Scilab Textbook Companion for Physics by R. Resnick, D. Halliday, K. S. Krane[1]

Created by
Joshi Prajakta Sanjay
3rd year,Engineering
Mechanical Engineering
Pune University
College Teacher
Gautam Chandekar
Cross-Checked by
K. V. P. Pradeep

July 31, 2019

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

Book Description

Title: Physics

Author: R. Resnick, D. Halliday, K. S. Krane

Publisher: John Wiley And Sons , New Delhi

Edition: 5

Year: 2009

ISBN: 978-81-265-1088-7

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

Chapter 1

MEASUREMENT

Scilab code Exa 1.1 C1P1

```
1  clear
2   clc
3  //To find speed in meters per second
4  //to find volume in cubic centimeters
5
6  // GIVEN::
7
8  //speed
9  speed1 = 55 //miles per hour
10 //volume of gasoline
11 volume1 = 16 //gallons
12
13 // SOLUTION:
14
15 //speed in meters per second
16 // since 1 mile = 1609 meters and 1 hour = 3600
     seconds
17 speed = (55)*(1609/1)*(1/3600) //miles per hour
18
19 //volume of gasoline in cubic centimeters
20 // since 1 gallon = 231 cubic inches and 1 inch =
```

```
    2.54 centimeter
21 volume = 16*(231/1)*(2.54/1)^3 // cubic centimeters
22 speed = round(speed)
23 printf ("\n\n Speed in meters per second speed =\n\n
    %2i m/s" ,speed);
24 printf ("\n\n Volume of gasoline in cubic
    centimeters volume =\n\n %.1e cm^3" ,volume);
```

Scilab code Exa 1.2 C1P2

```
1
2 clear
3  clc
4 //To find conversion factor between light year ang
    meters
5 //to find distance to the star proxima centuri
6
7 // GIVEN::
8
9 //distance
10 d = 4*10^16 //in light years
11 // velocity of light
12 v = 3.00*10^8 // m/s
13
14 // SOLUTION:
15
16 //conversion factor
17 // first finding conversion between 1 year and
    seconds
18 y = (1)*(365.25/1)*(24/1)*(60/1)*(60/1) // seconds
19 //now finding conversion between light year ang
    meters
20 light_year = (y*v) // meters
21
22 //to find distance to the star proxima centuri
```

```
23 distance = (d)*(1/light_year)// light years
24 light_year = nearfloat("pred",9.48e15)
25 printf ("\n\n Conversion between 1 year and seconds
     y = \n\n %.2e seconds" ,y);
26 printf ("\n\n Conversion between light year ang
     meters 1 light year =\n\n %.2e m" ,light_year);
27 printf ("\n\n Distance to the star proxima centuri
     distance =\n\n %.1f light years" ,distance);
```

Scilab code Exa 1.3 C1P3

```
1
2 clear
3  clc
4 //to find fractional and percentage uncertainty in
     your weight
5 //to find fractional and percentage uncertainty in
     cat's weight
6
7 // GIVEN::
8
9 //your weight
10 w1 = 119 //in lbs
11 // your and cat's combined weight
12 w2 = 128 // in lbs
13
14 // SOLUTION:
15
16 //fractional uncertainty in your weight
17 u1 = (1/119)
18 // percentage uncertainty in your weight
19 u2 = u1*100 //percentage
20 //fractional uncertainty in cat's weight
21 u3 = (1/9)
22 //percentage uncertainty in cat's weight
```

```
23 u4 = u3*100 //percentage
24  printf ("\n\n Fractional uncertainty in your weight
       u1 =\n\n %.3f" ,u1);
25  printf ("\n\n Percentage uncertainty in your weight
       u2 =\n\n %.1f percent" ,u2);
26  printf ("\n\n Fractional uncertainty in cats weight
       =\n\n %.2f" ,u3);
27  printf ("\n\n Percentage uncertainty in cats weight
       =\n\n %1i percent" ,u4);
```

Scilab code Exa 1.5 C1P5

```
1
2
3 clear
4 clc
5 //to find value of plank time
6 // GIVEN::
7
8 //speed of light
9 c = 3.00e8  //m/s
10 //Newton's gravitational constant
11 G = 6.67e-11 // m^3/s^2.Kg
12 //plank's constant
13 h = 6.63e-34// Kg.m^2/s
14
15 // SOLUTION:
16
17 //plank time
18 tp = sqrt((G*h)/c^5)// seconds
19 //answer in the book is slightly different which is
     printing mistake
20  printf ("\n\n Plank time tp =\n\n %.2e seconds" ,tp
     );
```

Chapter 2

MOTION IN ONE DIMENSION

Scilab code Exa 2.1 C2P1

```
1
2 clear
3  clc
4 //to find distance travelled to the north and east
     does the airplane travel
5
6 // GIVEN::
7
8 //distance travelled by airplane
9 d = 209// in km
10 //angle made by airplane east of due north that is
     angle made with y direction
11 theta = 22.5// in degrees
12
13 // SOLUTION:
14
15 //angle made by airplane with x direction
16 fi = 90-theta//in degrees
17 // distance travelled to the north
```

```
18 dy = d*sind(fi)
19 //distance travelled to the east
20 dx = d*cosd(fi)
21 printf ("\n\n Angle made by airplane with x
      direction fi =\n\n %.1f degrees",fi);
22 printf ("\n\n Distance travelled to the north by
      airplane dx =\n\n %.1f km",dx);
23 printf ("\n\n Distance travelled to the east by
      airplane dy  =\n\n %3i km",dy);
```

Scilab code Exa 2.2 C2P2

```
1
2 clear
3  clc
4 //to find magnitude and direction of vector
      indicating location of car
5
6 // GIVEN::
7
8 //distance travelled due east on a level of road
9 //s is represented as ax+by.since b has no x
      component and a has no y componebt we can write
10 Sx = 32// in km
11 //distance travelled before stopping after taking
      turn due north
12 Sy = 47// in km
13
14
15 // SOLUTION:
16
17 //magnitude of distance travelled
18 x = sqrt(Sx^2 + Sy^2)//in meters
19 //direction of travelling
20 fi = atand(Sy/Sx)//in degrees
```

```
21 g = Sy/Sx
22 x = round(x)
23 fi = round(fi)
24 printf ("\n\n Distance travelled due east on a level
       of road Sx =\n\n %2i km",Sx);
25 printf ("\n\n Distance travelled before stopping
      after taking turn due north Sy =\n\n %2i km",Sy);
26 printf ("\n\n Magnitude ofdistance travelled by
      automobile x =\n\n %2i km",x);
27 printf ("\n\n Value of tanfi =\n\n %.2f ",g);
28 printf ("\n\n Direction of automobile travelling fi
      =\n\n %2i degrees north of east",fi);
```

Scilab code Exa 2.3 C2P3

```
1
2 clear
3  clc
4 //to find magnitude and direction of resultant of a
      and b and c vector
5
6 // GIVEN::
7
8 //coefficient in x direction for vector a
9 ax = 4.3
10 //coefficient in y direction for vector a
11 ay = -1.7
12 //coefficient in x direction for vector b
13 bx = -2.9
14 //coefficient in y direction for vector b
15 by = 2.2
16 //coefficient in x direction for vector c
17 cx = 0
18 //coefficient in y direction for vector c
19 cy = -3.6
```

```
20 //we can write a,b and c in vector form
21 a = [4.3 -1.7]
22 b = [-2.9 2.2]
23 c = [0 -3.6]
24
25 // SOLUTION:
26
27 //coefficient in x direction for resultant vector
28 sx = ax + bx + cx
29 //coefficient in y direction for resultant vector
30 sy = ay + by + cy
31 //direction of resultant vector
32 fi = atand(sy/sx)+360
33 printf ("\n\n Coefficient of resultant vector in x
     direction sx = \n\n %.1f",sx);
34 printf ("\n\n Coefficient of resultant vector in y
     direction sy =\n\n %.1f",sy);
35 printf ("\n\n Resultant vector s =\n\n %.1fi + %.1fj
     ',sx,sy);
36 printf ("\n\n Direction of resultant vector with
     positive x axis measured counterclockwise fi =\n\
     n %3i degrees",fi);
```

Scilab code Exa 2.4 C2P4

```
1
2 clear
3  clc
4 //to find magnitude of position,velocity and
     acceleration
5
6 // GIVEN::
7 //time
8 t = 3//in seconds
9 //coefficients
```

```
A = 1.00//in m/s^2
B = -32.0//in m/s
C = 5.0//in m/s^2
D = 12.0//in m


// SOLUTION:

//for position vector
//coefficient in x direction for resultant vector
rx = A*t^3 + B*t
//coefficient in y direction for resultant vector
ry = C*t^2 + D

//for velocity vector
//coefficient in x direction for resultant vector
//as v = dx/dt therefore differentiating rx and ry w.r.t t
vx = 3*A*t^2 + B
//coefficient in y direction for resultant vector
vy =2* C*t

//for acceleration vector
//as a = dv/dt therefore differentiating rx and ry again w.r.t t
//coefficient in x direction for resultant vector
ax = 6*A*t
//coefficient in y direction for resultant vector
ay = 2*C
printf ("\n\n Position vector r =\n\n %1i m i + %1i m j",rx,ry);
printf ("\n\n Velocity vector v =\n\n %1i m/s i + %1i m/s j",vx,vy);
printf ("\n\n Acceleration vector a =\n\n %1i m/s^2 i + %1i m/s^2 j",ax,ay);
```

Scilab code Exa 2.5 C2P5

```
1
2 clear
3  clc
4 //to find average velocity of car
5
6 // GIVEN::
7
8 //distance travelling by car
9 d1 = 5.2//in mi
10 //distance travelled while walking
11 d2 = 1.2//in mi
12 //time required to reach to gas station while
      walking
13 t1 = 27//in min
14 //speed of car
15 v = 43//in mi/h
16
17 // SOLUTION:
18
19 //net displacement
20 delta_x = d1 + d2//in mi
21 //speed of car in mi/minutes
22 v1 = v/60//in mi/minutes
23 //total elapsed time
24 delta_t1 = (d1/v1) + t1//in min
25 //total elapsed time in h
26 delta_t = delta_t1/60//in h
27 //average velocity
28 //applying kinematic equations
29 Vav_x = delta_x/delta_t//in mi/h
30 printf ("\n\n Net displacement delta_x =\n\n %.1f mi
      ",delta_x);
```

```
31 printf ("\n\n Total elapsed time delta_t =\n\n %.2f
     h",delta_t);
32 printf ("\n\n Average velocity of car required Vav_x
     =\n\n %.1f mi/h",Vav_x);
```

Scilab code Exa 2.6 C2P6

```
1
2 clear
3  clc
4 //to find average velocity for interval AD and DF
5 //to find slope of position curve at the points B
     and F and compare it with the value in velocity
     curve
6 //to find average acceleration in the interval AD
     and AF
7 //to find slope of velocity curve at the points D
     and compare it with the value in acceleration
     curve
8
9 // GIVEN::
10
11 //distance travelling by the point D has come
12 xD = 5.0//in m
13 //distance travelling by the point A has come
14 xA = 1.0//in m
15 //distance travelling by the point F has come
16 xF = 1.4//in m
17 //time elapsed by the point D has come
18 tD = 2.5//in seconds
19 //time elapsed by the point A has come
20 tA = 0.0//in seconds
21 //time elapsed by the point F has come
22 tF = 4.0//in seconds
23 //velocity at point D
```

```
vD = 0.0//in m/s
//velocity at point A
vA = 4.0//in m/s
//velocity at point F
vF = -6.2//in m/s



// SOLUTION:

//average velocity for the interval AD
//applying kinematic equations
Vav_x = (xD-xA)/(tD-tA)
//average velocity for the interval DF
//applying kinematic equations
Vavx = (xF-xD)/(tF-tD)
//slope of position curve at the point B
slope_B = (4.5-2.8)/(1.5-0.5)//refer to the graph
   2.6(b) given in the book on page no. 25
//slope of position curve at the point F
slope_F = (1.4-4.5)/(4.0-3.5)//refer to the graph
   2.6(b) given in the book on page no. 25
//average acceleration in the interval AD
//applying kinematic equations
Aav_x = (vD-vA)/(tD-tA)//in m/s^2
//average acceleration in the interval AF
//applying kinematic equations
Aavx = ((vF-vA)/(tF-tA))//in m/s^2
Aavx = nearfloat("pred",-2.6)
//slope of velocity curve at the point D
slope_D = (-0.9-0.9)/(3.0-2.0)//refer to the graph
   2.6(c) given in the book on page no. 25

printf ("\n\n Average velocity for the interval AD
   Vav_x=\n\n %.1f m/s",Vav_x);
printf ("\n\n Average velocity for the interval DF
   V_avx =\n\n %.1f m/s",Vavx);
printf ("\n\n Slope of position curve at the point B
```

```
      slpoe_B=\n\n %.1f m/s",slope_B);
57 printf ("\n\n Slope of position curve at the point F
    =\n\n %.1f m/s",slope_F);
58 //refer velocity time graph 2.6(c) given in the book
    on the page no.25
59 printf ("\n\n From velocity curve value of velocity
   at point B is \n\n 1.7m/s");
60 printf ("\n\n From velocity curve value of velocity
   at point Bis \n\n -6.2m/s");
61 printf ("\n\n Average acceleration for the interval
   AD Aav_x =\n\n %.1f m/s^2",Aav_x);
62 printf ("\n\n Average acceleration for the interval
   AF Aavx =\n\n %.1f m/s^2",Aavx);
63 printf ("\n\n Slope of velocity curve at the point D
    slope_D =\n\n %.1f m/s^2",slope_D);
64 //refer velocity time graph 2.6(d) given in the book
    on the page no.25
65 printf ("\n\n From acceleration curve value of
   acceleration at point D is \n\n -1.8m/s^2");
```

Scilab code Exa 2.7 C2P7

```
1
2 clear
3  clc
4 //to find acceleration of partical
5 //to find velocity of partical when it leaves the
    tube
6
7 // GIVEN::
8
9 //length of the tube
10 x = 2.0//in m
11 //velocity of partical when it enters in the tube i.
    e.at t=0s
```

```
12 v0x = 9.5*10^5//in m/s
13 //time when the partical emerges out of the tube
14 t = 8.0*10^-7//in m/s
15
16 // SOLUTION:
17
18 //acceleration of the partical
19 //applying kinematic equations
20 ax = (x-(v0x*t))/(0.5*t^2)//in m/s^2
21 //velocity of the partical when it leaves the tube
22 //applying kinematic equations
23 vx = v0x + (ax*t)
24
25 printf ("\n\n Acceleration of the partical ax =\n\n
     %.1e m/s^2",ax);
26 printf ("\n\n Velocity of the partical when it
     leaves the tube vx =\n\n %.1e m/s",vx);
```

Scilab code Exa 2.8 C2P8

```
1
2 clear
3  clc
4 //to find timw elapsed
5 //to find acceleration
6 //after apply brakes with constant acceleration time
      required to stop vehicle
7 //additionl distance covered after vehicle has
     stopped
8
9 // GIVEN::
10
11 //linitial velocity at t=0
12 v0x = 23.6//in m/s
13 //final velocity
```

```
14 vx = 12.5//in m/s
15 //distance travelled
16 delta_x = 105//in m
17 //velocity after vehicle has stopped
18 vxf = 0//in m/s
19
20 // SOLUTION:
21
22 //average velocity
23 //applying kinematic equations
24 vav_x = (v0x + vx)/2//in m/s
25 //time elapsed
26 time = delta_x/vav_x//in seconds
27 //acceleration
28 //applying kinematic equations
29 ax = (vx-v0x)/time//in m/s^2
30 //time required to stop vehicle
31 //applying kinematic equations
32 t = (vxf-v0x)/ax//in s
33 //total distance covered by vehicle
34 //applying kinematic equations
35 x = (v0x*t)+(0.5*ax*t^2)//in m/s^2
36 //additional distance travelled by vehicle
37 x_final = x - delta_x//in m
38 x = round(x)
39 x_final = round(x_final)
40 time = nearfloat("pred",5.81)
41
42 printf ("\n\n Average velocity vav_x =\n\n %.2f m/s"
     ,vav_x)
43 printf ("\n\n Time elapsed time =\n\n %.2f s",time);
44 printf ("\n\n Acceleration of vehicle ax =\n\n %.2f
     m/s^2",ax);
45 printf ("\n\n After apply brakes with constant
     acceleration time required to stop vehicle t =\n\
     n %.1f s",t);
46 printf ("\n\n Total distance covered by vehicle x =\
     n\n %3i m",x);
```

```
47 printf ("\n\n Additionl distance covered after
      vehicle has stopped x_final =\n\n %1i m",x_final)
      ;
```

Scilab code Exa 2.9 C2P9

```
1
2 clear
3  clc
4 //to find position and acceleration after t=1,2,3,4s
      have elapsed
5
6
7 // GIVEN::
8
9 //linitial velocity due free fall of body
10 v0y = 0//in m/s
11 //acceleration due to gravity
12 g = 9.8//in m/s^2
13 //time elapsed
14 t1 = 1.0//in s
15 t2 = 2.0//in s
16 t3 = 3.0//in s
17 t4 = 4.0//in s
18
19
20 // SOLUTION:
21
22 //velocity v = -(g*t)
23 //since initial velocity is zero
24 v1 = (v0y*t1)-(g*t1)//in m/s
25 v2 = (v0y*t2)-(g*t2)//in m/s
26 v3 = (v0y*t3)-(g*t3)//in m/s
27 v4 = (v0y*t4)-(g*t4)//in m/s
28 //since body is moving vertically downwards s0,
```

```
    velocity has -ve sign
//distance travelled y = -(0.5*g*t^2)
y1 = (v0y*t1)-0.5*(g*t1^2)//in m
y2 = (v0y*t2)-0.5*(g*t2^2)//in m
y3 = (v0y*t3)-0.5*(g*t3^2)//in m
y4 = (v0y*t4)-0.5*(g*t4^2)//in m
// -ve sign indicates body is travelling in -ve y
    direction
printf ("\n\n Distance travelled after elapsed time
    t1 =\n\n %.1f m",y1);
printf ("\n\n Distance travelled after elapsed time
    t2 =\n\n %.1f m",y2);
printf ("\n\n Distance travelled after elapsed time
    t3 =\n\n %.1f m",y3);
printf ("\n\n Distance travelled after elapsed time
    t4 =\n\n %.1f m",y4);
printf ("\n\n Velocity after elapsed time t1 =\n\n %
    .1f m/s",v1);
printf ("\n\n Velocity after elapsed time t2 =\n\n %
    .1f m/s",v2);
printf ("\n\n Velocity after elapsed time t3 =\n\n %
    .1f m/s",v3);
printf ("\n\n Velocity after elapsed time t4 =\n\n %
    .1f m/s",v4);
```

Scilab code Exa 2.10 C2P10

```

clear
 clc
//to find time required to reach highest point
//to find distance travelled by the ball till the
    highest position is reached
//to find time at which ball will be 27m above the
    groung
```

```
7
8  // GIVEN::
9
10 //initial speed of the ball
11 v0y = 25.2//in m/s
12 //final speed of the ball
13 vy = 0//in m/s
14 //acceleration due to gravity
15 g = 9.8//in m/s^2
16 //for calculating time distance of ball above the
      ground
17 y1 = 27.0//in meters
18
19
20
21 // SOLUTION:
22 //time required to reach highest psition
23 //applying kinematic equations
24 t = (v0y-vy)/g//in seconds
25 //distance travelled by the ball till the highest
      position is reached
26 //applying kinematic equations
27 y = (v0y*t)-(1/2*g*t^2)//in meters
28 //time at which ball will be 27m above the groung
29 //solving quadratic equation
30 y1 = poly([y1 -(v0y) (1/2*g)],'t','coeff')
31 c = roots(y1)
32 t1 = c(1)
33 t2 = c(2)
34 //velocity of ball at t1
35 vy1 = v0y-(g*t1)//in m/s
36 //velocity of ball at t2
37 vy2 = v0y-(g*t2)//in m/s
38
39 printf ("\n\n Time required to reach highest psition
       t =\n\n %.2f s",t);
40 printf ("\n\n Distance travelled by the ball till
      the highest position is reached y =\n\n %.1f m",y
```

```
    );
41 printf ("\n\n Time at which ball will be 27m above
    the groung t1 =\n\n %.2f s",t1);
42 printf ("\n\n Time at which ball will be 27m above
    the groung t2 =\n\n %.2f s",t2);
43 printf ("\n\n Velocity of ball at t1 =\n\n %.1f m/s"
    ,vy1);
44 printf ("\n\n Velocity of ball at t2 =\n\n %.1f m/s"
    ,vy2);
```

