overcome skin friction drag(in W
      ):
16 P=Fd*U
17 printf("\n\nRESULTS\n\n")
18 printf("\n\nDrag force: %f N\n\n",Fd)
19 printf("\n\nPower required to overcome skin friction
       drag: %.3f W\n\n",P)
```

check Appendix AP 27 for dependency:

```
9_06.sci
```

Scilab code Exa 9.06 Bending moment

```
1  //Bending moment//
2  pathname=get_absolute_file_path('9.06.sce')
3  filename=pathname+filesep()+'9.06-data.sci'
4  exec(filename)
5  //Velocity in m/sec:
6  V=s*5/18
7  //Reynolds number:
8  Re=d*V*D/u
9  //Value of Cd is obtained as:
10 Cd=0.35;
11 //Area(in m^2):
12 A=L^2;
13 //Moment about the chimney base(in N-m):
14 M0=Cd*A*D/4*d*V^2
15 printf("\n\nRESULTS\n\n")
16 printf("\n\nBending moment at the bottom of the
       chimney: %.3f N-m\n\n",M0)
```

check Appendix AP 26 for dependency:

```
9_07.sci
```

Scilab code Exa 9.07 Time required

```
1  //Time required//
2  pathname=get_absolute_file_path('9.07.sce')
3  filename=pathname+filesep()+'9.07-data.sci'
4  exec(filename)
5  //Time required to decelerate to 100 mph(in seconds)
       :
```

```
6  t =( s1 - s2 ) *2* w /( s1 * s2 ) / Cd / d / A / g *3600/5280
7  printf ( " \n\nRESULTS \n\n " )
8  printf ( " \n\nTime  required  to  decelerate  to  100  mph :
      %.3 f  seconds \n\n " , t )
```

---

check Appendix AP 25 for dependency:

```
9_08.sci
```

Scilab code Exa 9.08 Optimum cruise speed

```
1  //Optimum  cruise  speed //
2  pathname = get_absolute_file_path ( ' 9.08. sce ' )
3  filename = pathname + filesep () + '9.08 − data . sci '
4  exec ( filename )
5  // Plotting  velocity  with  drag  force
6  V =175:25:455;
7
8  [m n ]= size ( V ) ;
9  for  i =1: n
10    CL ( i ) =2* W / p *(3600/ V ( i ) /5280) ^2/ A ;
11    Cd ( i ) = Cd0 + CL ( i ) ^2/%pi / ar ;
12    Fd ( i ) = Cd ( i ) / CL ( i ) * W ;
13    FD ( i ) = Fd ( i ) /1000;
14  end
15  plot ( V , FD )
16  xtitle ( ' Flight  speed  vs  thrust ' , ' Flight  Speed ( in  mph
       ) ' , ' Drag  Force ( in  1000 lbf ) ' )
17  //Optimum  cuise  speed  at  speed  level  is  obtained  to
       be  320  mph  from  the  graph .
18  Vosl =320;
19  // Ratio  of  speeds  at  30000  ft  and  at  sea  level  is
       given  by :
20  r = sqrt (1/0.375) ;
21  // Stall  speed  at  30000 ft  is ( in  mph ) :
22  Vs3 = Vssl * r ;
```

```
23  //Optimum  Cruise  speed  at  30000 ft (in  mph):
24  Vo3=Vosl*r;
25  printf("\n\nRESULTS\n\n")
26  printf("\n\nOptimum  cruise  speed  at  sea  level : %.3 f
        mph\n\n",Vosl)
27  printf("\n\nStall  speed  at  30000  ft : %.3 f  mph\n\n",
        Vs3)
28  printf("\n\nOptimum  cruise  speed  at  30000  ft : %.3 f\n
        \n",Vo3)
```

check Appendix AP 24 for dependency:

```
9_09.sci
```

Scilab code Exa 9.09 Aerodynamic and Radius

```
1  //Aerodynamic  and  Radius//
2  pathname=get_absolute_file_path('9.09.sce')
3  filename=pathname+filesep()+'9.09−data.sci'
4  exec(filename)
5  //Reynolds  number:
6  //Value  of  wD/2V:
7  W=0.5*N*D/1000/V*2*%pi/60
8  Red=V*D/v;
9  //For  this  value,  CL  is  obtained  as:
10 CL=0.3;
11 //Aerodynamic  lift (in  N):
12 FL=%pi/8*CL*(D/1000)^2*d*V^2;
13 //Radius  of  curvature  of  the  path  in  the  vertical
        plane(in  m)  with  topspin:
14 Rts=V^2/(g+FL/(m/1000));
15 //Radius  of  curvature  without  topspin(in  m):
16 Rwts=V^2/g;
17 printf("\n\nRESULTS\n\n")
18 printf("\n\nAerodynamic  lift  acting  on  the  ball :%.3 f
        N\n\n",FL)
```

```
19  printf("\n\nRadius of curvature of the path when
        ball has topspin:%.3f m\n\n",Rts)
20  printf("\n\nRadius of curvature of the path when
        ball has topspin: %.3f m\n\n",Rwts)
```

Chapter 10

Fluid Machinery

check Appendix AP 23 for dependency:

10_01.sci

check Appendix AP 17 for dependency:

10_11.sci

Scilab code Exa 10.01 input and power

```
1  //input and power//
2  pathname=get_absolute_file_path('10.01.sce')
3  filename=pathname+filesep()+'10.01-data.sci'
4  exec(filename)
5  //Impeller exit width b2(in feet):
6  b2=Q*12/(2*%pi*R2*Vrb2*7.48*60)
7  //Torque of the Shaft, Tshaft(in ft-lbf):
8  Tshaft=w*R2^2*p*Q*2*%pi/3600/7.48/144
9  //Power, Wm(in hp):
10 Wm=w*Tshaft*2*%pi/60/550
11 printf("\n\nRESULTS\n\n")
12 printf("\n\nImpeller exit width: %.3f feet\n\n",b2)
13 printf("\n\Torque input: %.3f ft-lbf\n\n",Tshaft)
14 printf("\n\nPower: %.3f hp\n\n",Wm)
```

Scilab code Exa 10.1 Performance curves

```
1  //Performance curves//
2  pathname=get_absolute_file_path('10.11.sce')
3  filename=pathname+filesep()+'10.11-data.sci'
4  exec(filename)
5  [nQ mQ]= size(Q1);
6  [np mp]= size(p1);
7  [nP mP]= size(P1);
8  //Volume flow rate for fan 2(in cfm):
9  j=1:mQ;
10 Q2=Q1(j)*(N2/N1)*(D2/D1)^3
11 //Pressure values for fan 2(in inches of H2O):
12 j=1:mp;
13 p2=p1(j)*(d2/d1)*((N2/N1)^2)*((D2/D1)^2)
14 //Power values for fan 2(in hp):
15 j=1:mP;
16 P2=P1(j)*(d2/d1)*((N2/N1)^3)*((D2/D1)^5)
17 plot(Q2,p2)
18 xtitle('Performance curves','Volume flow rate(in cfm
       )','Pressure head(in inches of water)')
19 printf("\n\nType (resume) to continue or (abort) to
       exit\n\n")
20 pause
21 clf
22 plot(Q2,P2)
23 xtitle('Performance curves','Volume flow rate(in cfm
       )','Power(in hp)')
24 printf("\n\nType (resume) to continue or (abort) to
       exit\n\n")
25 pause
26 clf
27 plot(Q2,Eff)
28 xtitle('Performance curves','Volume flow rate(in cfm
```

```
      ) ',' Eficiency ( in  percentage ) ')
29  // Specific  speed  of  fan ( in  US  customary  units )  at
        operating  point :
30  Nscu =  1150*110000^0.50*0.045^0.75/7.4^0.75
31  // Specific  speed  of  fan  ( in  SI  units )  at  operating
        point :
32  Nssi =120*3110^0.5*0.721^0.75/1.84 e3^0.75
```

---

check Appendix AP 22 for dependency:

```
10_02.sci
```

Scilab code Exa 10.02 volume and power

```
1  // volume  and  power //
2  pathname = get_absolute_file_path ( '10.02. sce ')
3  filename = pathname + filesep ()+ '10.02 − data . sci '
4  exec ( filename )
5  U =0.5*( Dh + Dt )/2*1200*2*% pi /60
6  k = tand ( alpha1 )+ cotd ( betta1 )
7  Vn1 = U / k
8  V1 = Vn1 / cosd ( alpha1 )
9  Vt1 = V1 * sind ( alpha1 )
10  Vrb1 = Vn1 / sind ( betta1 )
11  // Volume  flow  rate  ( in  m^3/ sec ) :
12  Q =% pi /4* Vn1 *( Dt ^2 - Dh ^2)
13  k =( U - Vn1 * cotd ( betta2 ))/ Vn1
14  alpha2 =  atand ( k )
15  V2 = Vn1 / cosd ( alpha2 )
16  Vt2 = V2 * sind ( alpha2 )
17  // Rotor  Torque  ( in  N–m ) :
18  Tz = p * Q *( Dh + Dt )/4*( Vt2 - Vt1 )
19  // Power  required  ( in  W ) :
20  Wm = w *2*% pi /60* Tz
21  printf ("\ n \ nRESULTS \ n \ n ")
22  printf ("\ n \ nVolume  flow  rate : %.3 f  m^3/ sec \ n \ n " ,Q )
```

```
23  printf("\n\nRotor  Torque:  %.3 f  N–m\n\n",Tz)
24  printf("\n\nPower  required :  %.3 f  W\n\n",Wm)
```

check Appendix AP 21 for dependency:

```
10_03.sci
```

Scilab code Exa 10.03 Pump Power

```
1  //Pump  Power //
2  pathname=get_absolute_file_path ( ' 10.03. sce ' )
3  filename=pathname+filesep ()+ '10.03− data . sci '
4  exec( filename )
5  [nQ mQ]= size (Q);
6  [nps mps]=size (ps);
7  [npd mpd]= size (pd);
8  [nI mI]= size (I);
9  //Correct  measured  static  pressures  to  he  pump
       centreline  p1 ,  p2 (in  psig ) :
10   j =1: mps ;
11    p1=ps (j)+px*g*zs /144
12     j =1: mpd ;
13    p2=pd (j)+px*g*zd /144
14    //The  value  of  Pump  head (in  feet ) :
15    j =1: mps ;
16    Hp=(p2 (j)-p1 (j))/(px*g)*144
17    //Values  of  Hydraulic  Power  delivered (in  hp ) :
18     j =1: mps ;
19     Wh=Q(j).*(p2 (j)-p1 (j))/7.48/60*144/550
20     //Values  of  motor  power  output (in  hp ) :
21    j =1: mI ;
22    Pin=Effm*sqrt (3)*PF*E*I (j)/746
23    //Values  of  Pump  Efficiecy :
24     j =1: mI ;
25    Effp=  Wh(j)./Pin (j)*100
26     //Plotting  pump  characteristics :
```

60