Scilab code Exa 2.11 C2P11

```
1
2 clear
3  clc
4 //maximum distance travelled by the rocket above the
      water surface
5
6
7 // GIVEN::
8
9 //distance below the water surface
10 //this can be written as y-y0 = s =125
11 s = 125//in meters
12 //initial velocity of rocket
13 v0y = 0//in m/s
14 //acceleration due to gravity
15 g = 9.8//in m/s^2
16 //time required to reach the water surface
17 t = 2.15//in seconds
18 //velocity of rocket at highest position
19 v2 = 0//in m/s^2
20
21 // SOLUTION:
22 //acceleration of rocket in upward direction
```

```
23 //applying kinematic equations
24 ay = (2*s)/t^2//in m/s^2
25 //final velocity of the rocket at the surface of
     water
26 //applying kinematic equations
27 vy = v0y+(ay*t)//in m/s
28 //now taking v3 as initial velocity of rocketi.e.
     velocity at the water surface level
29 //so, at highest rocket will have 0 velocity which
     we will take as final velocity of rocket
30 //time required to reach highest position from water
     surface
31 //applying kinematic equations
32 time = (vy-v0y)/g//in seconds
33 //maximum distance travelled by the rocket above the
     water surface
34 //applying kinematic equations
35 y = (vy*time)-(0.5*g*time^2)//in meters
36 time = nearfloat("pred",11.8)
37 y = nearfloat("pred",688)
38 printf ("\n\n Acceleration of rocket in upward
     direction ay = \n\n %.1f m/s^2",ay);
39 printf ("\n\n Final velocity of the rocket at the
     surface of water vy = \n\n %3i m/s",vy);
40 printf ("\n\n Time required to reach highest
     position from water surface time = \n\n %.1f
     seconds",time);
41 printf ("\n\n Maximum distance travelled by the
     rocket above the water surface y = \n\n %2i m",y)
     ;
```

Chapter 3

FORCE AND NEWTONS LAWS

Scilab code Exa 3.1 C3P1

```
1
2  clc
3
4  //To Find the final velocity of sled
5
6  //Given :
7  //refer to figure 3-7(a) and3-7(b) from page no. 49
8  //mass of sled
9  m =240//in kg
10 //distance travelled
11 d =2.3//in m
12 //force applied
13 Fsw =130//in N
14
15 //solution:
16
17 //calculating first acceleration
18 //applying newton's second law
19 ax =Fsw/m //m/ s ^2
```

```
20 //calculating time required to move sled by distance
       d
21 //applying kinemtic equation
22 t =((2*d)/ax)^(1/2) //in seconds
23 // calculating velocity
24 //applying kinemtic equation
25 vx =ax*t //m/ s
26 printf ("\n\n Acceleration ax = \n\n %.2fm/s^2" ,ax)
27 printf ("\n\n final velocity vx = \n\n %.1f m/s" ,vx
       );
```

Scilab code Exa 3.2 C3P2

```
1
2 //To Find the force to be apply on sled
3 //referring to data from problem 3.1 on page no.48
4  clc;
5
6  //Given :
7  //refer to figure 3-7(a) and3-7(b) from page no. 49
8 m =240; //kg
9 d =2.3; //distance travelled in m
10 Fsw =130; //force in N
11
12 //solution:
13 //calculating acceleration
14 //applying newton's second law
15 ax1 = Fsw/m //m/s^2
16 //calculating time
17 //applying kinematic equation
18 t =  sqrt((2*d)/ax1) // s e c o n d s
19 //calculating velocity
20 //applying one dimensinal kinematic equation
21 vx =ax1*t; //m/ s
22 v0x = -(ax1*t); //m/s
```

```
23 t2 = 4.5 // s e c o n d s
24 // calculating first acceleration using equation vx
     = v0x + ax*t
25 ax =(v0x-vx)/t2; //m/ s ^2
26 ax = nearfloat("succ",0.71)
27
28 // calculating force
29 //applying newton's second law
30 F_dashsw = m*ax; // N
31 F_dashsw = nearfloat("pred",-170)
32 F_dashsw1 = F_dashsw/(0.4535*9.81)
33
34 printf ("\n\n Acceleration ax1 = \n\n %.2f m/s^2" ,
     ax1)
35 printf ("\n\n Time t = \n\n %.1f s" ,t)
36 printf ("\n\n Final velocity vx = \n\n %.1f m/s" ,vx
     )
37 printf ("\n\n Final velocity v0x = \n\n %.1f m/s" ,
     v0x)
38 printf ("\n\n Acceleration ax = \n\n %.2f m/s^2" ,ax
     )
39 printf ("\n\n Constant force F_dashsw = \n\n %3i N"
     ,F_dashsw);
40 printf ("\n\n Constant force F_dashsw1 = \n\n %2i lb
     " ,F_dashsw1);
```

Scilab code Exa 3.3 C3P3

```
1
2  clc
3
4 //To find force acting on crate
5
6 //Given :
7 //refer to figure 3-8(a) and3-8(b) from page no. 49
```

```
// mass
m =360 //kg
// initial velocity of crate
vx1 =62//km/ph
// final velocity of crate
v0x1 = 105 //km/ph
// time elapsed
t =17 //seconds

//solution:
//calculating initial velocity in m/s
vx =(62*5)/18 //in m/s
// calculating final velocity in m/s
v0x =(105*5)/18 //in m/s
//calculating acceleration
ax =(vx-v0x)/t //in m/s^2
//calculating force
//applying newton's secong law
Fct =m*ax //in seconds
ax = nearfloat("succ",-0.70)
Fct = nearfloat("pred",-250)

printf ("\n\n Acceleration a = \n\n %.2fm/s^2" ,ax)
printf ("\n\n Force acting on crate Fct =\n\n %.3iN"
    ,Fct);
```

Scilab code Exa 3.4 C3P4

```


clear
 clc

//To find force acting on crate

```

```
8  //Given :
9  //refer to figure 3-17(a) and3-17(b) from page no.
       55
10 // mass of first crate
11 m1 =4.2 //kg
12 // mass of second crate
13 m2 =1.4 //kg
14 // force on first crate
15 P1w =3 //in N
16
17 //solution://for two crate remain in contact acc(
       crate 1)=acc(crate 2). we will call this as
       common acelration as a.
18 // calculating common acceleration of both crate in
       m/s^2
19 //applying newton's secong law
20  a =P1w/(m1+m2) //m/s
21 // calculating force exerted on crate 2 by 1
22 //applying newton's secong law
23  f21 =m2*a //  m/s^2
24  f21 = nearfloat("succ",0.76)
25
26 printf ("\n\n Calculating common acceleration of
       both crate a = \n\n %.2fm/s^2" ,a)
27 printf ("\n\n Force acting on crate f21 = \n\n %.2f
       N" ,f21);
```

Scilab code Exa 3.5 C3P5

```
1
2
3
4 clear
5  clc
6
```

```
7 //To find force acting on crate
8
9  //Given :
10   //refer to figure 3-18(a) and3-18(b) from page no.
       55
11  // mass of flat-bed cart
12  mc =360 //kg
13  // mass of box
14  mb =150 //kg
15  // magnitude of acceleration for cart
16  ac =0.167 // m/s^2
17   // magnitude of acceleration for box
18  ab =1 // m/s^2
19
20
21 //solution:
22 //force on cart
23 //applying newton's second law
24 Fcb =mc*ac //in N
25 // force on box
26 //applying newton's second law
27 Fbw =Fcb+(mb*ab)// in N
28
29 printf ("\n\n Force acting on crate Fcb = \n\n %2i N
     " ,Fcb);
30 printf ("\n\n Force acting on box Fbw = \n\n %2i N"
     ,Fbw);
```

Scilab code Exa 3.6 C3P6

```
1
2
3
4 clear
5  clc
```

```
6
7  //To find frictional force of the box on the cart
8 // referrinf to same problem as 3-5 on page no.55
9
10  //Given :
11  // mass of flat-bed cart
12  mc =360 //kg
13  // mass of box
14  mb =150 //kg
15  // magnitude of acceleration for cart
16  ac =0.167 // m/s^2
17   // magnitude of acceleration for box
18  ab =1 // m/s^2
19
20
21 //solution:
22 // force on cart
23 //applying newton's second law
24  Fcb =mc*ac //in N
25 // force on box
26 //applying newton's second law
27 Fbw =Fcb+(mb*ab)// in N
28 //frictional force
29 //applying newton's second law
30 Fcb =(mc*Fbw)/(mc+mb)// in N
31 Fcb = nearfloat("succ",150)
32 //answer of Fcb slightli varies.but answer by scilab
       is same as on calculator
33 printf ("\n\n Frictional force  of box on the cart
     fcb = \n\n %3iN" ,Fcb);
```

Scilab code Exa 3.7 C3P7

```
1
2
```

```
3  clc
4 //to find net force on passenger ang scale reading
     while descending and ascending
5
6 // GIVEN::
7  //refer to figure 3-19(a) and3-19(b) from page no.
      56
8 //mass of passenger
9 m = 72.2 // in Kg
10 //acceleration of elevator while descending
11 a0y = 0// in m/s^2
12 // acceleration of elevator while ascending
13 ay = 3.20//in m/s^2
14 //acceleration due to gravity
15 g = 9.81//in m/s^2
16
17 // SOLUTION:
18
19 //passenger while descending
20 //applying newton's second law
21 Fps_d = m*(g+a0y)//in m/s^2
22 Fps_d1 = Fps_d/(g*.4535)//in lb
23 //passenger while ascending
24 //applying newton's second law
25 Fps_a = m*(g+ay)//in m/s^2
26 Fps_a1 = Fps_a/(g*.4535)//in lb
27 printf ("\n\n Net force on passenger while
     descending Fps_d = \n\n %3i N" ,Fps_d);
28 printf ("\n\n Net force on passenger while
     descending Fps_d1 = \n\n %3i lb" ,Fps_d1);
29 printf ("\n\n Net force on passenger while ascending
      Fps_a = \n\n %3i N" ,Fps_a);
30 printf ("\n\n Net force on passenger while ascending
      Fps_a1 = \n\n %3i lb" ,Fps_a1);
31 printf ("\n\n Scale raeding will not change while
     descending due to constant acceleration
     whilescale reading will increase while ascending
     due to increase in acceleration");
```

Chapter 4

MOTION IN TWO AND THREE DIMENSIONS

Scilab code Exa 4.1 C4P1

```
1
2
3  clc
4 //to find ship's velocity and position relative to
     its locationwhen the tractor beam first appeared
5
6 // GIVEN::
7 //refer to figure 4-1 from page no.66
8 //problem mainly divides into two parts
9 //1. t=0 to t=4 seconds    //FIRST PART
10 //2. t=4 to t=7 seconds    //SECOND PART
11
12 //1.for first part i.e.t=0 to t=4 seconds
13 // time interval for the first part is (4-0)=4
14 t1 = 4//in seconds
15 //initial position is (0,0)
16 x01 = 0
17 y01 = 0
18 //initial velocity in x direction for first part
```

```
19 v0x1 = 15//in km/s
20 //initial velocity in y direction for first part
21 v0y1 = 0//in km/s
22 //acceleration in x direction for the first part
23 ax1 = 0//in km/s^2
24 //acceleration in y direction for the first part
25 ay1 = 4.2//in km/s^2
26
27 //1.for second part i.e.t=4 to t=7 seconds
28 // time interval for the second part is (7-4)=3
29 t2 = 3//in seconds
30 //initial velocity in x direction for first part
31 v0x2 = 15//in km/s
32 //initial velocity in y direction for first part
33 v0y2 = 16.8//in km/s
34 //acceleration in x direction for the first part
35 ax2 = 18//in km/s^2
36 //acceleration in y direction for the first part
37 ay2 = 4.2//in km/s^2
38
39 // SOLUTION:
40
41 //1.for first part i.e.t=0 to t=4 seconds
42 //final velocity in x direction
43 vx1 = v0x1 + ax1*t1//in km/s
44 //final velocity in y direction
45 vy1 = v0y1 + ay1*t1//in km/s
46 //distance travelled in x direction
47 x1 = x01 + v0x1*t1 + (0.5*ax1*t1^2)//in km
48 //distance travelled in y direction
49 y1 = y01 + v0y1*t1 + (0.5*ay1*t1^2)//in km
50
51 //1.for second part i.e.t=4 to t=7 seconds
52 //now the position of ship is (x1,y1)
53 x02 = x1
54 y02 = y1
55 //final velocity in x direction
56 //applying kinematic equations
```

```
57 vx2 = v0x2 + ax2*t2//in km/s
58 //final velocity in y direction
59 //applying kinematic equatio
60 vy2 = v0y2 + ay2*t2//in km/s
61 //distance travelled in x direction
62 //applying kinematic equatio
63 x2 = x02 + v0x2*t2 + (0.5*ax2*t2^2)//in km
64 //distance travelled in y direction
65 //applying kinematic equatio
66 y2 = y02 + v0y2*t2 + (0.5*ay2*t2^2)//in km
67 //distance travelled by of ship
68 r = sqrt(x2^2 + y2^2)//in km
69 //velocity of ship
70 v = sqrt(vx2^2 + vy2^2)//in km/s
71 //position of ship
72 theta = atand(vy2/vx2)//in degrees
73 y2 = round(y2)
74 r = round(r)
75 printf ("\n\n Final velocity in x direction for
      first part vx1 = \n\n %.1f km/s",vx1);
76 printf ("\n\n Final velocity in y direction for
      first part vy1 = \n\n %.1f km/s",vy1);
77 printf ("\n\n Distance travelled in x direction for
      first part x1 = \n\n %.1f km",x1);
78 printf ("\n\n Distance travelled in y direction for
      first part y1 = \n\n %.1f km",y1);
79
80 printf ("\n\n Final velocity in x direction for
      second part vx2 = \n\n %.1f km/s",vx2);
81 printf ("\n\n Final velocity in y direction for
      second part vy2 = \n\n %.1f km/s",vy2);
82 printf ("\n\n Distance travelled in x direction for
      second part x2 = \n\n %3i km",x2);
83 printf ("\n\n Distance travelled in y direction for
      second part y2 = \n\n %3i km",y2);
84 printf ("\n\n Distance travelled by ship r = \n\n
      %3i km",r);
85 printf ("\n\n Velocity of ship v = \n\n %2i km/s",v)
```

```
    ;
86 printf ("\n\n Position of ship theta = \n\n %2i
    degrees",theta);
```

Scilab code Exa 4.2 C4P2

```
1
2
3
4  clc
5 //to find direction in which crate moving
6
7 // GIVEN::
8 //refer to figure 4-3(a),(b,(c) from page no.68
9 //mass of crate
10 m = 62//in kg
11 //initial velocity of crate in x direction
12 v0x = 6.4//in m/s
13 //initial velocity of crate in y direction
14 v0y = 0//in m/s
15 //force applied in opposite direction
16 Fct = 81//in N
17 //force applied in perpendicular direction
18 Fcj = 105//in N
19 //time interval while application of force
20 t = 3//in seconds
21
22 // SOLUTION:
23
24 //in x direction -Fct = m*ax
25 //in y direction Fcj = m*ay
26 //acceleration in x direction
27 //applying newton's second laww of motion
28 ax = -(Fct/m)//in m/s^2
29 //acceleration in y direction
```

```
30 ay = (Fcj/m)//in m/s^2
31 //component of velocity of crate in x direction
32 //applying kinematic equatio
33 vx = v0x + ax*t
34 //component of velocity of crate in y direction
35 //applying kinematic equatio
36 vy = v0y + ay*t
37 //resultant velocity of crate
38 v = sqrt(vx^2 + vy^2)//in m/s
39 //direction of velocity of crate
40 theta = atand(vy/vx)//in degrees
41 theta = nearfloat("succ",64)
42
43 printf ("\n\n Acceleration in x direction ax = \n\n
     %.2f m/s^2",ax);
44 printf ("\n\n Acceleration in y direction ay = \n\n
     %.2f m/s^2",ay);
45 printf ("\n\n Component of velocity of crate in x
     direction vx = \n\n %.1f m/s",vx);
46 printf ("\n\n Component of velocity of crate in y
     direction vy = \n\n %.1f m/s",vy);
47 printf ("\n\n Resultant velocity of crate v = \n\n %
     .1f m/s",v);
48 printf ("\n\n Direction of velocity of crate theta =
     \n\n %2i degrees",theta);
```

Scilab code Exa 4.3 C4P3

```
1
2
3
4  clc
5 //to find direction in which crate moving
6
7 // GIVEN::
```

```
8 //refer to figure 4-8 from page no.70
9 //velocity of plane
10 v =155//in km/h
11 //horizontal velocity of package
12 v0x = 155//in km/h
13 //since initial velocity of package is same that of
      plane but in horizontal direction
14
15 //elevation of plane directly above the target
16 y = -225//in meters
17 // y is negetive as packages are falling in downward
       direction
18 //acceleration due to gravity
19 g = 9.81//in m/s^2
20
21
22
23 // SOLUTION:
24
25 //time of fall
26 t = sqrt(-(2*y)/g)//in seconds
27 //horizontal distance travelled by the package in
      time t
28 //applying kinematic equations
29 x= ((v0x*t)/3600)*1000//in meters
30 //angle of sight should be
31 alpha = atand(x/abs(y))//in degrees
32 x = round(x)
33 t = nearfloat("succ",6.78)
34 printf ("\n\n Time of fall t = \n\n %.2f seconds",t)
      ;
35 printf ("\n\n Horizontal distance travelled by the
      package in time t x = \n\n %3i meters",x);
36 printf ("\n\n Angle of sight should be alpha = \n\n
      %2i degrees",alpha);
```

Scilab code Exa 4.4 C4P4

```
1
2
3
4  clc
5 //to find time t1 at which the ball reaches highest
     position of its trajectory
6 //maximun height at which ball can reach
7 //total time of flight and range of ball
8 //velocity of ball when it strikes the ground
9
10 // GIVEN::
11
12 //initial velocity of ball
13 v0 =15.5//in m/s
14 //angle made by the ball with horizontal
15 fi0 = 36//in degrees
16
17
18 //acceleration due to gravity
19 g = 9.81//in m/s^2
20
21
22
23 // SOLUTION:
24
25 //vertical component of initial velocity of ball
26 v0y = v0*sind(fi0)//in m/s
27 //vertical component of initial velocity of ball
28 v0x = v0*cosd(fi0)//in m/s
29 //velocity at the top position of trajectory
30 vy = 0//in m/s
31 // time t1 at which the ball reaches highest
```

```
    position of its trajectory
32 //applying kinematic equatio
33 t1 = (v0y-vy)/g//in seconds
34 ///maximun height at which ball can reach
35 //as maximum height is reached at time t = t1
36 //applying kinematic equatio
37 y_max = v0y*t1-(0.5*g*t1^2)//in meters
38 //total time of flight and range of ball
39 //for total time displacement = 0 i.e.y = 0
40 //applying kinematic equatio
41 t2 = (2*v0y)/g//in seconds
42 //range of the ball
43 //here range is the horizontal distance travelled in
       time t2
44 //applying kinematic equatio
45 x = v0x*t2//in m/s
46 ///velocity of ball when it strikes the ground
47  //horizontal componebt of velocty of ball when it
       strikes the ground
48 vx = v0*cosd(fi0)//in m/s
49 //vertical component of velocity of ball when it
    strikes the ground i.e. at time t2
50 vy = v0y-(g*t2)//in m/s
51 //applying kinematic equatiovy = v0y-(g*t2)//in m/s
52 //magnitude of velocity of ball when it strikes the
    ground
53 v = sqrt(vx^2 + vy^2)//in m/s
54 //direction of ball when it strikes the ground from
    x axis
55 fi = atand(vy/vx)//in degrees
56 fi = round(fi)
57 printf ("\n\n Time t1 at which the ball reaches
    highest position of its trajectory t1 = \n\n %.2f
     seconds",t1);
58 printf ("\n\n Maximun height at which ball can reach
     y_max = \n\n %.1f meters",y_max);
59 printf ("\n\n Total time of flight and range of ball
     t2 = \n\n %.2f seconda",t2);
```

```
60 printf ("\n\n Range of the ball x = \n\n %.1f meters
   ",x);
61 printf ("\n\n Horizontal componebt of velocty of
   ball when it strikes the ground vx = \n\n %.1f m/
   s",vx);
62 printf ("\n\n Vertical component of velocity of ball
    when it strikes the ground i.e. at time t2 vy =
   \n\n %.1f m/s",vy);
63 printf ("\n\n Magnitude of velocity of ball when it
   strikes the ground v = \n\n %.1f meters",v);
64 printf ("\n\n Direction of ball when it strikes the
   ground from x axis fi = \n\n %2i degrees",fi);
```

Scilab code Exa 4.5 C4P5

```
1
2
3
4  clc
5 //to find magnitude of gravitational force exterted
   on the moon by the earth
6
7 // GIVEN::
8
9 //time required for i revolution
10 d = 27.3//in days
11 //radius of orbit
12 r1 = 238000 //in mi
13 //radius of orbit in meters
14 r = (238000*1609.344)//in meters
15 //mass of the moon
16 m = 7.36*10^22//in kg
17
18 // SOLUTION:
19
```

```
20 //time for one complete revolution in seconds
21 T = (27.3*86400)//in seconds
22 //speed of the moon
23 v = (2*3.14*r)/T//in m/s
24 v = nearfloat("pred",1019)
25 //centripital force by gravitational force
26 // equation of centripital force F_ME = mv^2/r
27 F_ME = (m*v^2)/r//in N
28 printf ("\n\n Time for one complete revolution in
   seconds T = \n\n %.2e seconds",T);
29 printf ("\n\n Speed of the moon v  = \n\n %4i m/s",v
   );
30 printf ("\n\n Magnitude of gravitational force
   exterted on the moon by the earth F_ME = \n\n %.2
   e N",F_ME);
```

Scilab code Exa 4.6 C4P6

```
1
2
3
4  clc
5 //to find weight of the satellite at h = 210km above
     the earth's surface
6 //to find tangential speed of satellite required
7
8 // GIVEN::
9
10 //mass of the satellite
11 m =1250//in kg
12 //altitude at which satellite is required to be
   placed
13 h = 210//in km
14 //radius of the earth
15 R = 6370//in km
```

```
16 //acceleration due to gravity
17 g = 9.2//in m/s^2
18
19 // SOLUTION:
20
21 //weight of the satellite at the altitude h = 210km
     above earth's surface
22 w = m*g//in N
23 //to find tangential speed of satellite required
24 //force of gravity is weight of the satellite i.e.
     F_SE = w
25 //radius of orbit of satellite
26 r = R + h//in km
27 v =sqrt(w*(r*1000)/m)//in m/s //taking radius in
     meters
28 v1 = v*(3600/1609.344)//in mi/h
29 v1 = nearfloat("pred",17401)
30 printf ("\n\n Weight of the satellite at the
     altitude h = 210km above earths surface w = \n\n
     %.2e N",w);
31 printf ("\n\n Tangential speed of satellite required
      v = \n\n %4i m/s",v);
32 printf ("\n\n Tangential speed of satellite required
      v = \n\n %5i mi/h",v1);
```

Scilab code Exa 4.7 C4P7

```
1
2
3
4  clc
5 //to find velocity of plane with respect to ground
6 //to find compass reading if pilot wishes to fly due
     east
7
```

```
// GIVEN::
//refer to figure 4-18(a),(b) from page no.77
//speed of air on the indicator
V_PA =215//in km/h
//velocity of wind blowing due north
V_AG = 65//in km/h


// SOLUTION:


//magnitude of velocity of plane with respect to
   ground
V_PG1 = sqrt(V_PA^2 + V_AG^2)//in km/h
//direction of plane
//angle made by the plane with east direction
alpha = atand(V_AG/V_PA)//in degrees

//magnitude of velocity of plane if pilot wishes to
   fly due east
//now velocity of plane with respect to groung
   points east
V_PG2 = sqrt(V_PA^2 - V_AG^2)//in km/h
//direction of plane
//angle made by the plane with east direction
bita = asind(V_AG/V_PA)//in degrees
V_PG1 = round(V_PG1)
V_PG2 = round(V_PG2)
printf ("\n\n Magnitude of velocity of plane with
   respect to ground V_PG1 = \n\n %3i km/h",V_PG1);
printf ("\n\n Angle made by the plane with east
   direction alpha = \n\n %.1f degrees",alpha);
printf ("\n\n Magnitude of velocity of plane if
   pilot wishes to fly due east V_PG2 = \n\n %3i km/
   h",V_PG2);
printf ("\n\n Angle made by the plane with east
   direction bita = \n\n %.1f degrees",bita);
```

Chapter 5

APPLICATIONS OF NEWTONS LAWS

Scilab code Exa 5.1 C5P1

```
1
2 clear
3  clc
4 //to find tension in three strings, TA,TB and TC in
     strings A,B and C respectively.
5
6 // GIVEN::
7 //refer to figure 5-4(a) on page no. 91
8 //mass of block
9 m = 15//in kg
10 //acceleration due to gravity
11 g = 9.81//in m/s^2
12
13 // SOLUTION:
14
15 //considering free body diagram 5-4(b) let TA,TB,TC
     are tensions in string A,B and C respectively.
16 //applying newton's second law to the knot i.e. SUM(
     forces) = mass*acceleration
```

```
17 //resolving forces first in y direction refer fig.
     5-4(d)
18 //resolving forces first in x and y direction refer
     fig. 5-4(c)
19 //solving equations by generating matrix
20
21 A = [-cosd(30) cosd(45) 0 ; sind(30) sind(45) -1 ; 0
      0 1]
22 B = [0;0;(m*g)]
23
24
25 C = A\B
26 //tension in sting A
27 TA = C(1);//in N
28 //tension in sting B
29 TB = C(2);//in N
30 //tension in sting C
31 TC = C(3);//in N
32 TA = round(C(1))
33 TB = round(C(2))
34 TC = round(C(3))
35
36 printf ("\n\n Tension in string A is TA = \n\n %3i N
     ",TA);
37 printf ("\n\n Tension in string B is TB = \n\n %3i N
     ",TB);
38 printf ("\n\n Tension in string C is TC = \n\n %3i N
     ",TC);
```

Scilab code Exa 5.2 C5P2

```
1
2 clear
3  clc
4 //to find tension in the string (1)when elevator
```

```
    descending with constant velocity and (2)
    ascending with the acceleration of 3.2 m/s^2
5
6  // GIVEN::
7  //refer to figure 5-2(a) on page no. 91
8  //mass of block
9  m = 2.4//in kg
10 //acceleration due to gravity
11 g = 9.81//in m/s^2
12 //acceleration of elevator in y direction while
    descending
13 ay1 = 0//in m/s^2  since elevator is moving with
    constant velocity
14 //acceleration of elevator in y direction while
    ascending
15 ay2 = 3.2//in m/s^2
16
17
18 // SOLUTION:
19
20 //when elevator is descending
21 //considering free body diagram 5-4b from page no.91
22 //resolving forces first in y direction
23 //applying newton's second law i.e. SUM(forces) =
    mass*acceleration
24 T1 = (m*(g+ay1))//in N
25
26 //when elevator is descending
27 //considering free body diagram 5-4b from page no.91
28 //resolving forces first in y direction
29 //applying newton's second law i.e. SUM(forces) =
    mass*acceleration
30 T2 = m*(g+ay2)//in N
31 T1 = round(T1)
32 printf ("\n\n Tension in the string when elevator
    descending with constant velocity T1 = \n\n %2i N
    ",T1);
33 printf ("\n\n Tension in the string when elevator
```

```
    ascending with acceleration of 3.2m/s^2 T2 = \n\n
     %2i N",T2);
```

Scilab code Exa 5.3 C5P3

```
1
2  clear
3   clc
4  //to analyse the motion if (1)cord is horizontal and
       (2)the cord is making an angle of 15 degree with
       the  horizontal
5
6  // GIVEN::
7  //refer to figure 5-7(a) on page no. 92
8  //mass of sled
9  m = 7.5//in kg
10 //force by which sled is pulled
11 P = 21.0//in N
12 //angle made by sled with horizontal
13 theta = 15//in degrees
14 //acceleration due to gravity
15 g = 9.81//in m/s^2
16
17 // SOLUTION:
18
19 //when cord is horizontal
20
21 //considering free body diagram 5-7b from page no
       .92.
22 //euating forces in x direction
23 //applying newton's secong law of motion
24 //horizontal acceleration
25 ax = P/m//in m/s^2
26 ////equating forces in y direction
27 //applying newton's secong law of motion
```

```
28 //force exerted bu surface
29 N = round(m*g)//in N
30
31 //when cord is  making an angle of 15 degree with
      the horizontal
32
33 //considering free body diagram 5-7c from page no
      .93.
34 //euating forces in x direction and applying newton'
      s secong law of motion
35 //acceleration
36 a_x = P*cosd(theta)/m//in m/s^2
37 ////euating forces in y direction
38 //applying newton's secong law of motion
39 //normal force exerted bu surface
40 N_2 = ceil((m*g)-(P*sind(theta)))//in N
41 N = round(N)
42 N_2 = ceil(N_2)
43
44 printf ("\n\n Normal force exerted bu surface when
      cord is horizontal N1 = \n\n %2i N",N);
45 printf ("\n\n Acceleration in x direction when cord
      is horizontal ax1 = \n\n %.2f m/s^2",ax);
46 printf ("\n\n Normal force exerted bu surface when
      cord is making an angle of 15 degree with the
      horizontal N2 = \n\n %2i N",N_2);
47 printf ("\n\n Acceleration in x direction when cord
      is making an angle of 15 degree with the
      horizontal ax2 = \n\n %.2f m/s^2",a_x);
```

Scilab code Exa 5.4 C5P4

```
1
2 clear
3  clc
```

```
4 //to find tension in the string and normal force
     exerted on the block by the plane
5 //to analyse the motion when the string is cut
6
7 // GIVEN::
8 //refer to figure 5-7(a) on page no. 92
9 //mass of block
10 m = 18//in kg
11 //angle of inclination of plane
12 theta = 27//in degrees
13 //acceleration due to gravity
14 g = 9.81//in m/s^2
15
16
17 // SOLUTION:
18
19 //refer to the figure 5-8a from page no. 93
20 //considering free body diagram 5-8b from page no
     .93.
21
22 //whenthe block is stationary on the plane
23 //equating forces in x direction
24 //applying newton's second law of motion
25 //tension in the string
26 T = m*g*sind(theta)//in N
27 //equating forces in y direction
28 //applying newton's second law of motion
29 //normal reaction by the surface
30 N = m*g*cosd(theta)//in N
31
32 //when the string is cut
33 //equating forces in x direction
34 //applying newton's second law of motion
35 //acceleration of block in x direction ax
36 ax = -(g*sind(theta))//in m/s^2
37 //-ve sign indicates acceleration acting in -ve x
     direction i.e. downwards
38 printf ("\n\n Tension in the string T = \n\n %2i N",
```

```
    T);
39 printf ("\n\n Normal force exerted on the block by
    the plane N = \n\n %3i N",N);
40 printf ("\n\n Acceleration of block in x direction
    when the string is cut ax  = \n\n %.2f m/s^2",ax)
    ;
```

Scilab code Exa 5.7 C5P7

```
1
2
3  clc
4 //to find tension in the string
5 //to find acceleration of blocks
6
7 // GIVEN::
8 //refer to figure 5-11(a) on page no. 95
9 //mass of first block
10 m1 = 9.5//in kg
11 //angle of inclination of plane
12 theta = 34//in degrees
13 //mass of second block
14 m2 = 2.6//in kg
15 //acceleration due to gravity
16 g = 9.81//in m/s^2
17
18
19 // SOLUTION:
20
21 //refer to the free body diagrams 5-11b and 5-11c
    from page no. 95
22
23
24 //for mass m1
25 //assuming m1 moves in positive x direction
```

```
26 //equating forces in x direction and applying newton
      's second law of motion
27 //equating forces in y direction and applying newton
      's second law of motion
28
29 //for mass m2
30 //equating forces in y direction and applying newton
      's second law of motion
31 //solving above equations simultaneously using
      matrix form
32 //acceleration of blocks
33 a = (m2-(m1*sind(theta)))*g/(m1 + m2)//in m/s^2
34 //if ans. for a is -ve then our assumption is wrong
      i.e. m1 is moving in -ve x direction but
      magnitude of ans is correct
35 //tension in the string
36 T = ((m1*m2*g)*(1 + sind(theta)))/(m1 + m2)//in N
37
38 printf ("\n\n Acceleration of blocks a = \n\n %.1f m
      /s^2",a);
39 printf ("\n\n Tension in the string T = \n\n %2i N",
      T);
```

Scilab code Exa 5.9 C5P9

```
1
2 clear
3  clc
4 //to find shortest distance in which automobile can
      stop
5
6
7 // GIVEN::
8 //refer to figure 5-16 on page no. 99
9 //initial velocity of automobile in mi/h
```

```
10 v01 = 60//in mi/h
11 //coefficient of static friction
12 mew_s = 0.60
13 //acceleration of gravity
14 g = 9.81//in m/s^2
15
16 // SOLUTION:
17 //N is normal reaction force by surface.
18 //refer to the free body diagrams 5-16b from page no
      . 95
19
20 //initial velocity of automobile in m/s
21 v0 = v01*(1609/3600)//in m/s
22 //applying newton's law in x and y direction
23 //applying kinematic equation of motion
24 //shortest distance in which automobile can stop
25 d = ((v0^2)/(2*mew_s*g))//in meters
26 d = ceil(d)
27 printf ("\n\n Shortest distance in which automobile
      can stop d  = \n\n %2i m",d);
```

Scilab code Exa 5.10 C5P10

```
1
2 clear
3  clc
4 //to find tension in the string and normal force
     exerted on the block by the plane
5 //to analyse the motion when the string is cut
6
7 // GIVEN::
8 //refer to problem 5.7 from page no. 95
9 //mass of first block
10 m1 = 9.5//in kg
11 //angle of inclination of plane
```

```
12 theta = 34//in degrees
13 //mass of second block
14 m2 = 2.6//in kg
15 //acceleration due to gravity
16 g = 9.81//in m/s^2
17 //coefficient of static friction
18 mew_s = 0.24
19 //coefficient of kinetic friction
20 mew_k = 0.15
21
22
23
24 // SOLUTION:
25 //T is tension in the spring and N is normal
      reaction force by surface.
26 //refer to the free body diagrams 5-17a from page no
      . 99
27
28 //for mass m1 and m2
29 //assuming m1 moves in positive x direction
30 //equating forces in x direction and applying newton
      's second law of motion
31 //equating forces in y direction and applying newton
      's second law of motion
32 //acceleration of blocks
33 a = (m2-(m1*(sind(theta)- (mew_k*cosd(theta)))))*g/(
      m1 + m2)//in m/s^2
34 //if ans. for a is -ve then our assumption is wrong
      i.e. m1 is moving in -ve x direction but
      magnitude of ans is correct
35 //tension in the string
36 T = (((m1*m2*g)*(1 + sind(theta) - (mew_k*cosd(theta
      ))))/(m1 + m2))//in N
37 T = round(T)
38
39 printf ("\n\n Acceleration of blocks a = \n\n %.1f m
      /s^2",a);
40 printf ("\n\n Tension in the string T = \n\n %2i N",
```

```
T);
```

Scilab code Exa 5.11 C5P11

```
1
2 clear
3  clc
4 //to find time required by the car to come to rest.
5 //to find the distance travelled by car before
     stopping.
6
7
8 // GIVEN::
9
10 //mass of car
11 m = 1260//in kg
12 //velocity of car
13 v0x = 29.2//in m/s
14 //rate at which breaking force increases with time
15 c = 3360//in N/s
16
17 // SOLUTION:
18 //assuming car's velocity is in +ve x direction
19 //applying newton's second law of motion
20 //applying kinematic equation of motion to derive
     velocity relation and distance travelled relation
21 //time required by the car to come to rest
22 t1 = sqrt((2*v0x*m)/c)//in seconds
23 //distance travelled by car before stopping
24 //here we are taking time is t1 and x0 is 0
25 x(t1) = 0 + (v0x*t1)-((c*(t1^3))/(6*m))//in meters
26 printf ("\n\n Time required by the car to come to
     rest t1 = \n\n %.2f seconds",t1);
27 printf ("\n\n Distance travelled by car before
     stopping x(t1) = \n\n %.1f m",x(t1));
```

Chapter 6

MOMENTUM

Scilab code Exa 6.1 C6P1

```
1
2 clear
3  clc
4 //to find impluse of the force exerted on the ball.
5 //to find average force assuming collision lasts for
      1.5ms
6 //to find the change in momentum of the bat
7
8 // GIVEN::
9 //refer to figure 6-8(a) on page no. 123
10 //mass of baseball
11 m = 0.14//in kg
12 //refer to figure 6.1
13 //horizontal speed of the ball
14 vi = 42//in m/s
15 //speed at which ball leaes i.e. final speed of the
     ball
16 vf = 50//in m/s
17 //angle at which  ball leaves
18 fi = 35//in degrees
19 //time for which collision lasts
```

```
delta_t = 0.0015//in seconds

// SOLUTION:

//refer to figure 6-8(a) on page no. 123
//component of final momentum in x direction
pfx = m*vf*cosd(fi)//in kgm/s
//component of final momentum in y direction
pfy = m*vf*sind(fi)//in kgm/s
//since initial momentum has only x componen
piy = 0//in kgm/s
//component of intial momentum in x direction
//considering our coordinate system as shown 6-8(a)
pix = m*(-vi)//in kgm/s
//using impluse momentum relation
//component of impluse in x direction
Jx = pfx-pix//in kgm/s
//component of impluse in y direction
Jy = pfy-piy//in kgm/s
//final magnitude of impluse
J = sqrt(Jx^2 + Jy^2)//in kgm/s
//direction in which impluse acts
theta = atand(Jy/Jx)//in degrees
//average force
//using impluse force relationship
Fav = J/delta_t//in N
Fav = nearfloat("succ",8200)

//applying newton's third law of motion
//for bat delta_px will be equal and opposite to
   that of ball
//component change in momemtum in x direction
delta_px = -(pfx - pix)//in kgm/s
//component change in momemtum in y direction
delta_py = -(pfy - piy)//in kgm/s

printf ("\n\n Component of final momentum in x
   direction pfx  = \n\n %.1f kgm/s",pfx);
```

```
56 printf ("\n\n Component of final momentum in y
      direction pfy  = \n\n %.1f kgm/s",pfy);
57 printf ("\n\n Component of intial momentum in x
      direction pix = \n\n %.1f kgm/s",pix);
58 printf ("\n\n Component of impluse in x direction Jx
       = \n\n %.1f kgm/s",Jx);
59 printf ("\n\n Component of impluse in y direction Jy
       = \n\n %.1f kgm/s",Jy);
60 printf ("\n\n Final magnitude of impluse J = \n\n %
      .1f kgm/s",J);
61 printf ("\n\n Direction in which impluse acts theta
     = \n\n %2i degrees",theta);
62 printf ("\n\n Average force Fav = \n\n %4i N",Fav);
63 printf ("\n\n Component change in momemtum in x
      direction delta_px = \n\n %.1f kgm/s",delta_px);
64 printf ("\n\n Component change in momemtum in y
      direction delta_py = \n\n %.1f kgm/s",delta_py);
```