```
27      plot(Q,Hp,"-o")
28      plot(Q,Pin,"-+")
29      plot(Q,Effp,"-*")
30      xtitle('Pump Characteristics','Volume flow rate(in
            gpm)',['Pump Efficincy(%)     ','    Pump Head(
          in feet)     ','      Pump Power input(in hp)    '])
31      legend('Hp','Pin','Effp')
```

check Appendix AP 20 for dependency:

```
10_06.sci
```

Scilab code Exa 10.06 Specific and relation

```
1  //Specific and relation//
2  pathname=get_absolute_file_path('10.06.sce')
3  filename=pathname+filesep()+'10.06-data.sci'
4  exec(filename)
5  //Specific speed in Us customary units:
6  Nscu=N*Qus^0.5/Hus^0.75
7  //Conversion to SI units:
8  w=1170*2*%pi/60;
9  Qsi=Qus/7.48/60*0.305^3;
10 Hsi=Hus*0.305;
11 //Energy per unit mass is:
12 h=g*Hsi;
13 //Specific speed in SI units:
14 Nssi=w*Qsi^0.5/h^0.75
15 //Conversion to hertz:
16 whz=N/60;
17 //Specific speed in European units:
18 Nseu=whz*Qsi^0.5/65.5^0.75
19 //Relation between specific speeds in Us customary
        units and European units:
20 Conversionfactor1=Nscu/Nseu
```

```
21  // Relation  between  specific  speeds  in  Us  customary
        units  and  SI  units:
22  Conversionfactor2=Nscu/Nssi
23  printf("\n\nRESULTS\n\n")
24  printf("\n\nSpecific  speed  in  US  customary  units: %
        .3 f  \n\n",Nscu)
25  printf("\n\nSpecific  speed  in  SI  units: %.3 f  \n\n",
        Nssi)
26  printf("\n\nSpecific  speedin  European  units: %.3 f  \n
        \n",Nseu)
27  printf("\n\nRelation  between  specific  speeds  in  Us
        customary  units  and  European  units: %.3 f  \n\n",
        Conversionfactor1)
28  printf("\n\nRelation  between  specific  speeds  in  Us
        customary  units  and  SI  units: %.3 f  \n\n",
        Conversionfactor2)
```

check Appendix AP 19 for dependency:

```
10_07.sci
```

Scilab code Exa 10.07 Comparison of head

```
1  // Comparison  of  head //
2  pathname=get_absolute_file_path('10.07.sce')
3  filename=pathname+filesep()+'10.07−data.sci'
4  exec(filename)
5  // Volume  flow  rate(in  gpm)  at  shut  off  condition  for
        N2:
6  Q2so=N2/N1*Q1so
7  // Volume  flow(in  gpm)  rate  at  best  efficiency  for  N2
        :
8  Q2be=N2/N1*Q1be
9  // Relation  between  pump  heads:
10 head_relation=(N2/N1)^2
11 // Head(in  feet)  at  shut  off  condition  for  N2:
```

```
12  H2so =( N2 / N1 )^2* H1so
13  // Head ( in feet ) at best efficiency condition for N2 :
14  H2be =( N2 / N1 )^2* H1be
15  Q1 =[ Q1so Q1be ];
16  Q2 =[ Q2so Q2be ];
17  H1 =[ H1so H1be ];
18  H2 =[ H2so H2be ];
19  plot ( Q1 , H1 , " -o " )
20  plot ( Q2 , H2 , " -* " )
21  xtitle ( ' Comparison of head for both conditions ' , '
        Volume Flow Rate ' , ' Head ' )
22  legend ( ' 1170 ' , ' 1750 ' )
```

check Appendix AP 18 for dependency:

```
10_08.sci
```

Scilab code Exa 10.08 NPSHA and NPSHR

```
1   // NPSHA and NPSHR //
2   pathname = get_absolute_file_path ( ' 10.08. sce ' )
3   filename = pathname + filesep () + ' 10.08 - data . sci '
4   exec ( filename )
5   // Diameter of pipe ( in feet ) :
6   Df = Di /12
7   // Area of crossection of pipe ( in ft ^2) :
8   A =% pi /4* Df ^2
9   // Velocity of flow ( in ft / sec ) :
10  V = Q /7.48/ A /60
11  // For water at T =80 F , viscosity =0.927 e -5 ft ^2/ sec ,
        Reynolds number :
12  Re = V * Df / v
13  // Friction loss Coefficient for this value of Re :
14  f =0.0237;
15  // For cast iron , roughness ( in feet ) :
16  e =0.00085
```

```scilab
17  //e/D is:
18  e/Df
19  //Total head loss(in feet):
20  HL=K+f*(SE+OGV)+f*(L/Df)+1
21  //The heads are(in feet):
22  H1=patm*144/(p*g)
23  Vh=V^2/2/g
24  //Suction head(in feet):
25  Hs=H1+h-HL*Vh
26  //NPSHA(in feet):
27  NPSHA=Hs+Vh-Hv1
28  //For a flow rate of 1000 gpm,NPSHR(in feet) for
         water at 80 F
29  NPSHR=10
30  //PLOTTING NPSHA AND NPSHR VERSUS VOLUME FLOW RATE:
31  //For 80 F
32  Qp=0:100:1500;
33  [nQp mQp]=size(Qp);
34   for j=1:mQp;
35   Vp(j)=Qp(j)/(7.48*A*60);
36   Vhp(j)=(Vp(j))^2/2/g;
37   Hs(j)=H1+h-HL*Vhp(j);
38   end
39
40  for j=1:mQp;
41    NPSHAp1(j)=Hs(j)+(Vhp(j))-Hv1;
42  end
43
44  plot(Qp,NPSHAp1,"-+")
45  plot(Qh,NPSHRp,"-o")
46  xtitle('Suction head vs Flow rate','Volume flow rate
        (gpm)','Suction Head(feet)');
47  printf("\n\nType (Resume) to continue or (abort) to
        end\n\n")
48  legend('NPSHA','NPSHR')
49  pause
50  clf
51
```

```
52  //For 180 F
53  for j=1:mQp;
54     NPSHAp2(j)=Hs(j)+(Vhp(j))-Hv2;
55  end
56  plot(Qp,NPSHAp2,"-+")
57  plot(Qh,NPSHRp,"-o")
58  xtitle('Suction head vs Flow rate','Volume flow rate
        (gpm)','Suction Head(feet)');
59  legend('NPSHA','NPSHR')
60  printf("\n\nRESULTS\n\n")
61  printf("\n\nNPSHA at Q=1000 gpm of water at 80 F: %
        .2f ft\n\n",NPSHA)
62  printf("\n\nNPSHR at Q=1000 gpm of water at 80 F: %
        .1f ft\n\n",NPSHR)
```

check Appendix AP 16 for dependency:

`10_12.sci`

Scilab code Exa 10.12 Power required

```
1   //Power required//
2   pathname=get_absolute_file_path('10.12.sce')
3   filename=pathname+filesep()+'10.12-data.sci'
4   exec(filename)
5   //From given graph, for maximum delivery condition,
        Q=48.5gpm.
6   //Volume of oil per revolution delivered by the pump
        (in in^3/rev):
7   vc=Qe/N*231
8   //Volumetric Effciency of pump at max flow:
9   Effv=vc/va
10  //Operating point of the pump is found to be at 1500
        psig,Q=46.5gpm
```

```
11  //Power delivered by the fluid(in hp):
12  Pf=Qo*po1/7.48/60*144/550
13  //Input power(in hp):
14  Pi=Pf/Effp
15  //The power delivered to the load(in hp):
16  Pl=Q*(po1)/7.48/60*144/550
17  //Power dissipated by throttling(in hp):
18  Pd=Pf-Pl
19  //The dissipation with the variable displacement
        pump(in hp):
20  Pvd=Q*(po2-po1)/7.48/60*144/550
21  //Power required for te load sensing pump if pump
        pressure is 100psi above that required by the
        load(in hp):
22  Pls=Q*100/7.48/60*144/550
23  printf("\n\nRESULTS\n\n")
24  printf("\n\nVolume of oil per revolution delivered
        by the pump: %.3f in^3/rev\n\n",vc)
25  printf("\n\nRequired pump power input: %.3f hp\n\n",
        Pi)
26  printf("\n\nPower deliverd to the load: %.3f hp\n\n"
        ,Pl)
27  printf("\n\nPower dissipated by throttling: %.3f hp\
        n\n",Pd)
28  printf("\n\nThe dissipation with the variable
        displacement pump: %.3f hp\n\n",Pvd)
29  printf("\n\nPower required for te load sensing pump
        if pump pressure is 100psi above that required by
         the load: %.3f hp\n\n",Pls)
```

check Appendix AP 15 for dependency:

```
10_14.sci
```

Scilab code Exa 10.14 propeller

```
1  // propeller //
2  pathname = get_absolute_file_path ( ' 10.14. sce ' )
3  filename = pathname + filesep () + ' 10.14 − data . sci '
4  exec ( filename )
5  // Propeller Thrust ( in MN) :
6  Ft = P / V
7  // Required power input  to the propeller ( in MW) :
8  Pin = P / Eff
9  // Calculating value of D( in m ) :
10  nD = V / J
11  D = ( Ft * 10 ^6 / p / ( nD ) ^2 / Cf ) ^0.5
12  // Operating speed ( in rpm ) is given by :
13  n = nD / D * 60
14  printf ( " \ n \ nRESULTS \ n \ n " )
15  printf ( " \ n \ nDiameter of the single propeller
       required to pwer the ship :%.3 f m \ n \ n " , D )
16  printf ( " \ n \ nOperating speed of the propeller : %.3 f
       rpm \ n \ n " , n )
```

---

check Appendix AP 14 for dependency:

```
10_16.sci
```

Scilab code Exa 10.16 Actual

```
1  // Actual //
2  pathname = get_absolute_file_path ( ' 10.16. sce ' )
3  filename = pathname + filesep () + ' 10.16 − data . sci '
4  exec ( filename )
5  // Tip speed ratio of windmill :
6  X = N * 2 * % pi / 60 * D / 2 / ( V * 5 / 18 )
7  // Accounting for whirl , max attainable efficiency is :
8  Efw = 0.53;
9  // Kinetic energy flux ( in W) is given by :
10  KEF = 0.5 * p * ( V * 5 / 18 ) ^3 * % pi * ( D / 2 ) ^2
11  // Actual Efficiency :
```

```
12  Effa=Po/KEF
13  //The maximum possible thrust occurs for an
        interference factor of:
14  amax=0.5;
15  //Thrust(in W):
16  Kx=p*(V*5/18)^2*%pi*(D/2)^2*2*amax*(1-amax)
17  printf("\n\nRESULTS\n\n")
18  printf("\n\nTip speed ratio of windmill:%.3f\n\n",X)
19  printf("\n\nActual Efficiency: %.3f\n\n",Effa)
20  printf("\n\nActual Thrust: %.3f W\n\n",Kx)
```

Chapter 11

Introduction to Compressible Flow

check Appendix AP 13 for dependency:

    11_01.sci

Scilab code Exa 11.01 Change

```
1  //Change//
2  pathname=get_absolute_file_path('11.01.sce')
3  filename=pathname+filesep()+'11.01-data.sci'
4  exec(filename)
5  //Density of air at entry:
6  d1=p1*10^3/R/T1
7  //Area(in m^2):
8  A=m/d1/V1
9  //Change in enthalpy of air(in kJ/kg):
10 dh=cp*(T2-T1)
11 //Change in internal energy of air(in kJ/kg):
12 du=cv*(T2-T1)
13 //Change in entropy(in kJ/(kg-K)):
14 ds=cp*log(T2/T1)-R/1000*log(p2/p1)
15 printf("\n\nRESULTS\n\n")
```

```
16  printf("\n\nDuct Area: %.3f m^2\n\n",A)
17  printf("\n\nChange in enthalpy of air: %.3f kJ/kg\n\
        n",dh)
18  printf("\n\nChange in internal energy of air:%.3f kJ
        /kg\n\n",du)
19  printf("\n\nChange in entropy: %.3f kg-K\n\n",ds)
```

check Appendix AP 12 for dependency:

```
11_03.sci
```

Scilab code Exa 11.03 Speed of sound

```
1   //Speed of sound//
2   pathname=get_absolute_file_path('11.03.sce')
3   filename=pathname+filesep()+'11.03-data.sci'
4   exec(filename)
5   //Values of altitude(in m):
6   Al=0:1000:15000
7   [nAl mAl]=size(Al);
8   //Values of temperature at given altitudes(in K):
9   T=[288.2 281.7 275.2 268.7 262.2 255.7 249.2 242.7
        236.2 229.7 223.3 216.8 216.7 216.7 216.7 216.7];
10  [nT mT]=size(T);
11  //Values of speed of sound at these altitudes(in m/
        sec):
12  j=1:mT;
13  c=sqrt(k*R*T(j))
14  //Speed of sound at sea level(in m/sec):
15  c1=sqrt(k*R*T(1))
16  plot(c,Al)
17  xtitle('Variation of sound speed with altitude','
        Speed of sound(m/sec)','Altitude(m)')
18  printf("\n\nRESULTS\n\n")
19  printf("\n\nSpeed of sound at sea level: %.3f m/sec\
        n\n",c1)
```

check Appendix for dependency:

```
11_04.sci
```

Scilab code Exa 11.04 pressure and change

```
1  // pressure and change //
2  pathname = get_absolute_file_path('11.04.sce')
3  filename = pathname + filesep () + '11.04 - data.sci'
4  exec(filename)
5  // Mach number at entry:
6  M1 = V1/sqrt(k*R*T1)
7  // Stagnation pressure at entry(in kPa):
8  p01 = p1*(1+(k-1)/2*M1^2)^(k/(k-1))
9  // Stagnation temperature at entry(in K):
10 T01 = T1*(1+(k-1)/2*M1^2)
11 // Static pressure at exit(in kPa):
12 p2 = p02/(1+(k-1)/2*M2^2)^(k/(k-1))
13 // Temperature at exit(in K):
14 T2 = T02/(1+(k-1)/2*M2^2)
15 // Change in entropy(in kJ/kg-K):
16 ds = cp*log(T2/T1) - R/1000*log(p2/p1)
17 printf("\n\nRESULTS\n\n")
18 printf("\n\nStagnation pressure at entry: %.3f kPa\n
       \n",p01)
19 printf("\n\nStagnation temperature at entry: %.3f K\
       n\n",T01)
20 printf("\n\nStatic pressure at exit: %.3f kPa\n\n",
       p2)
21 printf("\n\nTemperature at exit: %.3f K\n\n",T2)
22 printf("\n\nChange in entropy: %.3f kJ/kg-K\n\n",ds)
```

Chapter 12

Steady One Dimensional Compressible Flow

check Appendix AP 10 for dependency:

```
12_01.sci
```

Scilab code Exa 12.01 pressure and area

```scilab
1  // pressure and area //
2  pathname=get_absolute_file_path('12.01.sce')
3  filename=pathname+filesep()+'12.01-data.sci'
4  exec(filename)
5  //Here the stagnation quantities are constant.
6  // Stagnation temperature(in K):
7  T0=T1*(1+(k-1)/2*M1^2)
8  //Stagnation pressure(in kPa):
9  p0=p1*((1+(k-1)/2*M1^2)^(k/(k-1)))
10 //Finding T2/T1:
11 T=t2/t1
12 //Temperature at exit(in K):
13 T2=T*T1
14 //Finding p2/p1:
15 P=P2/P1
```

```
16  //Pressure at exit(in kPa):
17  p2=P2*p1
18  //Density of air at exit(in kg/m^3):
19  d2=p2*10^3/R/T2
20  //Velocity of air at exit(in m/sec):
21  V2=M2*sqrt(k*R*T2)
22  //Finding A2/A1:
23  a=a2/a1
24  //Area at exit(in m^2):
25  A2=a*A1
26  printf("\n\nRESULTS\n\n")
27  printf("\n\nStagnation temperature: %.3f K\n\n",T0)
28  printf("\n\nStagantion pressure: %.3f kPa\n\n",p0)
29  printf("\n\nTemperature a exit %.3f K\n\n",T2)
30  printf("\n\nPressure at exit: %.3f kPa\n\n",p2)
31  printf("\n\nDensity of air at exit: %.3f kg/m^3\n\n"
        ,d2)
32  printf("\n\nVelocity of air at exit: %.3f m/sec\n\n"
        ,V2)
33  printf("\n\nArea at exit: %.3f \n\n",A2)
```

check Appendix AP 9 for dependency:

    12_02.sci

Scilab code Exa 12.02 Mass flow

```
1  //Mass flow//
2  pathname=get_absolute_file_path('12.02.sce')
3  filename=pathname+filesep()+'12.02-data.sci'
4  exec(filename)
5  //Checking for chocking:
6  c=pb/p0;
7  if(c<=0.528)
8     //choked
9  else
```

```
10    //Not choked
11    //Therefore pressure at exit = back pressure
12    pe=pb;
13    //Mach number at exit:
14    Me=(((p0/pe)^((k-1)/k)-1)*(2/(k-1)))^0.5
15    //Temperature at exit(in K):
16    Te=T0/(1+(k-1)/2*Me^2)
17    //Velocity at exit(in m/sec):
18    Ve=Me*sqrt(k*R*Te)
19    //Density at exit(in kg/m^3):
20    de=pe*10^3/R/Te
21    //Mass flow rate of air(kg/sec):
22    m=de*Ve*Ae
23  end;
24  printf("\n\nRESULTS\n\n")
25  printf("\n\nMach number at exit: %.3f\n\n",Me)
26  printf("\n\nMass flow rate of air: %.3f kg/sec\n\n",
       m)
```

check Appendix AP 8 for dependency:

```
12_03.sci
```

Scilab code Exa 12.03 mass and area

```
1  //mass and area//
2  pathname=get_absolute_file_path('12.03.sce')
3  filename=pathname+filesep()+'12.03-data.sci'
4  exec(filename)
5  //Saturation pressure(in psia):
6  p0=p1*(1+(k-1)/2*M1^2)^(k/(k-1))
7  //Checking for choking:
8  x=pb/p0;
9  if(x>0.528)
10   //Not choked
11  else
```

```
12    //choked
13  end
14  //As there is choking :
15  Mt=1;
16  //Velocity at entry :
17  V1=M1*sqrt(k*R*(T1+460)*32.2)
18  //Density at the entry(in lbm/ft^3):
19  d1=p1/(R*(T1+460))*144
20  //Mass flow rate(in lbm/sec):
21  m=d1*V1*A1
22  //Finding the valueof A1/A*;
23  A=1/M1*((1+(k-1)/2*M1^2)/(1+(k-1)/2))^((k+1)/(2*(k
      -1)))
24  //For choked flow , At=A*
25  At=A1/A
26  printf("\n\nRESULTS\n\n")
27  printf("\n\nMach number at throat : %.3 f\n\n",Mt)
28  printf("\n\nMass flow rate : %.3 f lbm/sec\n\n",m)
29  printf("\n\nArea at throat : %.3 f ft^2\n\n",At)
```

check Appendix AP 7 for dependency:

```
12_04.sci
```

Scilab code Exa 12.04 throat

```
1  //throat//
2  pathname=get_absolute_file_path('12.04.sce')
3  filename=pathname+filesep()+'12.04−data.sci'
4  exec(filename)
5  //Temperature at the throat(in K):
6  Tt=T0/(1+(k-1)/2*Mt^2)
7  //Pressure at throat(in kPa):
8  pt=p0*(Tt/T0)^(k/(k-1))
9  //Density at throat(in kg/m^3):
10  dt=pt*1000/R/Tt
```