Scilab code Exa 6.2 C6P2

```
1
2 clear
3  clc
4 //to find velocity of carts after collision
5
6 // GIVEN::
7 //we consider +ve x direction as direction of motion
      of first cart
8 //refer to figure 6-9 on page no. 123
9 //mass of first cart
10 m1 = 0.24//in kg
11 //initial velocity of first cart
12 v1i = 0.17//in m/s
13 //initial velocity of second cart
14 //as 2nd cart is initially at rest
```

```
15 v2i = 0//in m/s
16 //mass of second cart
17 m2 = 0.68//in kg
18
19
20
21 // SOLUTION:
22
23 //refer to figure 6-9 on page no. 123
24 //using impluse force relationship
25 //magnitude of impluse i.e. area under graph 6-9 on
     page 123
26 J = 0.5*(0.014-0.003)*10//in kgm/s
27
28 //assuming direction of motion of first cart is in +
     ve x direction
29 //change in momentum in x direction for first cart
30 delta_p1x = -(J)//in kgm/s
31 //initial momentum of first cart in x direction
32 p1ix = m1*v1i//in kgm/s
33 //final momentum for first cart
34 p1fx = p1ix + delta_p1x//in kgm/s
35 //final velocity of first cart in x direction
36 v1fx = p1fx/m1//in m/s
37 v1fx = nearfloat("pred",-0.058)
38
39
40 //as direction of motion of first cart is in +ve x
     direction for second cart it will be in -ve x
     direction
41 //using newton's third law of motion
42 //change in momentum in x direction for second cart
43 delta_p2x = (J)//in kgm/s
44 //initial momentum of second cart in x direction
45 p2ix = m2*v2i//in kgm/s
46 //final momentum for second cart
47 p2fx = p2ix + delta_p2x//in kgm/s
48 //final velocity of second cart in x direction
```

```
49 v2fx = p2fx/m2//in m/s
50 printf ("\n\n Magnitude of impluse J = \n\n %.3f Kg.
      m/s",J);
51 printf ("\n\n Change in momentum in x direction for
      first cart delta_p1x  = \n\n %.3f Kg.m/s",
      delta_p1x);
52 printf ("\n\n Final momentum for first cart p1fx = \
      n\n %.3f Kg.m/s",p1fx);
53 printf ("\n\n Final velocity of first cart in x
      direction v1fx = \n\n %.3f m/s",v1fx);
54 printf ("\n\n Final momentum for second cart p2fx =
      \n\n %.3f Kg.m/s",p2fx);
55 printf ("\n\n Final velocity of second cart in x
      direction v2fx = \n\n %.3f m/s",v2fx);
```

Scilab code Exa 6.3 C6P3

```
1
2 clear
3  clc
4 //to find direction in which Fred is skating after
      breaking in contact
5
6 // GIVEN::
7 //we consider +ve x direction for initial motion
8 //refer to figure 6-11 on page no. 125
9 //mass of Fred
10 mF = 75//in kg
11 //mass of Ginger cart
12 mG = 55//in kg
13 //common velocity of Fred ang Ginger
14 vG = 3.2//in m/s
15 vF = 3.2//in m/s
16 //after breaking contact angle of Ginger skating
17 theta1 = 32//in degrees
```

```
18
19 // SOLUTION:
20
21 //refer to figure 6-11(a) on page no. 125
22 //using consevation of momentum
23 //x component of Ginger orignal momentum
24 PGx = mG*vG //in kgm/s
25 //x component of Fred orignal momentum
26 PFx = mF*vF //in kgm/s
27 //after they push off y component of Ginger momentum
28 PGy = PGx*tand(theta1)//in kgm/s
29 //after they push off y component of Fred momentum
     will be opposite that of Ginger
30 //using consevation of momentum
31 PFy = -(PGy)//in kgm/s
32 tan_theta = (PFy/PFx)
33 //direction in which Fred is skating after breaking
     in contact
34 theta = atand(PFy/PFx)//in degrees
35 PGy = round(PGy)
36 theta = round(theta)
37
38 printf ("\n\n X component of Ginger orignal momentum
      PGx = \n\n %3i Kg.m/s",PGx);
39 printf ("\n\n X component of Fred orignal momentum
     PFx = \n\n %3i Kg.m/s",PFx);
40 printf ("\n\n After they push off y component of
     Ginger momentum PGy = \n\n %3i Kg.m/s",PGy);
41 printf ("\n\n Value of tan_theta = \n\n %.3f degrees
     ",tan_theta);
42 printf ("\n\n Direction in which Fred is skating
     after breaking in contact theta = \n\n %2i
     degrees",theta);
```

Scilab code Exa 6.4 C6P4

```
1
2 clear
3  clc
4 //to find final velocity of man when seated in
     rowboat
5
6 // GIVEN::
7 //we consider +ve x direction as man's original
     velocity
8 //refer to figure 6-12 on page no. 126
9 //mass of man
10 mm = 65//in kg
11 //speed of man initially in x direction
12 vmx = 4.9//in m/s
13 //mass of rowboat
14 mb = 88//in kg
15 //speed of rowboat in x direction
16 vbx = 1.2//in m/s
17
18 // SOLUTION:
19
20 //refer to figure 6-12(a) and 6-12(b) on page no.
     126
21
22
23 //before man jumps
24 //momentum of man in x direction
25 pmx = mm*vmx//in kgm/s
26 //momentum of boat in x direction
27 pbx = mb*vbx//in kgm/s
28 //total initial momentum in x direction
29 pix = pmx + pbx//in kgm/s
30
31 //after man jumps
32 //combined final momentum of man and boat in x
     direction
33 //applying conservation of momentum for boat and man
34 //final velocity of man when seated in rowboat in x
```

```
    direction
35 vfx = (pix/(mm + mb))//in m/s
36 printf ("\n\n Total initial momentum in x direction
    pix = \n\n %.3i Kg.m/s",pix);
37 printf ("\n\n Final velocity of man when seated in
    rowboat in x direction vfx = \n\n %.1f m/s",vfx);
```

Scilab code Exa 6.5 C6P5

```
1
2 clear
3  clc
4 //to find velocity of second glider after collision
5
6 // GIVEN::
7
8 //we consider +ve x direction as initial motion of
    first glider
9 //mass of first glider
10 m1 = 1.25//in kg
11 //initial velocity of first glider in +ve x
    direction
12 v1ix = 3.62//in m/s
13 //mass of second glider
14 m2 = 2.30//in kg
15 //final velocity of first glider in +ve x direction
16 // - sign since after collision first glider is
    moving in -ve x direction
17 v1fx = -1.07//in m/s
18 //initial velocity of second glider in +ve x
    direction
19 //since 2nd glider is initially at rest
20 v2ix = 0//in m/s
21
22 // SOLUTION:
```

```
23
24 //applying conservation of momentum
25 //final velocitiy of second glider in +ve x
      direction
26 v2fx = (m1/m2)*(v1ix-v1fx)//in m/s
27 //change in momentums for glider having mass m1
28 delta_p1x = m1*(v1fx-v1ix)//in Kg.m/s
29 //change in momentums for glider having mass m2
30 delta_p2x = m2*(v2fx-v2ix)//in Kg.m/s
31
32 printf ("\n\n Velocitiy of second glider in +ve x
      direction after collision v2fx = \n\n %.2f m/s",
      v2fx);
33 printf ("\n\n Change in momentums for glider having
      mass m1 delta_p1x = \n\n %.2f Kg.m/s",delta_p1x);
34 printf ("\n\n Change in momentums for glider having
      mass m2 delta_p2x = \n\n %.2f Kg.m/s",delta_p2x);
```

Scilab code Exa 6.6 C6P6

```
1
2 clear
3  clc
4 //to find final velocity of combination of 1st and 2
      nd glider
5
6 // GIVEN::
7 //refer to problem 6-5 from page no. 127
8
9 //we consider +ve x direction as initial motion of
      first glider
10 //mass of first glider
11 m1 = 1.25//in kg
12 //initial velocity of first glider in +ve x
      direction
```

```
13 v1ix = 3.62//in m/s
14 //mass of second glider
15 m2 = 2.30//in kg
16 //initial velocity of second glider in +ve x
      direction
17 //since 2nd glider is initially at rest
18 v2ix = 0//in m/s
19
20
21 // SOLUTION:
22
23 //applying conservation of momentum
24 //final velocitiy of second glider in +ve x
      direction
25 vfx = (m1*v1ix)/(m1 + m2)//in m/s
26 //change in momentums for glider having mass m1
27 delta_p1x = m1*(vfx-v1ix)//in Kg.m/s
28 //change in momentums for glider having mass m2
29 delta_p2x = m2*(vfx-v2ix)//in Kg.m/s
30
31 printf ("\n\n Final velocity of combination of 1st
      and 2nd glider vfx = \n\n %.2f m/s",vfx);
32 printf ("\n\n Change in momentums for glider having
      mass m1 delta_p1x = \n\n %.2f Kg.m/s",delta_p1x);
33 printf ("\n\n Change in momentums for glider having
      mass m2 delta_p2x = \n\n %.2f Kg.m/s",delta_p2x);
```

Scilab code Exa 6.7 C6P7

```
1
2
3
4  clc
5 //to find final speed of larger craft
6
```

```
7 // GIVEN::
8 //refer to diagram 6-14 from page no. 127
9
10 //we consider +ve x direction as original motion of
     spaceship(and also that of final velocity of
     smaller craft)
11 //total mass of spaceship
12 //M = m//in kg
13 //let us consider m = 1
14 M = 1//in kg
15 //mass of smaller crafy
16 //m1 = m/4//in kg
17 m1 = 1/4//in kg
18 //mass of larger craft
19 //m2 =3* m/4//in kg
20 m2 =3* 1/4//in kg
21 //initial velocity of spaceship in +ve x direction
22 vix = 8.45//in km/s
23 //final speed of smaller craft in +ve x direction
24 v1fx = 11.63//in km/s
25
26
27 // SOLUTION:
28
29 //applying conservation of momentum
30 //final velocity of larger craft in +ve x direction
31 v2fx = (((m1 + m2)*vix)-(m1*v1fx))/m2//in m/s
32
33 printf ("\n\n Final velocity of larger craft in +ve
     x direction v2fx  = \n\n %.2f km/s",v2fx);
```

Scilab code Exa 6.8 C6P8

```
1
2
```

```
3
4  clc
5 //to find final speed and direction of second puck
     after collision
6
7 // GIVEN::
8 //refer to diagram 6-15 from page no. 128
9
10 //we consider +ve x direction as initial motion of
     first puck
11 //mass of first puck
12 //assume mass of first puck be 1kg
13 m1 = 1//in kg
14 //mass of second puck
15 //mass of second puck is 1.5 times mass of first
     puck
16 m2 = 1.5//in kg
17 //initial velocity of first puck in +ve x direction
18 v1ix = 2.48//in m/s
19 //initial velocity of second puck in +ve x direction
20 v2ix = 1.86//in m/s
21 //initial direction of second puck away from the
     direction of first puck
22 theta1 = 40//in degrees
23 //final velocity of first puck after collision
24 v1fx = 1.59//in m/s
25 //final direction of first puck after collision
26 theta2 = 50//in degrees
27
28 // SOLUTION:
29
30 //applying law of conservation of momentum in x and
     y direction
31 //solving equation
32 //final direction of second puck after collision
33 theta = atand(0.38/2.40)//in degrees
34 //final speed  of second puck after collision
35 v2f = 2.40/cosd(theta)//in m/s
```

```
36 printf ("\n\n Final speed  of second puck after
      collision v2f = \n\n %.2f m/s",v2f);
37 printf ("\n\n Final direction of second puck after
      collision theta = \n\n %.1f degrees",theta);
```

Scilab code Exa 6.9 C6P9

```
1
2
3
4  clc
5 //to find velocity of alpha partical after collision
6 //to find which type of collision is this listed in
     fig. 6-17
7
8 // GIVEN::
9 //refer to diagram 6-17 from page no. 130
10
11 //we consider +ve x direction as initial velocity of
      alpha partical
12 //mass of alpha partical m1 = 4.0u
13 //assume u = 1
14 ma = 4.0
15 //mass of oxygen nucleus m2 = 16.0u
16 //assume u = 1
17 mo = 16.0
18 //initial velocity of alpha partical in +ve x
     direction
19 vaix = 1.52*10^7//in m/s
20 //initial velocity of oxygen nucleus in +ve x
     direction
21 //as oxygen nucleus is initially at rest
22 voix = 0//in m/s
23 //final velocity of oxygen nucleus after collision
24 vofx = 6.08*10^6//in
```

```
25
26
27 // SOLUTION:
28
29 //applying law of conservation of momentum in x
      direction
30 //final velocity of alpha partical after collision
      in x direction
31 vafx = ((ma*vaix)-(mo*vofx))/ma//in m/s
32 //applying law of conservation of momentum in x
      direction
33 //we can find out collision is elstic collision as
      alpha partical only reverses the direction of
      momentum after collision
34 //relative velocity
35 vx = (ma*vaix+mo*voix)/(ma+mo)//in m/s
36
37 printf ("\n\n Final velocity of alpha partical after
       collision in x direction vafx = \n\n %.2e m/s",
      vafx);
38 printf ("\n\n Relative velocity vx  = \n\n %.3e m/s"
      ,vx);
39 printf ("\n\n Collision is elstic collision");
```

Chapter 7

SYSTEMS OF PARTICLES

Scilab code Exa 7.2 C7P2

```
1  clear
2  clc
3  //to find center of mass of system
4  //to find acceleration of center of mass
5
6  // GIVEN::
7
8  //refer to figure 7-10(a) fron page no. 144
9  //consider +ve x direction as our reference axis
10 //mass of first partical
11 m1 = 4.1//in kg
12 //mass of second partical
13 m2 = 8.2//in kg
14 //mass of third partical
15 m3 = 4.1//in kg
16 //from figure 7-20(a)
17 //x coordinate of first partical
18 x1 = -2//in cm
19 //y coordinate of first partical
20 y1 = 3//in cm
21 //x coordinate of second partical
```

```
22 x2 = 4//in cm
23 //y coordinate of second partical
24 y2 = 2//in cm
25 //x coordinate of third partical
26 x3 = 1//in cm
27 //y coordinate of third partical
28 y3 = -2//in cm
29 //magnitude of first external force
30 F1 = -6//in N //since acting in -ve x direction
31 //magnitude of second external force
32 F2 = 12//in N
33 //magnitude of third external force
34 F3 = 14//in N
35
36 // SOLUTION:
37 //refer to figure 7-10(a) and 7-10(b) fron page no.
     144
38 //assuming all external forces are applied at center
     of mass
39 //total mass of system
40 M = m1 + m2 + m3//in kg
41 //applying center of mass formula
42 //x coordinate of center of mass
43 x_cm = (1/M)*(m1*x1 + m2*x2 + m3*x3)//in cm
44 //y coordinate of center of mass
45 y_cm = (1/M)*(m1*y1 + m2*y2 + m3*y3)//in cm
46
47 //refer to figure 7-10(b)
48 //component of force F1 in x direction
49 F1x = F1//in N
50 //component of force F2 in x direction
51 F2x = F2*cosd(45)//in N
52 //component of force F3 in x direction
53 F3x = F3//in N
54 //component of force F1 in y direction
55 F1y = 0//in N
56 //component of force F2 in y direction
57 F2y = F2*sind(45)//in N
```

```
58 //component of force F3 in y direction
59 F3y = 0//in N
60 //x component of net external force acting on the
      center of mass
61 SUM_fextx = F1x + F2x + F3x//in N
62 //y component of net external force acting on the
      center of mass
63 SUM_fexty = F1y + F2y + F3y//in N
64 //magnitude of net external force acting on the
      center of mass
65 SUM_Fext = sqrt(SUM_fextx^2 + SUM_fexty^2)//in N
66 //direction in which net force acts
67 fi = atand(SUM_fexty/SUM_fextx)//in degrees with x
      axis
68 //acceleration of center of mass
69 a_cm = SUM_Fext/(M)//in m/s^2
70 SUM_Fext = nearfloat("succ",18.6)
71
72 printf ("\n\n x coordinate of center of mass x_cm =
      \n\n %.1f cm",x_cm);
73 printf ("\n\n y coordinate of center of mass y_cm =
      \n\n %.1f cm",y_cm);
74 printf ("\n\n Magnitude of net external force acting
       on the center of mass in x direction SUM_fextx =
       \n\n %.1f N",SUM_fextx);
75 printf ("\n\n Magnitude of net external force acting
       on the center of mass in y direction SUM_fexty =
       \n\n %.1f N",SUM_fexty);
76 printf ("\n\n Magnitude of net external force acting
       on the center of mass SUM_Fext = \n\n %.1f N",
      SUM_Fext);
77 printf ("\n\n Direction in which net force acts with
       x axis fi = \n\n %2i degrees",fi);
78 printf ("\n\n Acceleration of center of mass a_cm =
      \n\n %.1f m/s^2",a_cm);
```

Scilab code Exa 7.3 C7P3

```
1
2
3
4  clc
5 //to find location of second fragment
6
7 // GIVEN::
8
9 //refer to figure 7-11 fron page no. 145
10 //consider +ve x direction as our reference axis
11 //mass of projectile
12 M = 9.6//in kg
13 //initial velocity of projectile
14 v0 = 12.4//in m/s
15 //angle of projectile above horizontal
16 fi0 = 54//in degrees
17 //mass of first piece after explosion
18 m1 = 6.5//in kg
19 //time after which first piece id observed
20 t = 1.42//in seconds
21 //vertical distance at which first piece is observed
22 y1 = 5.9//in meters
23 //horizontal distance at which first piece is
     observed
24 x1 = 13.6//in meters
25 //acceleration due to gravity
26 g = 9.80//in m/s^2
27
28 // SOLUTION:
29
30 //refer to figure 7-11 from page no. 145
31 //mass of second piece
```

```
32 //by mass conservation
33 m2 = M-m1//in kg
34 //velocity of projectile in +ve x direction
35 v0x = v0*cosd(fi0)//in m/s
36 //velocity of projectile in +ve y direction
37 v0y = v0*sind(fi0)//in m/s
38 //using kinematic equation of motion
39 //x coordinate of position of original projectile
40 x = v0x*t//in m
41 //y coordinate of position of original projectile
42 y = (v0y*t)-(0.5*g*t^2)//in m
43 //applying center of mass formula
44 //x coordinate of posion of second piece
45 x2 = (M*x - m1*x1)/m2//in meters
46 //y coordinate of posion of second piece
47 y2 = (M*y - m1*y1)/m2//in meters
48 x = nearfloat("succ",10.4)
49 y = nearfloat("pred",4.3)
50 x2 = nearfloat("succ",3.7)
51 y2 = nearfloat("pred",0.9)
52
53 printf ("\n\n x coordinate of position of original
      projectile x = \n\n %.1f m",x);
54 printf ("\n\n y coordinate of position of original
      projectile y = \n\n %.1f m",y);
55 printf ("\n\n x coordinate of posion of second piece
       x2 = \n\n %.1f m",x2);
56 printf ("\n\n y coordinate of posion of second piece
       y2 = \n\n %.1f m",y2);
```

Scilab code Exa 7.7 C7P7

```
1
2 clear
3 clc
```

```
4 //to find speed of block after it has absorbed eight
      bullets
5
6 // GIVEN::
7 //refer to figure 7-16 fron page no. 148
8 //mass of bullet
9 m1 = 3.8//n gram
10 m = 3.8*10^-3//in kg
11 //speed of bullet
12 v = 1100//in m/s
13 //mass of wooden block
14 M = 12//in kg
15 //number of bulletes
16 N = 8
17
18 // SOLUTION:
19 //refer to figure 7-16 from page no. 148
20 //consider +ve x direction to the right as seen in
      fig. 7-16
21 //applying momentum conservation before bullets are
      stillin flight and after bullets are in the block
22 //speed of block after it has absorbed eight bullets
23 V = ((N*m)/(M + N*m))*v//in m/s
24 printf ("\n\n Speed of block after it has absorbed
      eight bullets V = \n\n %.1f m/s",V);
```