```
11  //Velocity at the throat(in m/s):
12  Vt=Mt*sqrt(k*R*Tt)
13  //Value of At/A*:
14  Ax=1/Mt*((1+(k-1)/2*Mt^2)/(1+(k-1)/2))^((k+1)/(2*(k
        -1)))
15  //Stagnation properties are constant
16  //As a result pressure at exit,
17  pe=pb;
18  //The Mach number at the exit is therefore given by
19  Me=sqrt(((p0/pe)^((k-1)/k)-1)*2/(k-1))
20  //Calculating the value of Ae/A*:
21  Ay=1/Me*((1+(k-1)/2*Me^2)/(1+(k-1)/2))^((k+1)/(2*(k
        -1)))
22  //Value of A*(in m^2):
23  A_star=Ae/Ay
24  //Area at throat(in m^2):
25  At=Ax*A_star
26  printf("\n\nRESULTS\n\n")
27  printf("\n\nTemperature at the throat: %.3f K\n\n",
        Tt)
28  printf("\n\nPressure at throat: %.3f kPa\n\n",pt)
29  printf("\n\nDensity at throat: %.3f kg/m^3\n\n",dt)
30  printf("\n\nVelocity at the throat: %.3f m/sec\n\n",
        Vt)
31  printf("\n\nMach number at the exit: %.3f\n\n",Me)
32  printf("\n\nArea at throat: %.3f m^2\n\n",At)
```

check Appendix AP 6 for dependency:

```
12_05.sci
```

Scilab code Exa 12.05 number and flow

```
1  //number and flow//
2  pathname=get_absolute_file_path('12.05.sce')
3  filename=pathname+filesep()+'12.05-data.sci'
```

```
4 exec(filename)
5 //Mach number at the exit:
6 Me=sqrt(((p0/pe)^((k-1)/k)-1)*2/(k-1))
7 //Temperature at exit(in K):
8 Te=T0/(1+(k-1)/2*Me^2)
9 //Mass flow rate(in kg/s):
10 m=pe*1000*Me*sqrt(k/R/Te)*Ae
11 printf("\n\nRESULTS\n\n")
12 printf("\n\nMach number at the exit: %.3f \n\n",Me)
13 printf("\n\nMass flow rate: %.3f kg/sec\n\n",m)
```
──

check Appendix AP 5 for dependency:

```
12_06.sci
```

Scilab code Exa 12.06 mass and volume

```
1 //mass and volume//
2 pathname=get_absolute_file_path('12.06.sce')
3 filename=pathname+filesep()+'12.06-data.sci'
4 exec(filename)
5 //Mach umber at section 1:
6 M1=sqrt((2/(k-1)*((p0/p1)^((k-1)/k)-1)))
7 //Temperature at section 1(in K):
8 T1=T0/(1+(k-1)/2*M1^2)
9 //Density at section 1(in kg/m^3):
10 d1=p1*1000/R/T1
11 //Velocity at section1(in m/sec):
12 V1=M1*sqrt(k*R*T1)
13 //Area at section 1(in m^2):
14 A1=%pi/4*D^2
15 //Mass flow rate(in kg/sec):
16 m=d1*A1*V1
17 //Mach number at section 2:
18 M2=sqrt((2/(k-1))*((T0/T2)-1))
19 //Velocity at section 2(in m/sec):
```

```
20  V2=M2*sqrt(k*R*T2)
21  //Density at section 2(in kg/m^3):
22  d2=d1*V1/V2
23  //Pressure at section 2(in kPa):
24  p2=d2/1000*R*T2
25  //Stagnation pressure at section 2(in kPa):
26  p02=p2*(1+(k-1)/2*M2^2)^(k/(k-1))
27  //Force exerted on control volume by duct wall(in N)
       :
28  F=(p2-p1)*1000*A1+m*(V2-V1)
29  printf("\n\nRESULTS\n\n")
30  printf("\n\nMass flow rate: %.3f kg/sec\n\n",m)
31  printf("\n\nLocal isentropic stagnation pressure at
       section 2:%.3f kPa\n\n",p02)
32  printf("\n\nForce exerted on control volume by duct
       wall:%.3f N\n\n",F)
```

check Appendix AP 4 for dependency:

```
12_07.sci
```

Scilab code Exa 12.07 length

```
1  //length//
2  pathname=get_absolute_file_path('12.07.sce')
3  filename=pathname+filesep()+'12.07-data.sci'
4  exec(filename)
5  //Mach number at section 1:
6  M1= sqrt(2/(k-1)*((p0/(p0+p1))^((k-1)/k)-1))
7  //Temperature at section 1(in K):
8  T1=T0/(1+(k-1)/2*(M1)^2)
9  V1=M1*sqrt(k*R*T1)
10 //Pressure at section 1(in kPa):
11 p1=g*dHg*(760-18.9)*10^-3
12 //Density at section 1(in kg/m^3):
13 d1=p1/R/T1
```

```
14  //At M1=0.190,
15  //(p/p*)1:
16  P1=5.745
17  //  (fLmax/Dh)1:
18  F1=16.38
19  //Value  of  L13(in m):
20  L13=F1*D/f
21  //Value  of  (p/p*)2:
22  P2=p2/p1*P1
23  //For  this  value,  Value  of  M2  is  obtained  as  0.4
24  M2=0.4;
25  //For M=0.4, fLmX/D=2.309
26  F2=2.309
27  //Value  of  L23(in m):
28  L23=F2*D/f
29  //Length  of  duct  between  section  1  and  2(in m):
30  L12=L13-L23
31  printf("\n\nRESULTS\n\n")
32  printf("\n\nLength  of  duct  required  for  choking  from
           section  1:  %3f m\n\n",L13)
33  printf("\n\nMach  number  section  2:  %.3f  \n\n",M2)
34  printf("\n\Length  of  duct  between  section  1  and  2: %
           .3f m\n\n",L12)
```

check Appendix AP 3 for dependency:

```
12_08.sci
```

Scilab code Exa 12.08 velocity and entropy

```
1  //velocity  and  entropy//
2  pathname=get_absolute_file_path('12.08.sce')
3  filename=pathname+filesep()+'12.08-data.sci'
4  exec(filename)
5  //Density  at  section  1(in  lbm/ft^3):
6  d1=p1*144/R/T1
```

```
 7  //Velocity at section 2(in ft/sec):
 8  V2=(p1-p2)*144/d1/V1*32.2+V1
 9  //Density at section 2(in lbm/ft3):
10  d2=d1*V1/V2
11  //Temperature at section 2(in R):
12  T2=p2/d2/R*144
13  //Mach number at section 2:
14  M2=V2/sqrt(k*R*32.16*T2)
15  //Stagnation Temperature at section 2(in R):
16  T02=T2*(1+(k-1)/2*M2^2)
17  //Stagnation pressure at section 2 (in psia):
18  p02=p2*(T02/T2)^(k/(k-1))
19  //Mach Number at section 1:
20  M1=V1/sqrt(k*R*32.16*T1)
21  //Stagnation temperature at section 1(in R):
22  T01=T1*(1+(k-1)/2*M1^2)
23  //Energy added(in Btu/lbm):
24  E=Cp*(T02-T01)
25  //Change in entropy(in Btu/(lbm-R)):
26  dS=Cp*log(T2/T1)-(Cp-Cv)*log(p2/p1)
27  printf("\n\nRESULTS\n\n")
28  printf("\n\nVelocity at section 2: %.3f ft/sec\n\n",
        V2)
29  printf("\n\nDensity at section 2: %.3f lbm/ft^3\n\n"
        ,d2)
30  printf("\n\nTemperature at section 2: %.3f R\n\n",T2
        )
31  printf("\n\nStagnation Temperature at section 2: %.3
        f R\n\n",T02)
32  printf("\n\nStagnation pressure at section 2: %.3f
        psia\n\n",p02)
33  printf("\n\nEnergy added: %.3f Btu/lbm\n\n",E)
34  printf("\n\nChange in entropy: %.3f Btu/(lbm-R)\n\n"
        ,dS)
```

check Appendix AP 2 for dependency:

`12_09.sci`

Scilab code Exa 12.09 Temperature and entropy

```
1  //Temperature and entropy//
2  pathname=get_absolute_file_path('12.09.sce')
3  filename=pathname+filesep()+'12.09-data.sci'
4  exec(filename)
5  //Mach nuber at section 1:
6  M1=V1/sqrt(k*R*T1)
7  //For these value of M1 and M2,the following values
       are obtained:
8  //(To/T0*)1:
9  t01=0.7934;
10 //(T0/T0*)2:
11 t02=0.9787;
12 //(p0/p0*)1:
13 P01=1.503;
14 //(p0/p0*)2:
15 P02=1.019;
16 //(T/T*)1:
17 t1=0.5289;
18 //(T/T*)2:
19 t2=0.9119;
20 //(p/p*)1:
21 P1=0.3636;
22 //(p/p*)2:
23 P2=0.7958;
24 //(V/V*)1:
25 v1=1.455;
26 //(V/V*)2:
27 v2=1.146;
28 //Value of T2/T1:
29 t=t2/t1
30 //Temperature at section 2(in K):
31 T2=t*T1
```

```
32  //Value of p2/p1:
33  p=P2/P1
34  //Pressure at section 2(in kPa):
35  p2=p*p1
36  //Value of V2/V1:
37  v=v2/v1
38  //Velocity at section 2(in m/sec):
39  V2=v*V1
40  //Density at section 2(in kg/m^3):
41  d2=p2*1000/R/T2
42  //At M1, T/T0=0.5556
43  T01=T1/0.5556
44  //At M2, T/T0=0.7764
45  T02=T2/0.7764
46  //Heat added(in kJ/kg):
47  E=Cp*(T02-T01)
48  //Change in entropy(kJ/kg-K):
49  dS=Cp*log(T2/T1)-R*log(p2/p1)/1000
50  printf("\n\nRESULTS\n\n")
51  printf("\n\nTemperature at section 2: %.3f K\n\n",T2
       )
52  printf("\n\nPressure at section 2: %.3f kPa\n\n",p2)
53  printf("\n\nVelocity at section 2: %.3f m/sec\n\n",
       V2)
54  printf("\n\nDensity at section 2: %.3f kg/m^3\n\n",
       d2)
55  printf("\n\nStagnation temperature at section 2: %.3
       f K\n\n",T02)
56  printf("\n\nHeat added: %.3f kJ/kg\n\n",E)
57  printf("\n\nChange in entropy: %.3f kJ/kg\n\n",dS)
```

check Appendix AP 1 for dependency:

```
12_10.sci
```

Scilab code Exa 12.10 Temperature

```scilab
1  //Temperature//
2  pathname=get_absolute_file_path('12.10.sce')
3  filename=pathname+filesep()+'12.10-data.sci'
4  exec(filename)
5  //Density at section 1(in kg/m^3):
6  d1=p1*1000/R/T1
7  //Mach number at section 1:
8  M1=V1/sqrt(k*R*T1)
9  //Stagnation temperature at section 1(in K):
10 T01=T1*(1+(k-1)/2*M1^2)
11 //Stagnation pressure at section 1(in kPa):
12 p01=p1*(1+(k-1)/2*M1^2)^(k/(k-1))
13 //The following values are obtained from the
       appendix:
14 //po2/p01:
15 p0=0.7209;
16 //T2/T1:
17 T=1.687;
18 //p2/p1:
19 p=4.5;
20 //V2/V1:
21 V=0.3750;
22 //Temperature at section 2 (in K):
23 T2=T*T1
24 //Pressure at section 2(in kPa):
25 p2=p*p1
26 //Velocity at section 2(in m/sec):
27 V2=V*V1
28 //Density at section 2 (in kg/m^3):
29 d2=p2*1000/R/T2
30 //Stagnation pressure at section 2(in kPa):
31 p02=p0*p01
32 //Stagnation temperature at section 2(in K):
33 T02=T01;
34 //Change in entropy(in kJ/(kg-K)):
35 dS=-R/1000*log(p0)
36 printf("\n\nRESULTS\n\n")
```

```
37  printf("\n\nTemperature at section 2 : %.3f K\n\n",
       T2)
38  printf("\n\nPressure at section 2: %.3f kPa\n\n",p2)
39  printf("\n\nVelocity at section 2: %.3f m/sec\n\n",
       V2)
40  printf("\n\nDensity at section 2 : %.3f kg/m^3\n\n",
       d2)
41  printf("\n\nStagnation pressure at section 2: %.3f
       kPa\n\n",p02)
42  printf("\n\nChange in entropy: %.3f kg-K\n\n",dS)
43  printf("\n\nStagnation temperature at section 2: %.3
       f K\n\n",T02)
```

Appendix

Scilab code AP 1 data

```
1  //Temperature at section 1(in K):
2  T1=278;
3  //Pressure at section 1(in kPa):
4  p1=65;
5  //Velocity at section 1(in m/sec):
6  V1=668;
7  //Value of gas constant(in N–m/kg–K):
8  R=287;
9  //Value of k:
10 k=1.4;
```

Scilab code AP 2 data

```
1  //Temperature at section 1(in K):
2  T1=333;
3  //Pressure at section 1(in kPa):
4  p1=135;
5  //Velocity at section 1(in m/sec):
6  V1=732;
7  //Mach number at section 2:
8  M2=1.2;
9  //Gas constant(in N–m/kg–K):
10 R=287;
11 //Value of k:
12 k=1.4;
13 //Specific heat at constant presure(in N–m/kg–K):
```

```
14  Cp =1;
```

Scilab code AP 3 data

```
1   //Temperature  at  section  1(in  R):
2   T1 =600;
3   //Pressure  at  section  1(in  psia):
4   p1 =20;
5   //Pressure  at  section  2(in  psia):
6   p2 =10;
7   //Velocity  at  section  1(in  ft/s):
8   V1 =360;
9   //Cross−sectional  area  of  the  duct(in  ft^2):
10  A =0.25;
11  //Gas  Constant(in  ft−lbf/lbm−R):
12  R =53.3;
13  //Value  of  k:
14  k =1.4;
15  //Specific  heat  at  constant  presure(in  Btu/lbm−R):
16  Cp =0.24;
17  //Specific  heat  at  constant  volume(in  Btu/lbm−R):
18  Cv =0.171;
```

Scilab code AP 4 data

```
1   //Stagnation  temperature(in  K):
2   T0 =296;
3   //Stagnation  pressure(in  mm  of  Hg):
4   p0 =760;
5   //Gauge  pressure  at  section  1(in  mm  of  Hg):
6   p1 = -18.9;
7   //Gauge  pressure  at  section2(in  mm  of  Hg):
8   p2 = -412;
9   //Mach  number  at  3:
10  M3 =1;
11  //Gas  constant:
12  R =287;
13  //Density  of  mercury(kg/m^3):
```

```
14  dHg =13500;
15  // Acceleration due to gravity(in m/sec^2):
16  g=9.8;
17  // Friction factor:
18  f=0.0235;
19  // Diameter of tube(in m):
20  D=7.16*10^-3;
21  // Value of k:
22  k=1.4;
```

Scilab code AP 5 data

```
1   // Diameter of pipe(in m):
2   D=7.16*10^-3;
3   // Stagnation pressure(in kPa):
4   p0=101;
5   // Stagnation temperature(in K):
6   T0=296;
7   // Pressure at section 1(in kPa):
8   p1=98.5;
9   // Temperature at section 2(in K):
10  T2=287;
11  // Gas constant(in N–m/kg–K):
12  R=287;
13  // Value of k:
14  k=1.4;
```

Scilab code AP 6 data

```
1   // Stagnation temperature(in K):
2   T0=350;
3   // Stagnation pressure(in kPa)\
4   p0=1000;
5   // Pressure at exit(in kPa)
6   pe=87.5;
7   // Back Pressure(in kPa):
8   pb=50;
9   // Area at exit(in m^2):
```

```
10  Ae =0.001;
11  //Gas Constant(in N–m/kg–K)
12  R=287;
13  //Value of k:
14  k=1.4;
```

Scilab code AP 7 data

```
1  //Stagnation temperature(in K):
2  T0=350;
3  //Stagnation pressure(in kPa):
4  p0=1000;
5  //Back Pressure(in kPa):
6  pb=954;
7  //Mach number at throat:
8  Mt=0.68;
9  //Area at exit(in m^2):
10  Ae=0.001;
11  //Value of k:
12  k=1.4;
13  //Gas Constant(in N–m/kg–K):
14  R=287;
```

Scilab code AP 8 data

```
1  //Mach number at entry:
2  M1=0.52;
3  //Temperature at entry(in F):
4  T1=40;
5  //Pressure at entry(in psia):
6  p1=60;
7  //Area at entry(in ft^2):
8  A1=0.013;
9  //Back pressure(in psia):
10  pb=30;
11  //Gas Consant(in ft–lbf/lbm–R)
12  R=53.3;
13  //Value of k:
```

```
14  k =1.4;
```

Scilab code AP 9 data

```
1  //Throat  area  of  nozzle(in  m^2):
2  Ae =0.001;
3  //Back  pressure  of  air(in  kPa):
4  pb =591;
5  //Stagnation  pressure(in  kPa):
6  p0 =1000;
7  //Stagnation  temperature(in  K):
8  T0 =333;
9  //Gas  Constant(in  N–m/kg–K):
10  R =287;
11  //Value  of  k:
12  k =1.4;
```

Scilab code AP 10 data

```
1  //Mach  number  at  entry:
2  M1 =0.3;
3  //Temperature  at  entry(in  K):
4  T1 =335;
5  //Pressure  at  entry(in  kPa):
6  p1 =650;
7  //Area  at  entry(in  m^2):
8  A1 =0.001;
9  //Mach  number  at  exit:
10  M2 =0.8;
11  ///Value  of  k:
12  k =1.4;
13  //For  the  Mach  no:0.3:
14  //T/T0:
15  t1 =0.9823,
16  //p/p0:
17  P1 =0.9395;
18  //d/d0:
19  den1 =0.9564;
```

```
20  //A/A*:
21  a1=2.035;
22  //For the Mach no:0.8:
23  //  T/T0:
24  t2=0.8865;
25  //p/p0:
26  P2=0.6560;
27  //d/d0:
28  den2=0.7400;
29  //A/A*:
30  a2=1.038;
31  //Gas Constant(in N–m/kg–K):
32  R=287;
```

Scilab code AP 11 data

```
 1  //Pressure at entry(in kPa):
 2  p1=350;
 3  //Temperature at entry(in K)
 4  T1=333;
 5  //Velocity at entry(in m/s):
 6  V1=183;
 7  //Mach no. at exit:
 8  M2=1.3;
 9  //Stagnation pressure at exit(in kPa):
10  p02=385;
11  //Stagnation temperature at exit(in K):
12  T02=350;
13  //Value of k:
14  k=1.4;
15  //Gas constant(in N–m/kg–K)
16  R=287;
17  //Specific heat at constant pressure(kJ/(kg–K):
18  cp=1;
```

Scilab code AP 12 data

```
 1  //Value of k:
```

```
2  k=1.4;
3  //Gas  Constant (in  Kj/(kg-K)):
4  R=287;
```

Scilab code AP 13 data

```
1  //Temperature  of  air  entering  the  cold  section (in  K)
      :
2  T1=440;
3  //Absolute  pressure  of  air  entering  the  cold  section
      (in  kPa):
4  p1=188;
5  //Velocity  of  air  entering  the  cold  section (in  m/sec
      ):
6  V1=210;
7  //Temperature  of  air  at  outlet :(in  K)
8  T2=351;
9  //Absolute  pressure  of  air  at  outlet (in  kPa):
10  p2=213;
11  //Rate  of  heat  loss  in  the  section (in  kJ/sec):
12  //Gas  Constant (in  N-m):
13  R=  287;
14  //Mass  flow  rate  of  air (in  kg/sec):
15  m=0.15;
16  //Specific  heat  at  constant  pressue (in  kJ/(kg-K)):
17  cp=1;
18  //Specific  energy  at  constant  volume (in  kJ/(kg-K)):
19  cv=0.717;
```

Scilab code AP 14 data

```
1  //Diameter  of  windmill (in  m):
2  D=26;
3  //Operating  speed (in  rpm):
4  N=20;
5  //Wind  speed (in  km/hr):
6  V=36;
7  //Power  Output (in  W):
```

```
8  Po =41000;
9  //Maximum efficiency occurs in following conditions:
10 //Efficiency:
11 Eff =0.593;
12 //Inteference Factor:
13 a=1/3;
14 //Density of air(in kg/m^3):
15 p=1.23;
```

Scilab code AP 15 data

```
1  //Total propulsion power requirement(in MW):
2  P=11.4;
3  //From the given curves,Value of coefficients
       atoptimum efficiency are as follows:
4  //Speed of advance coefficient:
5  J=0.85;
6  //Thrust Coefficient:
7  Cf =0.1;
8  //Torque Coefficint:
9  Ct =0.02;
10 //Efficiency:
11 Eff =0.66;
12 //Velocity of ship(in m/sec):
13 V=6.69;
14 //Density of water(in kg/m^3):
15 p=1025;
```

Scilab code AP 16 data