Scilab code Exa 7.8 C7P8

```
1 clear
2 clc
3 //to find velocity of the recoiling cannon with
      respect to the earth
4 //to find initial velocity vE of the ball with
      respect to the earth
5
```

```
6 // GIVEN::
7
8 //refer to figure 7-17 fron page no. 149
9 //mass of cannon
10 M = 1300//in kg
11 //mass of ball fired
12 m = 72//in kg
13 //speed of ball in horizontal x direction
14 vx = 55//in m/s
15
16 // SOLUTION:
17
18 //refer to figure 7-17 from page no. 149
19 //considering cannon and ball is our system and
     consider +ve x as right direction
20 //finding momentum of our system with respect to the
     earth
21 //applying conservation of momentum
22 Vx = -(m*vx)/(M + m)//in m/s //-ve signs as cannon
     recoils in left direction
23 //initial velocity vE of the ball with respect to
     the earth
24 vEx = vx + Vx//in m/s
25
26 printf ("\n\n Velocity of the recoiling cannon with
     respect to the earth Vx = \n\n %.1f m/s",Vx);
27 printf ("\n\n Initial velocity vE of the ball with
     respect to the earth vEx = \n\n %2i m/s",vEx);
```

Scilab code Exa 7.9 C7P9

```
1 clear
2 clc
3 //to find thrust produced by the rocket
4 //to find velocity of the spaceship after the
```

```
    rockets have fired
5
6  // GIVEN::
7
8  //total mass of spaceship
9  M = 13600//in kg
10 //initial speed of spaceship
11 vix = 960//in m/s
12 //rate at which rocket ejects gas
13 dM_by_dt = 146//in kg/s
14 //speed at which rocket ejects gas
15 vrel = 1520//in m/s
16 //mass of gas burned and ejected from spaceship
17 m = 9100//in kg
18
19 // SOLUTION:
20
21 //consider +ve x direction in the direction of
    spaceship's initial velocity
22 //thrust produced by the rocket
23 F = vrel*dM_by_dt//in N
24 //initial mass of gas
25 Mi = 13600//inkg
26 //final mass of gass
27 Mf = Mi-m//in kg
28 //rewriting equation of velocity and integrating
    velocity equation from initial to final
    conditions
29 //velocity of the spaceship after the rockets have
    fired
30 vfx = vix + (-vrel*(log(Mf/Mi)))//in m/s
31 vfx = nearfloat("pred",2641)
32 printf ("\n\n Thrust produced by the rocket F = \n\n
    %.2e N",F);
33 printf ("\n\n Velocity of the spaceship after the
    rockets have fired vfx = \n\n %4i m/s",vfx);
```

Scilab code Exa 7.10 C7P10

```
clear
clc
//to find force to be applied on conveyor belt to
   keep it moving with constant speed

// GIVEN::

//refer to figure 7-20 fron page no. 151
//rate at which sand is being dropped
dM_by_dt = 0.134//in kg/s
//speed at which sand is being dropped
vx = 0.96//in m/s

// SOLUTION:

//refer to figure 7-20 from page no. 151
//consider +ve x as direction of motion of belt and
   applying equation for systems of variable mass
//force to be applied on conveyor belt to keep it
   moving with constant speed
sum_F_extx = (vx*dM_by_dt)//in N
printf ("\n\n Force to be applied on conveyor belt
   to keep it moving with constant speed sum_F_extx
   = \n\n %.3f N",sum_F_extx);
```

Chapter 8

ROTATIONAL KINEMATICS

Scilab code Exa 8.1 C8P1

```
clear
clc
//to find average angular velocity of fan blade
//to find average angular aceleration of fan blade

// GIVEN::

//initial angular velocity of fan blade
wi1 = 48.6//in revolution per minute
wi = wi1/60//in rev/s
//final angular velocity of fan blade
//as finally fan blade comes to rest
wf = 0//in revolution per minute
//time required for fan blade to come to rest
delta_t = 32//in seconds
//no. of revolution completed by fan blade before
   come to rest
delta_fi = 8.8


// SOLUTION:
```

```
21 //using kinematic equation of motion for rotational
     motion
22 //average angular velocity of fan blade
23 w_av = delta_fi/delta_t//in rev/s
24 //average angular aceleration of fan blade
25 a_av = (wf-wi)/delta_t//in rev/s^2
26
27 printf ("\n\n Average angular velocity of fan blade
     w_av = \n\n %.2f rev/s",w_av);
28 printf ("\n\n Average angular acceleration of fan
     blade a_av = \n\n %.3f rev/s^2",a_av);
```

Scilab code Exa 8.2 C8P2

```
1 clear
2 clc
3 //to find angular posion at t = 2 seconds
4 //to find instantaneous angular acceleration of
     reference line at t = 0.5 seconds
5 // GIVEN:
6
7 //refer to the figure 8-1 from page no. 159
8 //in angular velocity function w = A*t + B*t^2
     values of conatanta
9 A = 6.2//in rad/s^2
10 B = 8.7//in red/s^2
11 //for calculatiing angular position time interval
12 t1 = 2//in seconds
13 //for calculatiing angular acceleration time
     interval
14 t2 = 0.50//in seconds
15 //initial condition
16 //reference line initially is at fi = 0 when t = 0
17
18 // SOLUTION:
```

```
19 //using kinematic equation of motion for rotational
     motion
20 //angular posion at t = 2 seconds
21 fi = ((1/2)*A*t1^2) + ((1/3)*B*t1^3)//in rad
22 //angular instantaneous acceleration at t = 0.5
     seconds
23 a = A + (2*B*t2)//in rad/s^2
24
25 printf ("\n\n Angular posion at t = 2 seconds fi = \
     n\n %.1f rad",fi);
26 printf ("\n\n Angular instantaneous acceleration at
     t = 0.5 seconds a = \n\n %.1f rav/s^2",a);
```

Scilab code Exa 8.3 C8P3

```
1 clear
2 clc
3 //to find angular displacement of grindstone 2.7
     seconds later
4 //to find angular speed of grindstone 2.7 seconds
     later
5 // GIVEN:
6
7 //refer to the figure 8-7 from page no. 165
8 //constant angular acceleration of grindstone in +ve
     z direction
9 az = 3.2//in rad/s^2
10 //time initervalfor calculating angular acceleration
     and angular displacement
11 t = 2.7//in seconds
12 //initially angular displacement
13 fi_0 = 0//in rad
14 //initially angular velocity in +ve z direction
15 w0z = 0//in rad/s
16
```

```
17 // SOLUTION:
18 //consider angular velocity in +ve z direction
19 //using kinematic equation of motion for rotational
      motion
20 //angular displacement of grindstone 2.7 seconds
      later
21 fi = fi_0 + (w0z*t) + (0.5*az*t^2)//in rad
22 //angular speed of grindstone 2.7 seconds later
23 wz = w0z + (az*t)//in rad/s
24
25 printf ("\n\n Angular displacement of grindstone 2.7
       seconds later fi = \n\n %.1f rad",fi);
26 printf ("\n\n Angular speed of grindstone 2.7
      seconds later wz = \n\n %.1f rad/s",wz);
```

Scilab code Exa 8.4 C8P4

```
1 clear
2 clc
3 //to find angular acceleration of grindstone
4 //to find total angle turned through during slowing
      down of grindstone
5
6 // GIVEN:
7 //refer to problem 8-3 and from page no. 165
8 //refer to the figure 8-7 from page no. 165
9 //initial angular speed of grindstone
10 w0z = 8.6//in rad/s
11 //final angular speed of grindstone
12 //as grindstone comes to rest
13 wz = 0//in rad/s
14 //time interval in which grindstone comes to rest
15 t = 192//in seconds
16 //initial angular displacement of grindstone
17 fi_0 = 0//in rad
```

```
18
19 // SOLUTION:
20 //consider angular velocity in +ve z direction
21 //using kinematic equation of motion for rotational
      motion
22 //angular acceleration of grindstone
23 az = (wz-w0z)/t//in rad/s^2
24 //total angle turned through during slowing down of
      grindstone
25 fi = fi_0 + (w0z*t) + ((1/2)*az*(t^2))//in rad
26 fi = nearfloat("pred",823)
27
28 printf ("\n\n Angular acceleration of grindstone az
      = \n\n %.3f rad/s^2",az);
29 printf ("\n\n Total angle turned through during
      slowing down of grindstone fi = \n\n %.3i rad",fi
      );
```

Scilab code Exa 8.5 C8P5

```
1 clear
2 clc
3 //to find linesr or tangential speed of a point on a
      rim
4 //to find tangential acceleration of a point on a
      rim
5 //to find radial acceleration of a point on a rim
6
7 // GIVEN:
8 //refer to problem 8-3 from page no. 165
9 //refer to the figure 8-7 from page no. 165
10 //radius of grindstone for first case
11 r1 = 0.24//in meters
12 //radius of grindstone for second case
13 r2 = 0.12//in meters
```

```
14 //initial angular speed of grindstone
15 w = 8.6//in rad/s
16 //constant angular acceleration of grindstone
17 a = 3.2//in rad/s^2
18 //time interval
19 t = 2.7//in seconds
20
21 // SOLUTION:
22 //using kinematic equation of motion for rotational
      motion
23
24 //for r1 = 0.24m
25 //linesr or tangential speed of a point on a rim
26 vT = w*r1//in m/s
27 //tangential acceleration of a point on a rim
28 aT = a*r1//in m/s^2
29 //radial acceleration of a point on a rim
30 aR = w^2*r1//in m/s^2
31
32 //for r1 = 0.12m
33 //linesr or tangential speed of a point on a rim
34 v_T = w*r2//in m/s
35 //tangential acceleration of a point on a rim
36 a_T = a*r2//in m/s^2
37 //radial acceleration of a point on a rim
38 a_R = w^2*r2//in m/s^2
39 aR = round(aR)
40
41 printf ("\n\n Linesr or tangential speed of a point
      on a rim for r1 = 0.24m vT = \n\n %.1f m/s",vT);
42 printf ("\n\n Tangential acceleration of a point on
      a rim for r1 = 0.24m aT = \n\n %.2f m/s^2",aT);
43 printf ("\n\n Radial acceleration of a point on a
      rim for r1 = 0.24m aR = \n\n %2i m/s^2",aR);
44 printf ("\n\n Linesr or tangential speed of a point
      on a rim for r1 v_T = 0.12m v_T = \n\n %.1f m/s",
      v_T);
45 printf ("\n\n Tangential acceleration of a point on
```

```
    a rim for r1 = 0.12m a_T = \n\n %.2f m/s^2",a_T);
46 printf ("\n\n Radial acceleration of a point on a
    rim for r1 = 0.12m a_R = \n\n %.1f m/s^2",a_R);
```

Scilab code Exa 8.6 C8P6

```
1  clear
2  clc
3  //to find tangential speed of point on the equator
       of pulsar
4
5  // GIVEN:
6  //rotational period of pulsar
7  T = 0.033//in seconds
8  //radius of pulsar
9  r = 15//in km
10
11 // SOLUTION:
12 //using kinematic equation of motion for rotational
       motion
13 //angular speed
14 w = (2*3.14)/T//in rad/s
15 //tangential speed of point on the equator of pulsar
16 vT = w*r//in km/s
17
18 printf ("\n\n Angular speed w = \n\n %3i rad/s",w);
19 //answer of vT is slightly varying.But answer of
       scilab program and calculator is same
20 printf ("\n\n Tangential speed of point on the
       equator of pulsar vT = \n\n %4i km/s",vT);
```

Chapter 9

ROTATIONAL DYNAMICS

Scilab code Exa 9.1 C9P1

```
clear
clc
//To find magnitude of torque due to gravity about
   the pivot point o

// GIVEN::

//refer to figure 9-5 from page no. 178
//mass of body
m = 0.17//in kg
//length of rod
L = 1.25//in meters
//angle of pendulum with vertical
theta = 10//in degrees
//acceleration due to gravity
g = 9.8//in m/s^2

// SOLUTION:

//magnitude of torque
tow = L*m*g*sind(theta)//in N.m
```

```
21
22 printf ("\n\n Magnitude of torque tow = \n\n %.2f N.
      m" ,tow);
```

Scilab code Exa 9.2 C9P2

```
1  clear
2  clc
3  //To find rotational inertia
4  //to find angular acceleration
5
6  // GIVEN::
7
8  //refer to figure 9-9 from page no. 181
9  //mass of first partical
10 m1 = 2.3//in kg
11 //mass of second partical
12 m2 = 3.2//in kg
13 //mass of third partical
14 m3 = 1.5//in kg
15 //force applied to m2
16 F = 4.5//in N
17 //angle made by force with horizontal
18 theta = 30//in degrees
19
20 // SOLUTION:
21
22 //consider firstly the axis passes through m1
23 r1f = 0.0//in m
24 r2f = 3.0//in m
25 r3f = 4.0//in m
26 //rotational inertia about the axis
27 I1 = (m1*r1f^2)+(m2*r2f^2)+(m3*r3f^2)//in Kg.m^2
28
29 //consider secondly the axis passes through m2
```

```
30 r1s = 3.0//in m
31 r2s = 0.0//in m
32 r3s = 5.0//in m
33 //rotational inertia about the axis
34 I2 = (m1*r1s^2)+(m2*r2s^2)+(m3*r3s^2)//in Kg.m^2
35
36 //consider thirdly the axis passes through m3
37 r1t = 4.0//in m
38 r2t = 5.0//in m
39 r3t = 0.0//in m
40 //rotational inertia about the axis
41 I3 = (m1*r1t^2)+(m2*r2t^2)+(m3*r3t^2)//in Kg.m^2
42 I1 = round(I1)
43 I2 = round(I2)
44 I3 = round(I3)
45
46 //from figure fi
47 fi = asind(3/5)//in degrees
48 //angle between F and line connecting m3 and m2
49 fi1 = theta + fi//in degrees
50 //value of moment arm
51 r_perpendicular = r3s*sind(fi1)//in m
52 //magnitude of torque about m3
53 tow_z = r_perpendicular*F//in N.m
54 //using rotational inertia about axis through m3
55 //angular acceleration
56 az = -(tow_z)/I3//in rad/s^2
57
58 printf ("\n\n Rotational inertia about the axis when
       the axis passes through m1 is I1 = \n\n %2i Kg.m
     ^2",I1);
59 printf ("\n\n Rotational inertia about the axis when
       the axis passes through m2 is I2 = \n\n %2i Kg.m
     ^2",I2);
60 printf ("\n\n Rotational inertia about the axis when
       the axis passes through m3 is I3 = \n\n %3i Kg.m
     ^2",I3);
61 printf ("\n\n Magnitude of torque about m3 tow_z = \
```

```
    n\n %.1f N.m",tow_z);
62 printf ("\n\n Angular acceleration az = \n\n %.2f
    rad/s^2",az);
```

Scilab code Exa 9.3 C9P3

```
1
2 clear
3  clc
4 //To find rotational inertia
5
6 // GIVEN::
7
8 //refer to figure 9-9 from page no. 181
9 //mass of first partical
10 m1 = 2.3//in kg
11 //mass of second partical
12 m2 = 3.2//in kg
13 //mass of third partical
14 m3 = 1.5//in kg
15
16 // SOLUTION:
17 //locating center of mass
18
19 x1 = 0//in m
20 x2 = 0//in m
21 x3 = 4.0//in m
22 //x coordinate of center of mass
23 x_cm = (m1*x1+m2*x2+m3*x3)/(m1+m2+m3)//in m
24
25 y1 = 0//in m
26 y2 = 3.0//in m
27 y3 = 0//in m
28 //y coordinate of center of mass
29 y_cm = (m1*y1+m2*y2+m3*y3)/(m1+m2+m3)//in m
```

```
30 //squqred distance from center of mass to each of
     particals
31 //for first partical
32 r1_square = x_cm^2 + y_cm^2//in m^2
33 //for second partical
34 r2_square = x_cm^2 + (y2-y_cm)^2//in m^2
35 //for third partical
36 r3_square = (x3-x_cm)^2 + y_cm^2//in m^2
37 //rotational inertia
38 I_cm = (m1*r1_square+m2*r2_square+m3*r3_square)//in
     Kg.m^2
39
40 r2_square = nearfloat("succ",3.40)
41 r3_square = nearfloat("pred",11.74)
42 I_cm = ceil(I_cm)
43
44 printf ("\n\n x coordinate of center of mass x_cm =
     \n\n %.2f m",x_cm);
45 printf ("\n\n y coordinate of center of mass y_cm =
     \n\n %.2f m",y_cm);
46 printf ("\n\n Squqred distance from center of mass
     for first partical r1_square = \n\n %.2f m^2",
     r1_square);
47 printf ("\n\n Squqred distance from center of mass
     for second partical r2_square = \n\n %.2f m^2",
     r2_square);
48 printf ("\n\n Squqred distance from center of mass
     for third partical r3_square = \n\n %2i m^2",
     r3_square);
49 printf ("\n\n Rotational inertia I_cm = \n\n %.1f Kg
     .m^2",I_cm);
```

Scilab code Exa 9.6 C9P6

```
1
```

```
2 clear
3  clc
4 //to find forces that is scale reading
5
6
7 // GIVEN::
8
9 //refer to figure 9-22(a) from page no. 189
10 //mass od beam
11 m = 1.8//in kg
12 //massof block
13 M = 2.7//in kg
14 //acceleration due to gravity
15 g = 9.8//in m/s^2
16
17 // SOLUTION:
18
19 //refer to figure 9-22(b) from page no. 189
20 //consider our system as beam and block together
21 //equating net torque to zero
22 //force Fr
23 Fr = (g/4)*(M+2*m)//in N
24 //equating forces iny direction as 0 for
     equillibrium condition
25 //force F1
26 F1 = (M+m)*g - Fr//in N
27 F1 = round(F1)
28
29 printf ("\n\n Force Fr = \n\n %2i N",Fr);
30 printf ("\n\n Force F1 = \n\n %2i N",F1);
```

Scilab code Exa 9.7 C9P7

```
1
2 clear
```

```
3  clc
4 //to find forces exerted on the ladder by the ground
      and by the wall
5
6
7 // GIVEN::
8
9 //refer to figure 9-23(a) from page no. 189
10 //length of ladder
11 L = 12//in meters
12 //mass of ladder
13 m = 45//in kg
14 //distance of upper end of ladder above the ground
15 h = 9.3//in meters
16 //mass of firefighter
17 M = 72//in kg
18 //acceleration due to gravity
19 g = 9.8//in m/s^2
20
21 // SOLUTION:
22
23 //refer to figure 9-23(b) from page no. 189
24 //distance from the wall to the foot of ladder
25 a = sqrt(L^2 - h^2)//in meters
26 //considering equillibrium conditions
27 //finding normal reaction by ground
28 N = (M+m)*g//in N
29 //force exerted on ladder by the wall
30 Fw = (g*a*(M/2 + m/3))/h//in N
31 N = round(N)
32 Fw = round(Fw)
33 printf ("\n\n Distance from the wall to the foot of
      ladder a = \n\n %.1f m",a);
34 //answer is slightly different than book.But answer
      of scilab program is same as that of calculator
35 printf ("\n\n Forces exerted on the ladder by the
      ground N = \n\n %3i N",N);
36 //answer is slightly different than book.But answer
```

```
    of scilab program is same as that of calculator
printf ("\n\n Forces exerted on the ladder by the
    wall Fw = \n\n %3i N",Fw);
```