```
1  //Operation speed(in rpm):
2  N=2000;
3  //Volume flow rate(in gpm):
4  Q=20;
5  //Pressure(in psig):
6  p=1500;
7  //Actual Pump Displacement(in^3/rev):
8  va =5.9;
```

```
 9  //Volume flow rate at operating condition(in gpm):
10  Qo=46.5;
11  //Volume flow rate at maximum delivery(in gpm):
12  Qe=48.5;
13  //Pressure at operation condition(in psi):
14  po1=1500;
15  //Efficiency of pump at operating condition:
16  Effp=0.84;
17  //Pressure at operating condition case 2(in psig):
18  po2=3000;
```

Scilab code AP 17 data

```
 1  //Diameter of fan 1 (in inches):
 2  D1=36;
 3  //Operating speed of fan 1(in rpm):
 4  N1=600
 5  //Density of air used in fan 1(in lbm/ft^3):
 6  d1=0.075;
 7  //Diameter of fan 2(in inches):
 8  D2=42;
 9  //Operating speed of fan 2(in rpm):
10  N2=1150;
11  //Density of aif usd in fan 2(in lbm/ft^3):
12  d2= 0.045;
13  //The following values are obtained from the given
        graph
14  //Values of volume flow rate(in cfm) through fan 1:
15  Q1= [0 10000 20000 30000 40000 50000 60000];
16  //Values of pressure( in inches of H2O):
17  p1=[ 3.68 3.75 3.50 2.96 2.12 1.02 0];
18  //Values of power(in hp):
19  P1=[ 11.1 15.1 18.6 21.4 23.1 23.1 21.0];
20  //Efficiency (in %):
21  Eff=[0 37 59 65 57 34 0];
```

Scilab code AP 18 data

```
1  //For 5 inch nominal pipe line, diameter D:
2  Di=5.047;
3  //Length of pipeline(in feet):
4  L=6;
5  //Operatng spped (in rpm):
6  N=1750;
7  //Water level abovepump centreline(in feet):
8  h=3.5;
9  //Temperature 1 of water(in Farenheit):
10 T1=80;
11 //Temperature 2 of water (in Farenheit):
12 T2=180;
13 //Volume flow rate of water(in gpm):
14 Q=1000;
15 //Minor loss Coefficients:
16 K=0.5; SE=30; OGV=8;
17 //Atmospheric pressure(in lbf/in^2):
18 patm=14.7;
19 //Density of air(slug/ft^3):
20 p=1.93;
21 //Acceleration due to gravity(in ft/sec^2):
22 g=32.2;
23 //Head(in feet) due to vapor pressure of water for T
      =80F:
24 Hv1=1.17;
25 //Head(in feet) due to vapor pressure of water for T
      =180F:
26 Hv2=17.3;
27 //Kinematic viscosity of water at 80F:
28 v=0.927e-5;
29 //Value of discharges for plotting NPSHR(in gpm):
30 Qh=[500 700 900 1100 1300]
31 //Values of NPSHR obtained from Fig. D3 of appendix
     D:
32 NPSHRp=[7 8 9.5 12 16]
```

Scilab code AP 19 data

```
1  //Volume flow rate(in gpm) at shut off condition for
      N1:
2  Q1so=0;
3  //Volume flow(in gpm) rate at best efficiency for N1
      :
4  Q1be=300;
5  //Head(in feet) at shut off condition for N1:
6  H1so=25;
7  //Head(in feet) at best efficiency condition for N1:
8  H1be=21.9
9  //Operation Speed 1:
10  N1=1170;
11  //Operation speed 2:
12  N2=1750;
```

Scilab code AP 20 data

```
1  //Head in Us customary units:
2  Hus=21.9;
3  //Volume flow rate in US customary units:
4  Qus=300;
5  //Working seed in rpm:
6  N=1170;
7  //Aceleration due to graviy in m/s^2
8  g=9.81;
```

Scilab code AP 21 data

```
1  //Rate of flow in gm:
2  Q=[0 500 800 1000 1100 1200 1400 1500];
3  //Suction pressure in psig:
4  ps=[ 0.65 0.25 -0.35 -0.92 -1.24 -1.62 -2.42 -2.89];
5  //Discharge pressure in psig:
6  pd=[53.3 48.3 42.3 36.9 33 27.8 15.3 7.3];
7  //Motor Current in amps:
8  I=[18 26.2 31 33.9 35.2 36.3 38 39];
9  //Acceleration due to gravity in ft/s^2:
10  g=32.2;
```

```
11  //Value of Zs in feet
12  zs =1;
13  //Density of air in slug/ft^3:
14  px =1.94;
15  //Value of ZD in feet:
16  zd =3;
17  //Density of fluid in slug/ft^3:
18  py =1000;
19  //Motor Efficiency:
20  Effm =0.9;
21  //Motor Supply in volts:
22  E =460;
23  //Power Factor:
24  PF =0.875;
```

Scilab code AP 22 data

```
1  //Tip Diameter in metres:
2  Dt =1.1;
3  //Hub Diameter in metres:
4  Dh =0.8;
5  //Operating Speed in rpm:
6  w =1200;
7  //Absolute inlet angle in degrees:
8  alpha1 =30;
9  //Blade inlet angle in degrees:
10 betta1 =30;
11 //Blade outlet angle in degrees:
12 betta2 =60;
13 //Density of air in kg/m^3
14 p =1.23;
```

Scilab code AP 23 data

```
1  //Volume flow rate in gpm:
2  Q= 150;
3  //Value of Vrb2 in ft/sec:
4  Vrb2 =10;
```

```
 5  //Radius of outter impeller in inches:
 6  R2=2;
 7  //Impeller Speed in rpm:
 8  w=3450;
 9  //Density of air in slug/ft^3
10  p=1.94;
```

Scilab code AP 24 data

```
 1  //Mass of the tennis ball(in grams):
 2  m=57;
 3  //Diameter ofthe ball (in mm):
 4  D=64;
 5  //Velocity with which te ball is hit(in m/s):\
 6  V=25;
 7  //Topspin given on the ball(in rpm):
 8  N=7500;
 9  //Acceleration due to gravity(in m/s^2):
10  g=9.81;
11  //Kinematic viscosity(in m^2/s):
12  v=1.46*10^-5
13  //Desity of air(in kg/m^3):
14  d=1.23;
```

Scilab code AP 25 data

```
 1  //Wing area(in ft^2):
 2  A=1600;
 3  //Aspect ratio:
 4  ar=6.5;
 5  //Groos weight of aircraft(in lbf):
 6  W=150000;
 7  //Coefficient of drag at zero lift :
 8  Cd0=0.0182;
 9  //Sonic speed at sea level(in mph):
10  c=759;
11  //Density of air(in slug/ft^3):
12  p=0.00238;
```

```
13 // Srall speed at sea level(in mph):
14 Vssl=175;
```

Scilab code AP 26 data

```
1 //Weight of the dragster(in lbf):
2 w=1600;
3 //Speed of dragster(in mph):
4 s1=270;
5 //Area of drag chute(in ft^2):
6 A=25;
7 //Speed of dragster after deceleraton(in mph):
8 s2=100;
9 //Acceleration due to gravity(in ft/sec^2):
10 g=32.2;
11 //Density of air(in slug/ft^3):
12 d=0.00238;
13 //Value of coefficient of drag:
14 Cd=1.42;
```

Scilab code AP 27 data

```
1 //Diameter of chimney(in m):
2 D=1;
3 //Height of chimney(in m):
4 L=25;
5 //Speed of wind(in kmph):
6 s=50;
7 //Density of air(in kg/m^3):
8 d=1.23;
9 //Viscosity of air(in kg/(m-s)):
10 u=1.79*10^-5;
11 //Pressure(in kPa):
12 p=101;
```

Scilab code AP 28 data

```
1 //Length of the supertanker(in m):
2 L=360;
3 //Width of supertanker(in m):
4 W=70;
5 //Draft of the supertanker(in m):
6 D=50;
7 //Cruising speed in water(in knots):
8 s=13;
9 //Kinematic viscosity at 10 C
10 v=1.37*10^-6;
11 //Density of sea water(in kg/m^3):
12 d=1020;
```

Scilab code AP 29 data

```
1 //Veocity of flow(in m/sec):
2 U=1;
3 //Length of flat plate(in m):
4 L=1;
5 //Density of water(in kg/m^3):
6 d=999;
7 //Kinematic viscosity of water(in m^2/sec):
8 v=10^-6;
```

Scilab code AP 30 data

```
1 //Lengh of side of the test section(in mm):
2 L=305;
3 //Freesteam speed at section 1(in m/sec):
4 U1=26;
5 //Displacement thickness at section 1(in mm):
6 d1=1.5;
7 //Displacment thickness at section 2(in mm):
8 d2=2.1;
```

Scilab code AP 31 data

99

```
 1  // Volume flw rate of ai(in m^3/sec):
 2  Q=1;
 3  // Diameter of pipe(in m):
 4  D=0.25;
 5  // Density of air (in kg/m^3):
 6  d1=1.23;
 7  // Acceleration due to gravity(in m/s^2):
 8  g=9.8;
 9  // Density of water(in kg/m^3):
10  d2=999;
11  // Maxmum range of manometer(in m):
12  h=0.3;
13  // Kinematic viscosity(in m^2/s):
14  v=1.46*10^-5;
```

Scilab code AP 32 data

```
 1  // Nozzle exit diameter(in mm):
 2  D=25;
 3  //N/R1 value value:
 4  N_R=3;
 5  //AR value:
 6  A_R=2;
 7  // Static head available from the main(in m):
 8  z0=1.5;
 9  // Acceleration due to gravity(in m/sec^2):
10  g=9.8;
11  // Value of Cp:
12  Cp=0.45;
```

Scilab code AP 33 data

```
 1  // Length of copper wire(in ft):
 2  L=10;
 3  // Inner diameer of pipe(in inches):
 4  D=1.5;
 5  // Dischare(in ft^3/sec):
 6  Q=0.566;
```

```
7  //Level of reservoir above pipe centreline(inn feet)
       :
8  h=85.1;
9  //Kinematic viscosity at 70 F(in ft^2/s):
10 v=1.05*10^-5;
11 //Acceleration due to gravity(in ft/sec^2):
12 g=32.2;
```

Scilab code AP 34 data

```
1  //Length of Al tubing (in ft):
2  L=500;
3  //Volume flow rate of pump output(in gpm):
4  Q=1500;
5  //Discharge pressure(in psig):
6  p1=65;
7  //Sprinkler pressure(in psig):
8  p2=30;
9  //Kinematic viscosity(in ft^2/sec):
10 v=1.21*10^-5;
11 //Density(in slug/ft^3):
12 p=1.94;
```

Scilab code AP 35 data

```
1  //Height of standpipe(in ft):
2  l=80;
3  //Length of longest pipe(in ft):
4  L=600;
5  //Diameter of pipe(in inches):
6  D=4;
7  //Friction factor:
8  f=0.031;
9  //Acceleration due to gravityin ft/sec^2):
10 g=32.2;
```

Scilab code AP 36 data

```
1  // Flow  rate  of  crude  oil ( in  bbl ) :
2  Q=1.6*10^6;
3  // Inside  diamete  of  pipe ( i  inches ) :
4  D=48;
5  // Maximum  allowable  pressure ( in  psi ) :
6  p2=1200;
7  // Minimum  pressure  required  to  keep  gases  dissolves (
     in  psi ) :
8  p1=50;
9  // Specific  gravity  of  crde  oil :
10  SG=0.93;
11  // Viscosity  at  140  F( in  lbf−s / ft ^2 ) :
12  u=3.5*10^-4;
13  // Efficincy  of  pump :
14  Effp=0.85;
15  // Density ( in  slug / ft ^3 ) :
16  d=1.94;
17  // Viscosity  ( in  lbf−sec ) :
18  u=3.5*10^-4;
19  // Friction  factor :
20  f=0.017;
```

Scilab code AP 37 data

```
1  // Volme  flow  rate  of  water ( in  m^3/ sec ) :
2  Q=0.0084;
3  // Length  of  horizontal  pipe ( in  m) :
4  L=100;
5  // Diameter  of  pipe ( in  m) :
6  D=0.075;
7  // Density  of  water ( in  kg /m^3 ) :
8  d=999;
9  // Friction  factor :
10  f=0.017;
11  // Minor  lossses  coefficient :
12  K=0.5;
13  // Viscosity ( in  kg /m−s ) :
14  u=10^-3;
```

```
15  //Acceleration due to gravity(in /sec^2):
16  g=9.8;
```

Scilab code AP 38 data

```
1  //Flow rate through capilarry viscometer(in mm^3/sec
       ):
2  Q=880;
3  //Tube length(in m):
4  L=1;
5  //Tube diameter(in mm):
6  D=0.5;
7  //Pressure drop(in kPa):
8  p=1000;
9  //Density of oil(in kg/m^3):
10 d=999;
```

Scilab code AP 39 data

```
1  //temperature fo operation(in F):
2  T=210;
3  //Diameter of te bearing(in inches):
4  D=3;
5  //Diametral clearance(in inches):
6  a=0.0025;
7  //Length of shaft(in inhes):
8  L=1.25;
9  //Speed of rotation of the shaft(in rpm):
10 N=3600;
11 //Viscosity of the oil(in lbf-s/ft^2):
12 u=2.01*10^-4;
13 //Specific gravity of SAE 10W:
14 SG=0.92;
15 //Density of water (in slug/ft^3)
16 p=1.94;
```

Scilab code AP 40 data

```
1  // Operation pressure of hydraulic system(in kPa):
2  p1 =20000;
3  // Operation temperature of hydraulic system(in C):
4  T=55;
5  // Piston diameter(in mm):
6  D=25;
7  // Viscosity of SAE 10W at 55C(in kg/(m-s):
8  u=0.018;
9  // Mean radial clearance of a cylinder(in mm):
10  a=0.005;
11  // Gauge pressure on lower pressure side of piston(in
       kPa):
12  p2=1000;
13  // Lenth of piston(in mm):
14  L=15;
15  // Denity of water(in kg/m^3):
16  dw=1000;
```

Scilab code AP 41 data

```
1  // Efficinc of pump:
2  Effp=0.8;
3  // Design specific speed(in rpm):
4  Nscu1=2000;
5  // Impeller diameter(in inches):
6  D1=8;
7  // Opertion sped at esign point flow condition(in rpm
      ):
8  N1=1170;
9  // Flow rate at design point flow condition(in gpm):
10  Q1=300;
11  // Density of water (in slug/ft^3):
12  d1=1.94;
13  // Acceleration due to gravity(in ft^2/sec):
14  g=32.2;
15  // Working speed 2(in rpm):
16  N2=1750;
```

Scilab code AP 42 data

```
1  //Width of the prototype(in ft):
2  wp=8;
3  //Frontal area of the prototype(in ft^2):
4  Ap=84;
5  //Model Scale:
6  S=1/16;
7  //Density of air(in kg/m^3):
8  d=1.23;
9  //Air speed in wind tunnel(in m/sec):
10 V=[18 21.8 26 30.1 35 38.5 40.9 44.1 46.7];
11 //Drag force(in N):
12 Fd=[3.1 4.41 6.09 7.97 10.7 12.9 14.7 16.9 18.9];
13 //Kinematic viscosity(in m^2/sec):
14 v=1.46*10^-5;
15 //Density of air(in kg/m^3):
16 d=1.23;
17 //Speed of prototype(in km/hr):\
18 Vp=100;
```

Scilab code AP 43 data

```
1  //Diameter of the prototype(in ft):
2  Dp=1;
3  //Speed of towing of prototype(in knots):
4  Vp=5;
5  //Diameter of model(in inches):
6  Dm=6;
7  //Drag for model at test condition(in lbf):
8  Fm=5.58;
9  //Density of seawater at 5 C for prototype(in slug/
      ft^3):
10 dp=1.99;
11 //Kinematic viscosity at 5 C for prototype(in ft^2/
      sec):
12 vp=1.69*10^-5;
13 //Density of air at STP for model(in slug/ft^3):
```

```
14  dm =0.00238;
15  //Kinematic  viscosity  of  air  at  STP  for  model(in  ft
        ^2/sec):
16  vm =1.57*10^-4;
```

Scilab code AP 44 data

```
1  //Depth  to  which  water  is  filled(in  m):
2  h=3;
3  //Length  of  pipe(in  m):
4  L=6;
5  //Diameter  of  pipe  (in  mm):
6  D=150;
7  //Acceleration  due  to  gravity(in  m/sec^2):
8  g=9.81;
```

Scilab code AP 45 data

```
1  //Area  of  cross  section  of  the  nozzle(in  in^2):
2  A4=0.864;
3  //Capacity  of  heater(in  kW):
4  Q=10
5  //Acceleration  due  to  gravity(in  ft/sec^2):
6  g=32.2;
7  //Water  level  in  reservoir  above  datum  line(in  ft):
8  z3=10;
9  //Density  of  water(In  slug/ft^3):
10  d=1.94;
```

Scilab code AP 46 data

```
1  //Speed  of  plane(in  km/hr):
2  V=150;
3  //Speed  at  point  B  relative  to  the  wing(in  m/sec):
4  Vb=60;
5  //Density  of  air(in  kg/m^3):
6  da=1.23;
```

```
7  //Atmospheris pressure(in N/m^2):
8  pa =1.01*10^5;
9  //At 1000m,
10 //p/pSL:
11 P1 =0.8870;
12 //d/dSL:
13 D1 =0.9075;
```

Scilab code AP 47 data

```
1  //Depth of water at the upstream(on feet):
2  Du =1.5;
3  //Depth of water at the vena contracta downstream
       from the gate(in inches):
4  Dd =2;
5  //Acceleration due to gravity(in ft/sec^2):
6  g =32.2;
```

Scilab code AP 48 data

```
1  //Length of tube above surface(in m):
2  l =1;
3  //Depth of exit below water surface(in m):
4  z =7;
5  //Acceleration due to gravity(in m/sec^2):
6  g =9.81;
7  //Density of water(in kg/m^3):
8  d =999;
9  //Atmospheric pressure(in N/m^2):
10 p1 =1.01*10^5;
```

Scilab code AP 49 data

```
1  //Area of nozzle at input(in m^2):
2  Ai =0.1;
3  //Area of nozzle at exit(in m^2):
4  Ae =0.02;
```

```
5  //Outlet velocity of flow(in m/sec):
6  V2=50;
7  //Density of air(in kg/m^3):
8  da=1.23;
```

Scilab code AP 50 data

```
1   //Pressure diference(in mm of mecury):
2   p=30;
3   //Density of water(in kg/m^3):
4   dw=1000;
5   //Aceleration due to gravity(in m/sec^2):
6   g=9.81;
7   //Density of air(in kg/m^3):
8   da=1.23;
9   //Specific gravity of mercury:
10  SG=13.6;
```

Scilab code AP 51 data

```
1   //Depth of the duct(in m):
2   d=0.3;
3   //Width of the duct(in m):
4   w=0.1;
5   //Inner radius of the bend(in m):
6   r=0.25;
7   //Pressure difference between the taps(in mm of Hg):
8   p=40;
9   //Density of water(in kg/m^3):
10  dw=999;
11  //Acceleration due to gravity(in m/sec^2):
12  g=9.8;
13  //Density of air(in kg/m^3):
14  da=1.23;
```

Scilab code AP 52 data

```
1  // Thickness  of  water  film (in mm):
2  h=1;
3  // Width  of  surface (in m):
4  b=1;
5  // Angle  of  inclination  of  surface :
6  theta=15;
7  // Density  of  water (in  kg/m^3):
8  d=999;
9  // Acceleration  du  to  gravity (in m/sec^2):
10 g=9.81;
11 // Viscosity (kg/m-s):
12 u=10^-3;
```

Scilab code AP 53 data

```
1  // Value  of  A(in  sec^-1):
2  A=0.3;
```

Scilab code AP 54 data

```
1  // Value  of  (in mm/sec):
2  U=4;
3  // Value  of  h(in mm):
4  h=4;
5  // Tme  at  which  to  find  position (in  sec):
6  t=1.5;
```

Scilab code AP 55 data

```
1  // Distance  f  piston  from  closed  end  of  the  cylinder
       at  the  give  instant (in m):
2  L=0.15;
3  // Density  of  gas (in  kg/m^3):
4  d=18;
5  // Velocity  of  piston (in m/sec):
6  V=12;
```

Scilab code AP 56 data

109