Scilab code Exa 9.8 C9P8

```
1
2 clear
3  clc
4 //to find tension in the wire
5 //to find force exerted by the hinge on the beam
6
7
8 // GIVEN::
9
10 //refer to figure 9-24(a) from page no. 190
11 //length of the beam
12 L = 3.3//in meters
13 //mass of beam
14 m = 8.5//in kg
15 //distance at which wire is connected
16 d = 2.1//in meters
17 //angle made by beam with horizontal
18 theta = 30//in degrees
19 //mass of body
20 M = 56//in kg
21 //acceleration due to gravity
22 g = 9.8//in m/s^2
23
24 // SOLUTION:
25
26 //refer to figure 9-24(b) from page no. 190
27 //angle alpha from geometry
28 alpha = atand((d-(L*sind(theta)))/(L*cosd(theta)))//
    in degrees
```

```
29 k = M*g+m*g;
30 j = m*g/2;
31 //applying equllibrium conditions to get 4 equations
32 A = [0 1 0 -1 ; 1 0 1 0 ; 1 -tand(theta) 0 0 ; 0 0 1
       -tand(alpha)];
33 b = [0 ; k ; j ; 0];
34 c = A\b
35 Fv = c(1)
36 Fh = c(2)
37 Tv = c(3)
38 Th = c(4)
39
40 Fv = round(Fv)
41 Fh = round(Fh)
42 Th = round(Th)
43 //resultant tension in the wire
44 T = sqrt(Th^2 + Tv^2)//in N
45 //resultant force exerted by the hinge on the beam
46 F = sqrt(Fh^2+ Fv^2)//in N
47 T = round(T)
48 F = round(F)
49 //angle made by vector F with horizontal
50 fi = atand(Fv/Fh)//in degrees
51
52 printf ("\n\n Vertical force Fv = \n\n %3i N",Fv);
53 printf ("\n\n Horizontal force Fh = \n\n %3i N",Fh);
54 printf ("\n\n vertical tension in in wire Tv = \n\n
      %3i N",Tv);
55 printf ("\n\n Horizontal tension in in wire Th = \n\
      n %3i N",Th);
56 printf ("\n\n Resultant tension in the wire T = \n\n
       %3i N",T);
57 printf ("\n\n Resultant force exerted by the hinge
      on the beam F = \n\n %3i N",F);
58 printf ("\n\n angle made by vector F with horizontal
       fi = \n\n %.1f degrees",fi);
```

Scilab code Exa 9.9 C9P9

```
1
2 clear
3  clc
4 //to find magnitude of torque
5 //to find resultant angular acceleration of the
     system
6
7
8 // GIVEN::
9
10 //refer to figure 9-25 from page no. 191
11 //force exerted
12 F = 115//in N
13 //distance from axis of rotation at which force is
     exerted
14 r = 1.50//in meters
15 //angle of apllication of force
16 theta1 = 32//in degrees
17 //direction of horizontal component
18 theta2 = 15//in degrees
19 //acceleration due to gravity
20 g = 9.8//in m/s^2
21 //radius od disk
22 R = 1.5//in meters
23 //thicknes of disk
24 d = 0.40//in cm
25 //mass of child
26 m = 25//in kg
27 //radius of position of child
28 r1 = 1.0//in meters
29
30
```

```
31 // SOLUTION:
32
33 //refer to figure 9-25 from page no. 191
34 //horizontal component of force
35 Fh = F*cosd(theta1)//in N
36 //component of force perpendicular to r
37 F_perpendicular = Fh*cosd(theta2)//in N
38 //vertical torque along the axis of rotation
39 tow = r*F_perpendicular//in N.m
40
41 //volume of disk
42 volume = %pi*(R*100)^2*d//in m^3
43 //consider density of steel
44 density = 7.9//in g/cm^3
45 //mass of merry-go-round
46 M = (volume*density)*10^-3//in kg
47 //rotational inertia of disk
48 Im = ((1/2)*M*R^2)///in kg.m^2
49 //rotational inertia of child
50 Ic = m*r1^2///in kg.m^2
51 //total rotational inertia
52 It = Im + Ic//in kg.m^2
53 //angular acceleration of the system
54 alpha_z = tow/It//in rad/s^2
55
56 printf ("\n\n Horizontal component of force Fh = \n\
      n %.1f N",Fh);
57 printf ("\n\n Component of force perpendicular to r
      F_perpendicular  = \n\n %.1f N",F_perpendicular);
58 printf ("\n\n Vertical torque along the axis of
      rotation tow = \n\n %3i N.m",tow);
59 printf ("\n\n Rotational inertia of disk Im = \n\n
      %3i kg.m^2",Im);
60 printf ("\n\n Rotational inertia of child Ic = \n\n
      %3i kg.m^2",Ic);
61 printf ("\n\n Total rotational inertia It = \n\n %3i
       kg.m^2",It);
62 printf ("\n\n Angular acceleration of the system
```

```
alpha_z = \n\n %.2f rad/s^2",alpha_z);
```

Scilab code Exa 9.10 C9P10

```
1
2 clear
3  clc
4 //to find acceleration of the falling block
5 //to find tension in the chord
6 //to find angular acceleration of the disk
7
8
9 // GIVEN::
10
11 //refer to figure 9-26(a) from page no. 192
12 //mass of disk
13 M = 2.5//in kg
14 //radius of disk
15 R = 20//in cm
16 //mass of block
17 m = 1.2//in kg
18 //acceleration due to gravity
19 g = 9.8//in m/s^2
20
21 // SOLUTION:
22
23 //refer to figure 9-26(b) from page no. 192
24 //applying newton's second law in y direction for
     block
25 //and applying rotational form of newton's second
     law for disk
26 //we get 2 equations and 2 unknowns
27 A = [m 1;(1/2*M) -1]
28 B = [(m*g);0]
29 c = A\B
```

```
30 //acceleration of block
31 a = c(1)//in m/s^2
32 //tension in the string
33 T = c(2)//in N
34 //angular acceleration of disk
35 az = a/(R*10^-2)//in rad/s^2
36 a_z = az/(2*%pi)//in rev/s^2
37
38 printf ("\n\n Acceleration of block a = \n\n %.1f m/
      s^2",a);
39 printf ("\n\n Tension in the string T = \n\n %.1f N"
      ,T);
40 printf ("\n\n Angular acceleration of disk az in rad
      /s^2 = \n\n %.1f rad/s^2",az);
41 printf ("\n\n Angular acceleration of disk a_z in
      rev/s^2 = \n\n %.1f rev/s^2",a_z);
```

Scilab code Exa 9.12 C9P12

```
1
2 clear
3  clc
4 //to find velocity of center of mass at time t
5 //to find value of t
6
7 // GIVEN::
8
9 //refer to figure 9-33(a) from page no. 192
10 //radius of solid cylinder
11 R = 12//in cm
12 //mass of solid cylinder
13 M = 3.2//in kg
14 //initial angular velocity of solid cylinder
15 w0 = 15//in rev/s
16 //coefficient of kinetic friction between surface
```

```
    and cylinder
mew_k = 0.21
//acceleration due to gravity
g = 9.8//in m/s^2

// SOLUTION:

//refer to figure 9-33(b) from page no. 192
w_0 = w0*2*%pi//in rad/rev
//applying newton's second law in x direction
//and applying rotational form of newton's second
    law
//velocity of center of mass
vcm = (1/3*w_0*(R*10^-2))//in m/s
//value of t
t = vcm/(mew_k*g)//in seconds

printf ("\n\n Velocity of center of mass vcm = \n\n
    %.1f m/s",vcm);
printf ("\n\n Value of t = \n\n %.1f seconds",t);
```

Scilab code Exa 9.13 C9P13

```

clear
 clc
//to find rotational velocity when it reaches end of
    the string

// GIVEN::

//refer to figure 9-34(a) from page no. 196
//total mass of yo-yo
M = 0.24//in kg
//radius of disk
```

```
12 R = 2.8//in cm
13 //radius of shaft
14 R0 = 0.25//in cm
15 //length of the string
16 L = 1.2//in meters
17 //initial velocity of yo-yo
18 v0 = 1.4//in m/s
19 //acceleration due to gravity
20 g = 9.8//in m/s^2
21
22 // SOLUTION:
23
24 //refer to figure 9-34(b) from page no. 196
25 //momemt of inertia
26 I = (1/2*(M*R^2))
27 //applying newton's second law
28 //and applying rotational form of newton's second
     law
29 //angular acceleration
30 az = (g*100/R0)*(1/(1+R^2/(2*R0^2)))//in rad/s^2
31 //angle through which yo-yo rotates
32 fi = L/(R0*10^-2)//in rad
33 //initial angular velocity
34 w0z = v0/(R0*10^-2)//in rad/s
35 //solving using equation to find out time
36 y = poly([-fi w0z (1/2*az)],'t','coeff')
37 c = roots(y)
38 //taking only positive value as it is time
39 t2 = c(2)//in seconds
40 //rotational velocity when it reaches end of the
     string
41 wz = w0z+(az*t2)//in rad/s^2
42
43 printf ("\n\n Angular acceleration az = \n\n %.1f
     rad/s^2",az);
44 printf ("\n\n Time for calculating rotational
     velocity  t2 = \n\n %.2f seconds",t2);
45 printf ("\n\n initial angular velocity w0z = \n\n
```

```
    %3i rad/s",w0z);
46 printf ("\n\n Rotational velocity when it reaches
    end of the string wz = \n\n %3i rad/s^2",wz);
```

Chapter 10

ANGULAR MOMENTUM

Scilab code Exa 10.2 C10P2

```
1 clear
2 clc
3 //To find which magnitude is greater
4 //angular momentum of earth associated with its
     rotation on its axis
5                                 //OR
6 //angular momentum of earth associated with its
     orbital motion around the sun
7
8 //Given:
9 //refer to figure 10-8 from page no. 213
10 //rotation period of the earth about its axis in
     hour
11 t1 = 24//in hour
12 //rotation period of earth about its axis in seconds
13 T1 = (t1*60*60)//in seconds
14 //T2 is time required by earth to complete one
     revolution around the sun
15 T2 = 3.16*10^7//in seconds
16 //mass of the earth
17 M = 5.98*10^24//in kg
```

```
18 //radius of the earth
19 RE = 6.37*10^6//in meters
20
21 //Solution:
22 //considering earth as a uniform sphere mmoment of
      inertia
23 I = (2/5)*M*RE^2
24 //angular speed
25 w1 = (2*3.14)/T1//in per seconds
26 //angular momentum of earth associated with its
      rotation
27 L_rot = I*w1//in kg m^2/s
28 //radius of orbit
29 R_orb = 1.50*10^11//in meters
30
31 //angular speed
32 w2 = (2*3.14)/T2//in per second
33 //velocity of rotation of earth around the sun
34 v = w2*R_orb//in m/s
35 //linear momentum
36 p = M*v
37 //angular momentum of earth associated with its
      orbital motion around the sun
38 L_orb = R_orb*p//in  kg m^2/s
39
40 printf ("\n\n Angular momentum of earth associated
      with its rotation on its axis is L_rot = \n\n %.2
      e kg m^2/s" ,L_rot);
41 printf ("\n\n Angular momentum of earth associated
      with its orbital motion around the sun L_orb = \n
      \n %.2e kg m^2/s",L_orb);
42 if (L_rot>L_orb) then
43     printf('\n\n Angular momentum of earth
          associated with its rotation on its axis is
          greater than angular momentum of earth
          associated with its orbital motion around the
           suns');
44 else
```

```
45      printf('\n\n Angular momentum of earth
           associated with its orbital motion around
           the sun is greater than angular momentum of
           earth associated with its rotation on its
           axis');
46  end
```

Scilab code Exa 10.4 C10P4

```
1  clear
2  clc
3  //to find centripital force austronautshould apply
       at distance 50 m from spacecraft
4  //to find centripital force austronautshould apply
       at distance 5 m from spacecraft
5
6  //Given:
7  //mass of austronaut
8  M = 120//in kg
9  //length of cord
10 ri = 180//in meters
11 // initial tangential velocity acquired by astronaut
12 vi = 2.5//in m/s
13
14 //Solution:
15 //appiying conservation of angular momentum
16 //initially required centripital force
17 F = (M*vi^2)/ri//in N
18 //when astonaut is at a distance of 50 m from
       spacecraft
19 r1 = 50//in meters
20 //velocity at this stage
21 v = (vi*ri)/r1//in m/s
22 //centripital force
23 f = (M*v^2)/r1//in N
```

```
24
25 printf ("\n\n Initially required centripital force F
       = \n\n %.1f N" ,F);
26 printf ("\n\n Tangential speed v = \n\n %.1f m/s" ,v
   );
27 printf ("\n\n Centripital force austronautshould
   apply at distance 50 m from spacecraft f = \n\n
   %3i N",f);
```

Scilab code Exa 10.5 C10P5

```
1 clear
2 clc
3 //to find angular speed of combination of disk
4
5 //Given:
6 //refer to figure 10-17(a)and(b) from page no. 219
7 //mass of disk
8 M = 125//in g
9 //radius of disk
10 r = 7.2//in centimeters
11 // initial angular speed of disc about vertical axis
12 omega_i = 0.84//in rev/s
13
14 //Solution:
15 //completely inelastic collision.
16 //appllying conservation of angular momentum
17 //ratio of rotational inertia of disks
18 R = (1/3)
19 //angular speed of combination of disk
20 omega_f = omega_i*(R)//in rev/s
21
22 printf ("\n\n Angular speed of combination of disk
   omega_f = \n\n %.2f rev/s" ,omega_f);
```

Chapter 11

ENERGY 1 WORK AND KINETIC ENERGY

Scilab code Exa 11.1 C11P1

```
1
2 clear
3  clc
4 //to find work done
5
6 // GIVEN::
7
8 //refer to figure 11-8(a) from page no. 232
9 //mass of block
10 m = 11.7//in kg
11 //distance by which block is pushed on inclined
     plane
12 s = 4.65//in meters
13 //height by which block is raised
14 h = 2.86//in meters
15 //acceleration due to gravity
16 g = 9.8//in m/s^2
17
18 // SOLUTION:
```

```
19
20 //refer to figure 11-8(b) from page no. 232
21 //from diagram sin(theta) can be calculated as
22 sin_theta = (h/s)
23 //angle between applied force and displacement of
      block
24 fi = 0//in degrees
25 //using newton's second law of motion
26 //force pushing the block
27 F = m*g*sin_theta//in N
28 //work done by force F
29 W = F*s*cosd(fi)//in J
30 //work done by raising block vertically
31 Work = m*g*h//in J
32 W = round(W)
33 Work = round(Work)
34 printf ("\n\n Force pushing the block F = \n\n %.1f
      N",F);
35 printf ("\n\n Work done by force F W = \n\n %3i J",W
      );
36 printf ("\n\n Work done by raising block vertically
      \n\n Work = \n\n %3i J",Work);
```

Scilab code Exa 11.2 C11P2

```
1
2 clear
3  clc
4 //to find work done by the chid
5
6 // GIVEN::
7
8 //refer to figure 11-9(a) from page no. 233
9 //mass of sled
10 m = 5.6//in kg
```

```
11 //distance by which sled is pushed horizontally
12 s = 12//in meters
13 //coefficient of kinetic friction
14 mew_k = 0.20
15 //angle made by the rope with horizontal
16 fi = 45//in degrees
17 //acceleration due to gravity
18 g = 9.8//in m/s^2
19
20 // SOLUTION:
21
22 //refer to figure 11-9(b) from page no. 233
23 //using newton's second law of motion
24 //we get three equations and three unknowns
25 A = [cosd(fi) -1 0;sind(fi) 0 1;0 1 -mew_k]
26 B = [0; m*g; 0]
27 c = A\B
28 //force applied by the child
29 F = c(1)//in N
30 //frictional force
31 f = c(2)//in N
32 //normal reaction
33 N = c(3)//in N
34 //work done by the child
35 W = F*s*cosd(fi)//in J
36
37
38 F = round(F)
39 W = round(W)
40 printf ("\n\n Force applied by the child F = \n\n
     %2i N",F);
41 printf ("\n\n Work done by the child W = \n\n %3i J"
     ,W);
```

Scilab code Exa 11.3 C11P3

```
1
2 clear
3  clc
4 //to find average power must be applied by the
     elevator motor
5
6 // GIVEN::
7
8 //weight of elevator
9 w = 5160//in N
10 //average weight of passenger
11 wp = 710//in N
12 //number of passengers
13 n = 20
14 //distance between floors
15 sf = 3.5//in meters
16 //time elasped
17 t = 18//in seconds
18 //acceleration due to gravity
19 g = 9.8//in m/s^2
20
21 // SOLUTION:
22
23 //total weight of elevator and passenger
24 //upward force exerted by motor
25 F = w+n*wp//in N
26 //total height by which elevator moves
27 s = sf*25//in meters
28 //work done must be applied by the elevator motor
29 W = F*s//in J
30 //average power
31 Pav = (W/t)*10^-3//in kW
32
33 //value of force F is slightly different than scilab
     answer
34 //but silab answer is same as calculator answer
35 printf ("\n\n Upward force exerted by motor F = \n\n
     %5i N",F);
```

```
36 printf ("\n\n Work done must be applied by the
      elevator motor W  = \n\n %.1e J",W);
37 printf ("\n\n Average power Pav = \n\n %2i kW",Pav);
```

Scilab code Exa 11.4 C11P4

```
1
2 clear
3  clc
4 //to find work done by gravity
5 //to find work done by the spring
6 //to find work done by the hand
7
8
9 // GIVEN::
10
11 //refer to figure 11-15(a) from page no. 237
12 //mass of block
13 m = 6.40//in kg
14 //distance streched by spring
15 d = 0.124//in meters
16 //acceleration due to gravity
17 g = 9.8//in m/s^2
18
19 // SOLUTION:
20
21 //refer to figure 11-8(b)and 11-5(c) from page no.
      237
22 //applying equillibrium condition in y direction
23 //force constant of spring
24 k = m*g/d//in N/m
25 //work done by gravity
26 Wg = m*g*d//in J
27 //work done by the spring
28 Ws = (-1/2)*k*d^2//in J
```

```
29 //-ve sign as force and displacement are in opposite
       directions
30 //work done by the hand
31 //intergrating force in y direction
32 Wh = m*g*(-d)+(1/2)*k*(-d)^2//in J
33 k = round(k)
34 printf ("\n\n Force constant of spring k = \n\n %3i
       N/m",k);
35 printf ("\n\n Work done by gravity Wg = \n\n %.2f J"
       ,Wg);
36 printf ("\n\n Work done by the spring Ws = \n\n %.2f
       J",Ws);
37 printf ("\n\n Work done by the hand Wh = \n\n %.2f J
       ",Wh);
```

Scilab code Exa 11.6 C11P6

```
1
2 clear
3  clc
4 //to find kinetic energy
5
6 // GIVEN::
7
8 //distance travelled by neutron
9 d = 6.2//in meters
10 //time for neutron travel
11 t = 160//in micrometers
12 //mass of neutron
13 m = 1.67e-27//in kg
14
15 // SOLUTION:
16
17 //speed of neutron
18 v = d/(t*10^-6)//in m/s
```

```
19 //applying formula for kinetic energy
20 //kinetic energy of neutron
21 K = (1/2)*m*v^2//in J
22 K1 = K*(6.242e18)//in eV
23 K = nearfloat("succ",1.26e-18)
24 K1 = nearfloat("succ",7.9)
25
26 printf ("\n\n Speed of neutron v = \n\n %.2e m/s",v)
   ;
27 printf ("\n\n Kinetic energy of neutron in J K = \n
   \n %.2e J",K);
28 printf ("\n\n Kinetic energy of neutron in eV K = \
   n\n %.1f eV",K1);
```

Scilab code Exa 11.7 C11P7

```
1
2 clear
3  clc
4 //to find speed of body when it strikes the ground
5
6 // GIVEN::
7 //mass of body
8 m = 4.5//in kg
9 //height from which body is dropped
10 h = 10.5//in meters
11 //acceleration due to gravity
12 g = 9.80//in m/s^2
13
14 // SOLUTION:
15 //using work-energy principle
16 //speed of body when it strikes the ground
17 v = sqrt(2*g*h)//in m/s
18 printf ("\n\n Speed of body when it strikes the
   ground v = \n\n %.1f m/s",v);
```

Scilab code Exa 11.8 C11P8

```
1
2 clear
3  clc
4 //to find spring compression
5
6 // GIVEN::
7 //mass of body
8 m = 3.63//in kg
9 //speed of block
10 v = 1.22//in m/s
11 //force constant for spring
12 k = 135//in
13
14 // SOLUTION:
15 //using work-energy principle
16 //spring compression
17 d = v*sqrt(m/k)//in meters
18 d1 = d*10^2//in
19 printf ("\n\n Spring compression d = \n\n %.3f m",d)
    ;
20 printf ("\n\n Spring compression d = \n\n %.1f cm",
    d1);
```

Scilab code Exa 11.9 C11P9

```
1
2 clear
3  clc
4 //to find speed of crate according to observer o
```

```
5 ////to find work and change in kinetic energy
6
7 // GIVEN:
8 //refer to figure 11-18(a),(b)from page no. 242
9 //force applied
10 Fx = 5.63//in N
11 //mass of crate
12 m = 12.0//in kg
13 //speed of train
14 vx = 15.0//in m/s
15 //distance travelled by crate
16 s = 2.4//in meters
17
18 // SOLUTION:
19 //using work-energy principle
20 //work done
21 W = Fx*s//in J
22 //initial kinetic energy according to observer in
     car
23 Ki = 0
24 ////final kinetic energy according to observer in
     car
25 Kf = W -Ki
26 //speed of crate according to observer o
27 vf = sqrt(2*Kf/m)//in m/s
28 //applying impulse-momentum theorem
29 //time interval
30 delta_t = (m*vf/Fx)//in seconds
31 //forward distance travelled
32 d = vx*delta_t//in meters
33 //total distance moved by crate
34 s_dash = d+s//in meters
35 //work done
36 W_dash = Fx*s_dash//in J
37 //final speed of crate
38 vf_dash = vx+vf//in m/s
39 //change in kinetic energy
40 deltaK_dash = (1/2*m*(vf_dash^2))-(1/2*m*(vx^2))
```

```
41 W_dash = round(W_dash)
42 deltaK_dash = round(deltaK_dash)
43 printf ("\n\n Final kinetic energy according to
      observer in car Kf = \n\n %.1f J",Kf);
44 printf ("\n\n Speed of crate according to observer o
       vf = \n\n %.2f m/s",vf);
45 printf ("\n\n Time interval delta_t = \n\n %.2f
      seconds",delta_t);
46 printf ("\n\n Work done W_dash = \n\n %3i J",W_dash)
      ;
47 printf ("\n\n Change in kinetic energy deltaK_dash =
       \n\n %3i J",deltaK_dash);
48 printf ("\n\n As W_dash = deltaK_dash work-energy
      principle is valid")
```

Scilab code Exa 11.10 C11P10

```
1
2 clear
3  clc
4 //to find conatance force to be applied
5
6 // GIVEN:
7 //refer to figure 11-21 from page no. 244
8 //initial angular velocity of spacecraft
9 wi = 2.4//in rev/s
10 //radius of spacecraft
11 R = 1.7//in meters
12 //mass of spacecraft
13 M = 245//in Kg
14 //final angular velocity of spacecraft
15 wf = 1.7//in rev/s
16 //rotation of spacecraft
17 theta = 3//in revolutions
18
```

```
19
20 // SOLUTION:
21
22 //moment of inertia of spacecraft
23 I = (2/3*M*R^2)//in Kg.m^2
24 //change in rotational kinetic energy
25 delta_k_dash = (1/2*I*(2*%pi*wf)^2)-(1/2*I*(2*%pi*wi
     )^2)//in J
26 //using work-energy principle
27 //work done = change in rotational kinetic energy
28 //thruster force F
29 F = (delta_k_dash/(-R*theta*2*%pi))//in N
30 F = nearfloat("pred",834)
31 printf ("\n\n Moment of inertia of spacecraft I = \n
     \n %3i Kg.m^2",I);
32 printf ("\n\n Change in rotational kinetic energy
     delta_k-dash = \n\n %.2e J",delta_k_dash);
33 printf ("\n\n Thruster force F = \n\n %3i N",F);
```

Scilab code Exa 11.11 C11P11

```
1
2 clear
3  clc
4 //to find kinetic energy lost by neutron
5
6 // GIVEN:
7
8 //initial kinetic energy of neutron
9 K1i = 5.0//in MeV
10 //mass of neuron mn
11 mn = 1//considering it as unity as other masses are
     given with reference to mn
12 //mass of neucleus of lead
13 mPb = 206*mn
```

```
14 //mass of neucleus of carbon
15 mC = 12*mn
16 //mass of neucleus of hydrogen
17 mH = mn
18
19 // SOLUTION:
20
21 //As collision is elastic collision
22 //using conservation of energy principle
23
24 //collision with neucleus of lead
25 //final kinetic energy of neutron
26 K1f = K1i*((mn-mPb)/(mn+mPb))^2//in MeV
27 //kinetic energy lost by neutron
28 K_lostl = K1i-K1f//in MeV
29
30
31 //collision with neucleus of carbon
32 //final kinetic energy of neutron
33 K1f_C = K1i*((mn-mC)/(mn+mC))^2//in MeV
34 //kinetic energy lost by neutron
35 K_lostC = K1i-K1f_C//in MeV
36
37
38 //collision with neucleus of lead
39 //final kinetic energy of neutron
40 K1f_H = K1i*((mn-mH)/(mn+mH))^2//in MeV
41 //kinetic energy lost by neutron
42 K_lostH = K1i-K1f_H//in MeV
43
44 printf ("\n\n Collision with neucleus of lead")
45 printf ("\n\n Final kinetic energy of neutron K1f =
     \n\n %.1f MeV",K1f);
46 printf ("\n\n Kinetic energy lost by neutron K_lostl
     = \n\n %.1f MeV",K_lostl);
47 printf ("\n\n Collision with neucleus of carbon")
48 printf ("\n\n Final kinetic energy of neutron K1f_C
     = \n\n %.1f MeV",K1f_C);
```

```
49 printf ("\n\n Kinetic energy lost by neutron K_lostC
       = \n\n %.1f MeV",K_lostC);
50 printf ("\n\n Collision with neucleus of hydrogen")
51 printf ("\n\n Final kinetic energy of neutron K1f_H
     = \n\n %.1f MeV",K1f_H);
52 printf ("\n\n Kinetic energy lost by neutron K_lostH
       = \n\n %.1f MeV",K_lostH);
```

Scilab code Exa 11.12 C11P12

```
1
2 clear
3  clc
4 //to find initial speed of bullet
5 //to find lost in kinetic energy
6
7 // GIVEN:
8 //refer to figure 11-23 from page no. 246
9 //mass of block
10 M = 5.4//in Kg
11 //mass of bullet
12 m = 9.5e-3//in Kg
13 //height to which block rises
14 h = 6.3e-2//in meters
15 //acceleration due to gravity
16 g = 9.8//in m/s^2
17
18 // SOLUTION:
19
20 //applying work-energy principle
21 //initial speed of bullet
22 vi = ((M+m)/m)*(sqrt(2*g*h))//in m/s
23 //ratio of final to initial kinetic enerdy
24 Kf_by_Ki = (m/(M+m))
25 //initialkinetic energy remains after collision
```

```
26 Kr = (Kf_by_Ki)*100//in percentage
27 //kinetic energy stored inside pendullum
28 Ks = 100-Kr//in percentage
29 //answer of vi is slightly different than textbook.
     but answer by calculator is same as that of
     scilab
30 printf ("\n\n Initial speed of bullet vi = \n\n %3i
     m/s",vi);
31 printf ("\n\n Ratio of final to initial kinetic
     enerdy Kf/Ki = \n\n %.4f ",Kf_by_Ki);
32 printf ("\n\n Initial kinetic energy remains after
     collision Kr = \n\n %.2f percent",Kr);
33 printf ("\n\n Kinetic energy stored inside pendullum
     Ks = \n\n %.2f percent",Ks);
```

Chapter 12

ENERGY 2 POTENTIAL ENERGY

Scilab code Exa 12.1 C12P1

```
1
2 clear
3  clc
4 //to find change in gravitational potential energy
5
6 // GIVEN:
7 //mass of elevator
8 m = 920//in Kg
9 //height above the groung
10 h = 412//in meters
11 //acceleration due to gravity
12 g = 9.8//in m/s^2
13
14 // SOLUTION:
15 //applying potential energy formula
16 //change in gravitational potential energy
17 delta_U = m*g*h//in J
18 delta_U1 = delta_U*10^-6//in MJ
19
```

```
20 printf ("\n\n Change in gravitational potential
     energy delta_U = \n\n %.1e J",delta_U);
21 printf ("\n\n Change in gravitational potential
     energy delta_U1 = \n\n %.1f MJ",delta_U1);
```

Scilab code Exa 12.2 C12P2

```
1
2 clear
3  clc
4 //to find potential energy stored in the spring
5
6 // GIVEN:
7 //foce constant of spring
8 k = 1.25e8//in N/m
9 //compression in spring
10 x = 5.6e-2//in meters
11
12 // SOLUTION:
13 //applying spring force formula
14 //potential energy stored in the spring
15 U = (1/2*k*x^2)//in J
16 printf ("\n\n Potential energy stored in the spring
     U = \n\n %.2e J",U)
```

Scilab code Exa 12.3 C12P3

```
1
2 clear
3  clc
4 //to find speed of ball
5
6 // GIVEN:
```

```
7 //refer to figure 12-1
8 //compression in spring
9 d = 3.2e-2//in meters
10 //mass of ball
11 m = 12e-3//in Kg
12 //force constant of spring
13 k = 7.5//in N/cm
14
15 // SOLUTION:
16 //applying conservation of energy principle
17 //speed of ball
18 vm = d*sqrt((k*10^2)/m)//in m/s
19
20 printf ("\n\n Speed of ball vm = \n\n %.1f m/s",vm)
```

Scilab code Exa 12.4 C12P4

```
1
2
3 clear
4  clc
5 //to find speed of ball
6
7 // GIVEN:
8 //refer to figure 12-6 on page no. 263
9 //lift of car
10 y = 25//in meters
11 //acceleration due to gravity
12 g = 9.8//in m/s^2
13
14 // SOLUTION:
15 //applying conservation of energy principle
16 //speed of car
17 v = sqrt(2*g*y)//in m/s
18 printf ("\n\n Speed of car v = \n\n %2i m/s",v)
```

Scilab code Exa 12.7 C12P7

```
1
2 clear
3  clc
4 //to find speed of ball
5
6 // GIVEN:
7 //refer to problem 9-10
8 //mass of disk
9 M = 2.5//in kg
10 //distance of fall
11 y = 0.56//in meters
12 //mass of block
13 m = 1.2//in kg
14 //acceleration due to gravity
15 g = 9.8//in m/s^2
16
17 // SOLUTION:
18 //applying conservation of mechanocal energy
      principle
19 //speed of block
20 v = sqrt((4*m*g*y)/(M+2*m))//in m/s
21 printf ("\n\n Speed of ball v = \n\n %.1f m/s",v)
```

Chapter 13

ENERGY 3 CONSERVATION OF ENERGY

Scilab code Exa 13.1 C13P1

```
1
2 clear
3  clc
4 //to find change in internal energy
5
6 // GIVEN:
7 //mass of baseball
8 m = 0.143//in kg
9 //height of tower
10 h = 443//in m
11 //terminal velocity
12 v = 42//in m/s
13 //acceleration due to gravity
14 g = 9.8//in m/s^2
15
16 // SOLUTION:
17
18 //initial potential energy
19 Ui = m*g*h//in J
```

```
20 //final potential energy
21 Uf = 0//in J
22 //change in potential energy
23 delta_U = (Uf-Ui)//in J
24 //final kinetic energy
25 Kf = (1/2)*(m*v^2)//in J
26 //initial kinetic energy
27 Ki = 0//in J
28 //change in kinetic energy
29 delta_K = (Kf-Ki)//in J
30 //applying conservation of energy principle
31 //change in internal energy
32 delta_Eint = (-delta_U-delta_K)//in J
33 delta_U = round (Uf-Ui)
34 delta_K = round(Kf-Ki)
35 delta_Eint = round(-delta_U-delta_K)
36
37 printf ("\n\n Change in potential energy delta_U  =
     \n\n %3i J",delta_U)
38 printf ("\n\n Change in kinetic energy delta_K  = \n
     \n %3i J",delta_K)
39 printf ("\n\n Change in internal energy delta_Eint
     = \n\n %3i J",delta_Eint)
```

Scilab code Exa 13.2 C13P2

```
1
2 clear
3  clc
4  //to find gain in internal energy
5 //to find speed of block
6
7
8 // GIVEN:
9 //mass of block
```

```
10 m = 4.5//in Kg
11 //angle of inclination
12 theta = 30//in degrees
13 //initial speed
14 v = 5.0//in m/s
15 //distance travelled
16 d = 1.5//in meters
17 //acceleration due to gravity
18 g = 9.8//in m/s^2
19
20 // SOLUTION:
21 //applying conservation of energy principle
22 //consider block+plane+earth as our system
23 //final potential energy
24 Uf = m*g*(d*sind(theta))//in J
25 //initial potential energy
26 Ui = 0//in J
27 //change in potential energy
28 delta_U = Uf-Ui//in J
29 //final kinetic energy
30 Kf = 0//in J
31 //initial kinetic energy
32 Ki =(1/2)*m*v^2//in J
33 //change in kinetic energy
34 delta_K = Kf-Ki//in J
35 //change in mechanical energy in system
36 delta_U_plus_delta_K = delta_U+delta_K//in J
37 //applying conservation of energy principle
38 //gain in internal energu
39 delta_E_int = -(delta_U_plus_delta_K)//in J
40 //final kinetic energy for downhill journy
41 //here delta_K = 2*delta_E_int as round tripi.e.
     uphill and downhill motion
42 KF = (-(2*delta_E_int))+(-delta_K)//in J
43 //speed of block
44 vf = sqrt(2*KF/m)//in m/s
45 KF = round(KF)
46
```

```
47 printf ("\n\n Change in potential energy delta_U  =
     \n\n %2i J",delta_U)
48 printf ("\n\n Change in kinetic energy delta_K  = \n
     \n %2i J",delta_K)
49 printf ("\n\n Change in mechanical energy in system
     delta_U_plus_delta_K   = \n\n %2i J",
     delta_U_plus_delta_K)
50 printf ("\n\n Gain in internal energy delta_E_int  =
      \n\n %2i J",delta_E_int)
51 printf ("\n\n Final kinetic energy for downhill
     journy KF  = \n\n %2i J",KF)
52 printf ("\n\n Speed of block vf = \n\n %.1f m/s",vf)
```

Scilab code Exa 13.3 C13P3

```
1
2 clear
3  clc
4 //to find speed of center of mass
5 //to find change in stored internal energy
6
7
8 // GIVEN:
9 //refer to figure 13-5 on page no. 285
10 //mass of ice skater
11 M = 50//in Kg
12 //force exerted
13 F = 55//in N
14 //distance moved by center of mass
15 scm = 32e-2//in m
16 // SOLUTION:
17 //consider newton's third law and center of mass
     equation
18 //speed of center of mass
19 vcm = sqrt(2*F*scm/M)//in m/s
```

```
20 //applying conservation of energy principle
21 //change in stored internal energy
22 delta_Eint = -(1/2)*(M*vcm^2)//in J
23
24 printf ("\n\n Speed of center of mass vcm  = \n\n %
   .2f m/s",vcm)
25 printf ("\n\n Change in stored internal energy
   delta_Eint = \n\n %.1f J",delta_Eint)
```

Scilab code Exa 13.4 C13P4

```
1
2 clear
3  clc
4 //to find speed of John after contact is broken
5 //to find change in stored internal energy of skater
6
7
8 // GIVEN:
9 //refer to figure 13-9(a),(b) on page no. 288
10 //mass of John skater
11 M = 50//in Kg
12 //mass of Jim skater
13 M1 = 72//in Kg
14 //force exerted by Jim
15 Fext = 55//in N
16 //distance through which force is applied
17 s = 32e-2//in m
18 //distabce moved by center of mass
19 scm = 58e-2//in m
20
21 // SOLUTION:
22 //consider John as our system
23 //applyinq consevation of energy principle
24 //applying center of mass equation
```

```
25 //change in kinetic energy
26 delta_Kcm = Fext*scm//in J
27 //speed of John after contact is broken
28 vcm = sqrt(2*delta_Kcm/M)//in m/s
29 //change in John's internal energy
30 delta_E_int_John = Fext*s-Fext*scm//in J
31 //change in Jim's internal energy
32 delta_E_int_Jim = -(Fext*s)//in J
33
34 printf ("\n\n Change in kinetic energy delta_Kcm =
     \n\n %.1f J",delta_Kcm)
35 printf ("\n\n Speed of John after contact is broken
     vcm = \n\n %.2f m/s",vcm)
36 printf ("\n\n Change in Johns internal energy
     delta_E_int_John = \n\n %.1f J",delta_E_int_John)
37 printf ("\n\n Change in Jim internal energy
     delta_E_int_Jim = \n\n %.1f J",delta_E_int_Jim)
```

Scilab code Exa 13.5 C13P5

```
1
2 clear
3  clc
4 //to find change in stored internal energy of system
      of block+surface
5 //distance travelled by block befire coming to rest
6
7 // GIVEN:
8 //mass of block
9 M = 5.2//in Kg
10 //initial horizontal velocity of block
11 vcm = 0.65//in m/s
12 //coefficient of kinetic friction
13 mew = 0.12
14 //acceleration due to gravity
```

```
15 g = 9.8//in m/s^2
16
17 // SOLUTION:
18 //applyinq consevation of energy principle
19 //change in stored internal energy of system of
      block+surface
20 //final kinetic energy is zero as block comes to
      rest
21 delta_Eint = -(0-(1/2*M*vcm^2))//in J //-ve sign as
      kinetic energy is lost
22 //distance travelled by block befire coming to rest
23 scm = (vcm^2/(2*mew*g))//in m
24
25 printf ("\n\n Final kinetic energy is zero as block
      comes to rest delta_Eint = \n\n %.1f J",
      delta_Eint)
26 printf ("\n\n Distance travelled by block befire
      coming to rest scm = \n\n %.2f m",scm)
```

Scilab code Exa 13.6 C13P6

```
1
2 clear
3  clc
4 //to find energy and direction of outgoing particl 3
     H
5
6 // GIVEN:
7 //refer to figure 13-11 from page no. 290
8 //difference in internal energy of initial and final
      partical
9 delta_Eint = 4.03//in MeV
10 //initial kinetic energy of deuteron
11 Ki = 1.50//in MeV
12 //initial kinetic energy of proton
```

```
13 K1 = 3.39//in MeV
14 //mass of hydrogen
15 m1 = 1.01//u
16 //mass of deuteron
17 m2 = 2.01//u
18 //mass of proton
19 m3 = 3.02//u
20
21 // SOLUTION:
22 //applyinq consevation of energy principle
23 //final kinetic energy
24 Kf = delta_Eint+Ki//in MeV
25 //final kinetic energy of outgoing partical 3H
26 K3 = Kf-K1//in MeV
27 //applying conservation of momentum principle
28 //value of cosfi
29 f = sqrt((m2*Ki)/(m3*K3))
30 //direction of outgoing particl 3H
31 fi = acosd(sqrt((m2*Ki)/(m3*K3)))//in degrees
32
33 printf ("\n\n Final kinetic energy Kf = \n\n %.2f
     MeV",Kf)
34 printf ("\n\n Final kinetic energy of outgoing
     partical 3H K3 = \n\n %.2f MeV",K3)
35 printf ("\n\n Value of cosfi = \n\n %.3f ",f)
36 printf ("\n\n Direction of outgoing particl 3H fi =
     \n\n %.1f degree",fi)
```

Scilab code Exa 13.7 C13P7

```
1
2 clear
3  clc
4 //to find kinetic energy of radon and alpha partical
5
```

```
6
7  // GIVEN:
8  //decrease in internal energy
9  delta_E = 4.87//in MeV
10 //mass of alpha partical
11 mHe = 4.00//in u
12 //mass of radon partical
13 mRn = 222.0//in u
14
15 // SOLUTION:
16 //applying conservation of energy principle
17 //we get two equations
18 //one for ratio of kinetic energies and second for
      total kinetic energy
19 //solving two equations using matrix
20 A = [1 (-mHe/mRn);1 1]
21 b = [0;4.87]
22 c = A\b
23 //ratio of kinetic energies
24 KRn_by_KHe = mHe/mRn
25 //total kinetic energy of products
26 Kf = delta_E//in MeV
27 //kinetic energy of radon partical
28 K_Rn = c(1)//in  MeV
29 //kinetic energy of alpha partical
30 K_He = c(2)//in  MeV
31
32 printf ("\n\n Ratio of kinetic energies KRn_by_KHe =
       \n\n %.4f",KRn_by_KHe)
33 printf ("\n\n Total kinetic energy of products Kf =
      \n\n %.2f MeV",Kf)
34 printf ("\n\n Kinetic energy of radon partical K_Rn
       = \n\n %.3f MeV",K_Rn)
35 printf ("\n\n Kinetic energy of alpha partical K_He
       = \n\n %.2f MeV",K_He)
```