```
1  //Volume of tak(in m^3):
2  V=0.1;
3  //Temperature of line and tank(in K):
4  T=293;
5  //Initial tank gauge pressure(in N/m^2):
6  p1=1*10^5;
7  //Absolute line pressure(in N/m^2):
8  p=2*10^6;
9  //Rate of rise of temperature after opening of the
      valve(in C/sec):
10 r=0.05;
11 //Atmospheric pressure(in N/m^2):
12 patm=1.01*10^5;
13 //Gas Constant(in N-m/(kg-K)):
14 R=287;
15 //Value of cv(in N-m/kg-K):
16 cv=717;
```

Scilab code AP 57 data

```
1  //Pressure at entry(in psia):
2  p1=14.7;
3  //Temperature at entry(in F):
4  T1=70;
5  //Pressure at exit(in psia):
6  p2=50;
7  //Temprature a exit(in F):
8  T2=100;
9  //Cross sectional area of the pipe at exit(in ft^2):
10 A2=1;
11 //Mass flow rate(in lbf/sec):
12 m=20;
13 //Power input to the compressor(in hp):
14 Ws=-600;
15 //Value of cp(in Btu/lbm-R):
16 cp=0.24;
17 //Value of gas constant(in ft-lbf/(lbm-R))
18 R=53.3;
```

Scilab code AP 58 data

```
1  //Inlet gauge pressure(in kPa):
2  p=20;
3  //Volume flow rate of water through the sprinkler(in
        l/min):
4  Q=7.5;
5  //Speed of rotstion of sprinkler(in rpm):
6  w=30;
7  //Diameter of jet f sprinkle(in mm):
8  D=4;
9  //Radius of sprinkler(in mm):
10 R=150;
11 //Supply pressure to sprinkler(in kPa):
12 p=20;
13 //Angle at which jet is sprayed wrt horizontal:
14 alpha=30;
15 //Density of water(in kg/m^):
16 d=999;
```

Scilab code AP 59 data

```
1  //Initial mass of th rocket(in kg):
2  M0=400;
3  //Rate of fuel consumption(in kg/sec):
4  me=5;
5  //Exhaust velocity(in m/sec):
6  Ve=3500;
7  //Acceleration due to gravity(in m/sec^2):
8  g=9.81;
9  //Time after which velocity is to be calculated(in
        sec):
10 t=10;
```

Scilab code AP 60 data

```
1  //Mass of vane and cart(in kg):
2  M=75;
3  //Turning angle of vane:
4  theta=60;
5  //Speed of water leaving nozzle horizontally(in m/
       sec):
6  V=35;
7  //Exit area of nozzle(in m^):
8  A=0.003;
9  //Density of water(in kg/m^3):
10 d=999;
```

Scilab code AP 61 data

```
1  //Vane turning angle:
2  theta=60;
3  //Speed of vane(in m/sec):
4  U=10;
5  //Area of nozzle(in m2):
6  A=0.003;
7  //Flow velocity of water(in m/sec):
8  V=30;
9  //Density of water(in kg/m^3):
10 d=999;
```

Scilab code AP 62 data

```
1  //Nozzle inlet diameter(in inchess):
2  D1=3;
3  //Nozzle exit diameter(in inches):
4  D2=1;
5  //Desired volume flow rate(in ft^3/sec):
6  Q=0.7;
7  //Density of water(in slug/ft^3):
8  d=1.94;
```

Scilab code AP 63 data

```
1  // Velocity  of  conveyor  belt (in  ft/sec):
2  Vbelt =3;
3  // Velocity  of  sand  alling  onto  belt (in  ft/sec):
4  Vsand =5;
5  // Flow  rate (in  lbm/sec):
6  m =500;
```

Scilab code AP 64 data

```
1   // Pressure  at  inlet  tothe  elbow (in  N/m^2):
2   p1 =2.21*10^5;
3   // Area  of  crosssection (in  m^2):
4   A1 =0.01;
5   // Velocity  at  secton  2(in  m/sec):
6   V2 =16;
7   // Area  of  cross  section  of  section  2(in  m^2):
8   A2 =0.0025;
9   // Atmospheric  pressure (in  kPa):
10  patm =1.012*10^5;
```

Scilab code AP 65 data

```
1   // Diameter  of  channel (in  m):
2   D1 =1.5;
3   // Velcity  of  flow  in  channel (in  m/sec):
4   V1 =0.2;
5   // Diameter  at  section  2(in  m):
6   D2 =0.0563;
7   // Velocity  a  section  2(in  m/sec):
8   V2 =5.33;
9   // Density  of  water (in  kg/m^3):
10  d =999;
11  // Acceleration  due  to  gravity (in  m/sec2):
12  g =9.81;
```

Scilab code AP 66 data

```
1  // Height of the container (in ft):
2  l=2;
3  // Area of cross section (in ft^2):
4  A=1;
5  // Weight of container (in lbf):
6  W=5;
7  // Water depth (in ft):
8  h=1.9;
9  // Area of opening 1(in ft^2):
10 A1=0.1;
11 // Velocity at opening 1(in ft/sec):
12 V1=-5;
13 // Area of opening 2(in ft^2):
14 A2=0.1;
15 // Area of opening 1(in ft^2):
16 A3=0.1;
17 // Density of water(in slug/f^3):
18 d2=1.94;
```

Scilab code AP 67 data

```
1  // Velocity of water leaving the nozle(in m/sec):
2  V=15;
3  // Area of nozzle(in m^2):
4  A=0.01;
5  // Density of water(in kg/m^3):
6  d=999;
```

Scilab code AP 68 data

```
1  // Volume of tank(in m^3):
2  V=0.05;
3  // Pressure of air(In kPa):
4  p=800;
5  // Temperature of tank(in C):
6  T=15;
7  // Velocity of leavig air(in m/sec):
8  v=311;
```

```
 9 //Density of air (in kg/m^3):
10 d=6.13;
11 //Area ofvalve exit(in mm^2):
12 A=65;
```

Scilab code AP 69 data

```
1 //Flow velocity ahead of the plate(in m/sec):
2 U=30;
3 //Boundary layer tckness at location d(in mm):
4 t=5;
5 //Density of fluid air(in k/m^3):
6 d=1.24;
7 //Plate wdth perpendicular to the plate(in m):
8 w=0.6;
```

Scilab code AP 70 data

```
 1 //Area of 1 (in ft^2):
 2 A1=0.2;
 3 //Area of 2 (in ft^2):
 4 A2=0.5;
 5 //Area of 3 (in ft^2):
 6 A3=0.4;
 7 //Area of 4 (in ft^2):
 8 A4=0.4;
 9 //Density of water (in slug/ft^3):
10 d=1.94;
11 //Mass flow rate out of section 3(in slug/sec):
12 m3=3.88;
13 //Volme flow rate in section 4 (in ft^3/sec):
14 Q4=1;
15 //Velocity at 1(in ft/sec):
16 V1=10;
```

Scilab code AP 71 data

```
1  //Width of gate(in m):
2  w=5;
3  //Depth of water(in m):
4  D=4;
5  //Density of water(in kg/m^3);
6  d=999;
7  //Accelration deto gravity(in m/sec^2):
8  g=9.81;
9  //Value of a (in m):
10 a=4;
11 //Point where force acts(in m):
12 l=5;
```

Scilab code AP 72 data

```
1  //Pressure apllied on the door(in psfg):
2  p0=100;
3  //Length of door(in feet):
4  L=3;
5  //Breadth of the door(in feet):
6  b=2;
7  //Density of liqiuid(in lbf/ft^3):
8  d=100;
```

Scilab code AP 73 data

```
1  //Length of gate(in m):
2  L=4;
3  //Width of gate(in m):
4  w=5;
5  //Depth of gate under water(in m):
6  D=2;
7  //Density of water(in kg/m^3:
8  d=999;
9  //Acceleration due to gravity(in m/sec^2):
10 g=9.81;
11 //Angle of gate with horizontal:
12 theta=30;
```

Scilab code AP 74 data

```
1  //Elevation of Denver(in ft):
2  z1=5280;
3  //Pressure at Denver(in mm of Hg):
4  p1=24.8;
5  //Temperature at Denver(in F):
6  T1=80;
7  //Elevation at Vail Pass(in ft):
8  z2=10600;
9  //Temperature at Vsil Pass(in F):
10 T2=62;
11 //Value of R in ft-lbf/lbm-R):
12 R=53.3;
13 //Acceleration due togravity(in ft/sec^2):
14 g=32.2;
15 //Value of adiabatic constant:
16 k=1.4;
```

Scilab code AP 75 data

```
1  //Acceleration due to gravity(in ft/sec^2):
2  g=32.2;
3  //Specific gravity of mercury:
4  SGm=13.6;
5  //Specific gravity of oil:
6  SGo=0.88;
7  //Specific gravity of water:
8  SGw=1;
9  //Density of water(in slug/ft^3):
10 d=1.94;
11 //Heights of liquid in various tubes(in inches):
12 d1=10;
13 d2=3;
14 d3=4;
15 d4=5;
```

```
16  d5 =8;
```

Scilab code AP 76 data

```
1  //Surface  tension  of  water(in  mN/m):
2  STw=72.8*10^-3;
3  //Surface  Tension  of  mercury(in  mN/m):
4  STm=375*10^-3;
5  //Contact  angle  for  water:
6  thetaw=0;
7  //COntact  angle  for  mercury:
8  thetam=140;
9  //Density  of  water(in  kg/m^3):
10  dw=1;
11  //Density  of  mercury(in  kg/m^3):
12  dm=13.6;
13  //Acceleration  de  to  gravity(in  m/sec):
14  g=9.81;
```

Scilab code AP 77 data

```
1  //Liquid  Viscosity(in  cp):
2  u=0.65;
3  //Specific  gravity:
4  SG=0.88;
5  //Density  of  water(in  slug/ft^3):
6  d=1.94;
7  //Velocity  with  which  plate  is  moved(in  m/sec):
8  U=0.3;
9  //Distance  between  the  plates(in  mm):
10  D=0.3;
```

Scilab code AP 78 data

```
1  //Mass  of  ball(in  kg):
2  m=0.2;
3  //Height  fom  which  ball  is  dropped(in  m):
```

```
4  y0 =500;
5  // Value of k:
6  k=2*10^-4;
7  // Accleration due to gravity(in m/sec^2):
8  g=9.81;
```

Scilab code AP 79 data

```
1  // Mass of oxygen present (in kg):
2  m=0.95;
3  // Initial temperatur (in K):
4  T1=300;
5  // Final temperature of oxygen (in K):
6  T2=900;
7  // Pressure of oxygen (in kPa):
8  p=150;
9  // Specific heat at constant pressure (in J/kg-K):
10 cp=909.4;
```