Chapter 14

GRAVITATION

Scilab code Exa 14.1 C14P1

```
1 clear
2  clc
3 //to find magnitude of gravitational force exerted
     on cantaloupe on the surface of earth
4 //due to (a)the Earth (b)the Moon (c)the Sun
5
6 // GIVEN:
7 //mass of cantaloupe
8 mc = 1.00//in Kg
9 //acceleration due to gravity
10 g = 9.8//in m/s^2
11 //Gravitational constant
12 G = 6.67e-11//in N.m^2/Kg^2
13 //mass of moon
14 m_M = 7.36e22//in Kg
15 //mass of sun
16 m_S = 1.99e30//in Kg
17 //radius of moon
18 r_M = 3.82e8//in m
19 //radius of sun
20 r_S = 1.50e11//in m
```

```
21
22 // SOLUTION:
23 //applying newton's law of universal gravitation
24 //gravitational force exerted on cantaloupe on the
      surface of earth
25 //due to (a)the Earth
26 FcE = mc*g//in N
27 //gravitational force exerted on cantaloupe on the
      surface of earth
28 //due to (a)the Moon
29 FcM = G*((mc*m_M)/(r_M)^2)//in N
30 //gravitational force exerted on cantaloupe on the
      surface of earth
31 //due to (a)the Sun
32 FcS = G*((mc*m_S)/(r_S)^2)//in N
33
34 printf ("\n\n Gravitational force exerted on
      cantaloupe on the surface of earth\n due to (a)
      the Earth FcE = \n\n %.1f N",FcE)
35 printf ("\n\n Gravitational force exerted on
      cantaloupe on the surface of earth\n due to (b)
      the Moon FcM = \n\n %.2e N",FcM)
36 printf ("\n\n Gravitational force exerted on
      cantaloupe on the surface of earth\n due to (c)
      the Sun FcS = \n\n %.2e N",FcS)
```

Scilab code Exa 14.2 C14P2

```
1 clear
2  clc
3 //to find magnitude and direction of gravitational
      force
4
5 // GIVEN:
6 //refer to figure 14-4 on page no. 302
```

```
7  //mass of astronaut
8  ma = 105//in Kg
9  //mass of first asteroid
10 m1 = 346//in Kg
11 //radius of first asteroid
12 r1 = 215//in m
13 //mass of second asteroid
14 m2 = 184//in Kg
15 //radius of second asteroid
16 r2 = 142//in m
17 //angle between forces
18 theta = 120//in degrees
19 //Gravitational constant
20 G = 6.67e-11//in N.m^2/Kg^2
21
22 // SOLUTION:
23 //applying newton's law of universal gravitation
24 //magnitude of gravitational force due to first
      asteroid
25 Fa1 = G*((ma*m1)/(r1^2))//in N
26 //magnitude of gravitational force due to second
      asteroid
27 Fa2 = G*((ma*m2)/(r2^2))//in N
28 //magnitude of total gravitational force
29 //using parallelogram method
30 Fa = sqrt((Fa1^2)+(Fa2^2)+(2*Fa1*Fa2*cosd(theta)))
31 //direction of gravitational force
32 fi = atand((Fa2*sind(theta))/(Fa1+(Fa2*cosd(theta)))
      )//in degrees
33 Fa = nearfloat("pred",5.80e-11)
34
35 printf ("\n\n Magnitude of gravitational force due
      to first asteroid Fa1 = \n\n %.2e N",Fa1)
36 printf ("\n\n Magnitude of gravitational force due
      to second asteroid Fa2 = \n\n %.2e N",Fa2)
37 printf ("\n\n Magnitude of total gravitational force
       Fa = \n\n %.2e N",Fa)
38 printf ("\n\n Direction of gravitational force fi =
```

```
\n\n %.1f degree",fi)
```

Scilab code Exa 14.3 C14P3

```
clear
 clc
//to find free fall acceleration of neutron ster and
    asteroid ceres

// GIVEN:
//mass of neutron star
Mn = 1.99e30//in Kg
//radius of neutron star
Rn = 12e3//in m
//mass of asteroid ceres
Mc = 1.2e21//in Kg
//radius of asteroid ceres
Rc = 4.7e5//in m
//Gravitational constant
G = 6.67e-11//in N.m^2/Kg^2

// SOLUTION:
//applying newton's law of universal gravitation and
    newton's second law of motion
//free fall acceleration of neutron sterid
g0 = G*(Mn/(Rn^2))//in m/s^2
//free fall acceleration of austeroid ceres
go = G*(Mc/(Rc^2))//in m/s^2

printf ("\n\n Free fall acceleration of neutron
   sterid g0 = \n\n %.1e m/s^2",g0)
printf ("\n\n Free fall acceleration of austeroid
   ceres go = \n\n %.2f m/s^2",go)
```

Scilab code Exa 14.4 C14P4

```
clear
 clc
//to find speed of partical at r = 0

// GIVEN:

//mass of Earth
ME = 5.98e24//in Kg
//radius of Earth
RE = 6.37e6//in m
//Gravitational constant
G = 6.67e-11//in N.m^2/Kg^2

// SOLUTION:
//applying newton's law of universal gravitation and law of conservation of energy
//speed of partical at r = 0
v = sqrt((G*ME)/(RE))//in m/s
printf ("\n\n Speed of partical at r = 0 is v = \n\n %.2e m/s",v)
```

Scilab code Exa 14.5 C14P5

```
clear
 clc
//to find speed of canister when it enters the Earth's atmosphere

// GIVEN:

```

```
7 //mass of Earth
8 ME = 5.98e24//in Kg
9 //radius of Earth
10 RE = 6.37e6//in m
11 //initial speed of canister
12 vi = 525//in m/s
13 //distance above earth's surface
14 h = 100e3//in m
15 //Gravitational constant
16 G = 6.67e-11//in N.m^2/Kg^2
17
18 // SOLUTION:
19 //applying newton's law of universal gravitation and
       law of conservation of energy
20 //speed of canister when it enters the Earth's
     atmosphere
21 vf_square = vi - ((2*G*ME)*((1/(3*RE))-(1/(RE+h))))
     //in m^2/s^2
22 vf = sqrt(vi - ((2*G*ME)*((1/(3*RE))-(1/(RE+h)))))//
     in m/s
23 vf = nearfloat("succ",9.05e3)
24 vf_square = nearfloat("succ",8.18e7)
25
26 printf ("\n\n Square of speed of canister when it
     enters the Earths atmosphere vf_square = \n\n %.2
     e m^2/s^2",vf_square)
27 printf ("\n\n Speed of canister when it enters the
     Earths atmosphere vf = \n\n %.2e m/s",vf)
```

Scilab code Exa 14.7 C14P7

```
1 clear
2 clc
3 //to find mass of Sun and mass of Jupiter
4
```

```
5 // GIVEN:
6
7 //orbital radius of earth
8 re = 1.50e11//in m
9 //period of revolution for earth
10 Te = 3.15e7//in seconds
11 //orbital radius of Moon
12 rm = 4.22e8//in m
13 //period of revolution for Moon
14 Tm = 1.53e5//in seconds
15 //Gravitational constant
16 G = 6.67e-11//in N.m^2/Kg^2
17
18 // SOLUTION:
19 //applying Kepler's law of peroids
20 //mass of Sun using Earth's orbital motion
21 M = (4*(%pi^2)*(re^3))/(G*(Te^2))//in Kg
22 //mass of Jupiter using Moon's orbital motion
23 M_ = (4*(%pi^2)*(rm^3))/(G*(Tm^2))//in Kg
24
25 printf ("\n\n Mass of Sun using Earth orbital motion
     M = \n\n %.2e Kg",M)
26 printf ("\n\n Mass of Jupiter using Moon orbital
    motion M = \n\n %.2e Kg",M_)
```

Scilab code Exa 14.8 C14P8

```
1 clear
2 clc
3 //to find height above the Earth
4
5 // GIVEN:
6
7 //period of the satellite
8 T = 86400//in seconds
```

```
9  //mass of Earth
10 ME = 5.98e24//in Kg
11 //radius of Earth
12 RE = 6.37e6//in meters
13 //Gravitational constant
14 G = 6.67e-11//in N.m^2/Kg^2
15
16 // SOLUTION:
17 //applying Kepler's law of peroids
18 //radius of orbit of satellite
19 r = ((G*T^2*ME)/(4*%pi^2))^(1/3)//in meters
20 //height above the Earth
21 h = r-RE//in meters
22 r = nearfloat("pred",4.22e7)
23 h = nearfloat("pred",3.58e7)
24
25 printf ("\n\n Radius of orbit of satellite r = \n\n
     %.2e m",r)
26 printf ("\n\n Height above the Earth h = \n\n %.2e m
     ",h)
```

Scilab code Exa 14.9 C14P9

```
1  clear
2  clc
3  //to find aphelion or farthest distance of Halley's
     comet from the Sun and eccentricity of it's orbit
4
5  // GIVEN:
6  //refer to figure 14-16 from page no. 313
7  //period of the Halley's comet
8  T = 76 //in years
9  //mass of Sun
10 M = 2.0e30//in Kg
11 //closest approach to the Sun
```

```
12 //minimum distance of Halley's comet from Sun
13 Rp = 8.8e10//in meters
14 //Gravitational constant
15 G = 6.67e-11//in N.m^2/Kg^2
16
17 // SOLUTION:
18 //applying Kepler's law of peroids
19 //semimajor axis
20 a = ((G*(T*365*24*60*60)^2*M)/(4*%pi^2))^(1/3)//in
      meters //taking T in seconds
21 //refer to figure 14-14
22 //maximum distance of Halley's comet from Sun
23 Ra = (2*a)-Rp//in meters
24 //eccentricity of Halley's orbit
25 e = 1-(Rp/a)
26
27 printf ("\n\n Semimajor axis a = \n\n %.1e m",a)
28 printf ("\n\n Maximum distance of Halley comet from
      Sun Ra = \n\n %.1e m",Ra)
29 printf ("\n\n Eccentricity of Halley orbit e = \n\n
      %.2f ",e)
```

Scilab code Exa 14.10 C14P10

```
1 clear
2 clc
3 //to find energy,period,semimajor axis of B before
      and after burn
4
5 // GIVEN:
6 //refer to figure 14-19 from page no. 315
7 //mass of spacecraft
8 m = 3250//in Kg
9 //height above Earth
10 h = 270//in Km
```

```
11 //radius of earth
12 RE = 6370//in Km
13 //mass of earth
14 ME = 5.98e24//in Kg
15 //decrease in velocity after burn
16 d = 0.95//in percent
17 //Gravitational constant
18 G = 6.67e-11//in N.m^2/Kg^2
19
20 // SOLUTION:
21 //before burn
22 //semimajor axis before burn
23 a = RE+h//in Km
24 //energy before burn
25 E = -(G*m*ME)/(2*a*(1000))//in J
26 //period before burn
27 //applying Krpler's law of peroids
28 T = ((4*(%pi^2)*((a*1000)^3))/(G*ME))^(1/2)//in
      seconds
29 //kinetic energy before burn
30 K = -(E)//in J
31 //velocity before burn
32 v = sqrt((2*K)/m)//in m/s
33
34 //after burn
35 //velocity after burn
36 v_dash = (1-(d*0.01))*v//in m/s
37 //kinetic energy after burn
38 K_dash = 1/2*(m)*(v_dash)^2//in J
39 //potential energy after burn
40 U_dash = -(K)//in J
41 //total energy after burn
42 E_dash = K_dash+(2*U_dash)//in J
43 //semimajor axis after burn
44 a_dash = -((G*m*ME)/(2*E_dash))//in meters
45 //period after burn
46 T_dash = ((4*(%pi^2)*((a_dash)^3))/(G*ME))^(1/2)//in
       seconds
```

```
47 T = nearfloat("pred",5381)
48 E_dash = nearfloat("succ",-9.94e10)
49 T_dash = nearfloat("succ",5240)
50
51 printf ("\n\n Semimajor axis before burn a = \n\n
     %4i Km",a)
52 printf ("\n\n Energy before burn E = \n\n %.2e J",E)
53 printf ("\n\n Period before burn T = \n\n %4i s",T)
54 printf ("\n\n Kinetic energy before burn K = \n\n %
     .2e J",K)
55 printf ("\n\n Velocity before burn v = \n\n %.2e m/s
     ",v)
56 printf ("\n\n Velocity after burn v_dash = \n\n %.2e
      m/s",v_dash)
57 printf ("\n\n Kinetic energy after burn K_dash = \n\
     n %.2e J",K_dash)
58 printf ("\n\n Total energy after burn E_dash = \n\n
     %.2e J",E_dash)
59 printf ("\n\n Semimajor axis after burn a_dash = \n\
     n %.2e m",a_dash)
60 printf ("\n\n Period after burn T_dash = \n\n %4i s"
     ,T_dash)
```

Chapter 15

FLUID STATICS

Scilab code Exa 15.1 C15P1

```
clear
clc
//to find density of oil

// GIVEN:
//refer to figure 15-6 from page no. 336
//height of water level above oil on one side
d = 12.3//in mm
//height of water level above oil on second side
a = 67.5//in mm
//density of water
rho_w = 1.000e3//in Kg/m^3

// SOLUTION:
//equating pressure on both sides
//density of oil
rho = rho_w*((2*a)/((2*(a)+d)))//in Kg/m^3

printf ("\n\n Density of oil rho = \n\n %3i Kg/m^3",
   rho)
```

Scilab code Exa 15.2 C15P2

```
clear
clc
//to find applied force
//to find distance by which car is raised

// GIVEN:
//refer to figure 15-9 from page no. 338
//diameter of smaller piston
Di = 2.2//in cm
//combined mass
M = 1980//in Kg
//diameter of larger piston
D0 = 16.4//in cm
//length of pump handle
L = 36//in cm
//distance of pivot to the piston
x = 9.4//in cm
//acceleration due to gravity
g = 9.8//in m/s^2
//vertical distance by which hand moves
h = 28//in cm

// SOLUTION:
//area of larger piston
A0 = %pi*(D0/2)^2//in cm^2
//area of smaller piston
Ai = %pi*(Di/2)^2//in cm^2
//applied force to the smaller piston
Fi = M*g*(Ai/A0)//in N
//using Newton's third law of motion
//applied force at the end of pump handle
Fh = Fi*(x/L)//in N
```

```
33 //distance moved by smaller piston
34 di = h*(x/L)//in cm
35 //equating pressure on each side
36 //distance moved by larger piston and car is raised
   by
37 d0 = di*(Ai/A0)//in cm
38
39 printf ("\n\n Applied force to the smaller piston Fi
    = \n\n %3i N",Fi)
40 printf ("\n\n Applied force at the end of pump
   handle Fh = \n\n %2i N",Fh)
41 printf ("\n\n Distance moved by smaller piston di =
   \n\n %.1f cm",di)
42 printf ("\n\n Distance moved by larger piston and
   car is raised by d0 = \n\n %.2f cm",d0)
```

Scilab code Exa 15.3 C15P3

```
1 clear
2 clc
3 //to find fraction of total volume of iceberg is
   exposed
4
5 // GIVEN:
6 //density of water
7 rho_w = 1024//in Kg/m^3
8 //density of ice
9 rho_i = 917//in Kg/m^3
10
11 // SOLUTION:
12 //applying Archimedes' principle
13 //ratio of volume of water displaced to volume of
   submerged portion of ice
14 Vw_by_Vi = (rho_i/rho_w)*100//in percent
15 //percent of iceberg exposed
```

```
16 V = 100-(Vw_by_Vi)//in percent
17
18 printf ("\n\n Ratio of volume of water displaced to
   volume of submerged portion of ice Vw_by_Vi = \n\
   n %.1f percent",Vw_by_Vi)
19 printf ("\n\n Percent of iceberg exposed  = \n\n %.1
   f percent",V)
```

Scilab code Exa 15.4 C15P4

```
1 clear
2 clc
3 //to find atmospheric pressure
4
5 // GIVEN:
6 //height of mercury column in barometer
7 h = 740.35//in mm
8 //temperature
9 T = -5.0//in degree
10 //density of mercure
11 rho = 1.3608e4//in Kg/m^3
12 //acceleration due to gravity
13 g = 9.7835//in m/s^2
14
15 // SOLUTION:
16 //atmospheric pressure
17 po = rho*g*h*10^-3//in Pa
18 //taking h in meters
19
20 printf ("\n\n Atmospheric pressure po  = \n\n %.4e
   Pa",po)
```

Scilab code Exa 15.5 C15P5

```
clear
clc
//to find surface tension of liquid

// GIVEN:
//refer to figure 15-15(a) on page no. 343
//upward force
p = 3.45e-3//in N
//length of wire
d = 4.85//in cm
//linear mass density
mew = 1.75e-3//in Kg/m
//acceleration due to gravity
g = 9.7835//in m/s^2

// SOLUTION:
//refer to figure 15-15(a) on page no. 343
//using equilibrium condition
//surface tension of liquid
Gamma = (p-(mew*(d*10^-2)*g))/(2*d*(10^-2))//in N/m
//taking d in meters

printf ("\n\n Surface tension of liquid Gamma = \n\
   n %.3f N/m",Gamma)
```

Chapter 16

FLUID DYNAMICS

Scilab code Exa 16.1 C16P1

```
clear
clc
//to find volume flow rate of water

// GIVEN:
//refer to figure 16-5 on page no. 354
//cross sectional area
A1 = 1.2//in cm^2
//cross sectional area
A2 = 0.35//in cm^2
//vertical distance between two levels
h = 45//in mm
//acceleration due to gravity
g = 9.8//in m/s^2

// SOLUTION:
//applying equation of continuity and conservation
   of energy between two levels
//speed of water at level 1
v1 = sqrt((2*g*(h*10^-3)*(A2^2))/(A1^2-A2^2))//in m/
   s //taking h in meters
```

```
20 //volume flow rate
21 V1 = v1*100//in speed v1 in cm/s
22 R = A1*V1//in cm^2/s
23
24 printf ("\n\n Speed of water at level 1 v1 = \n\n %
      .3f m/s",v1)
25 printf ("\n\n Volume flow rate R = \n\n %2i cm^3/s",
      R)
```

Scilab code Exa 16.2 C16P2

```
1 clear
2 clc
3 //to find pressure in horizontal pipe and flow speed
      in pressure in smaller pipe
4
5 // GIVEN:
6 //refer to figure 16-7 on page no. 356
7 //height of storage tower
8 h = 32//in m
9 //diameter of storage tower
10 D = 3.0//in m
11 //diameter of horizontal pipe
12 d = 2.54//in m
13 //delivery rate of water
14 R = 0.0025//in m^3/s
15 //diameter of smaller pipe
16 d_dash = 1.27//in cm
17 //distance above the ground for water supply
18 yC = 7.2//in m
19 //initial pressure
20 p0 = 1.01e5//in Pa
21 //density of water
22 rho = 1.0e3//in Kg/m^3
23 //acceleration due to gravity
```

```
24 g = 9.8//in m/s^2
25
26 // SOLUTION:
27 //area at leval A
28 A_A = %pi*(1.5)^2//in m^2
29 //area at leval B
30 A_B = %pi*(0.0127)^2//in m^2
31 //area at leval C
32 A_C = %pi*((d_dash*10^-2)/2)^2//in m^2
33 //applying equation of continuity
34 //speed of water at point A
35 vA = R/A_A//in m/s
36 //speed of water at point B
37 vB = R/A_B//in m/s
38 //applying Bernoulli's equation
39 //pressure in pipe at B
40 pB = p0+(rho*g*h)-((1/2)*rho*(vB^2))//in Pa
41 //applying equation of continuity
42 //speed of water at point C
43 vC = R/A_C//in m/s
44 //take h = yA
45 //applying Bernoulli's equation
46 //pressure in pipe at C
47 pC = p0-((1/2)*rho*(vC^2))+(rho*g*(h-yC))//in Pa
48 pB = nearfloat("succ",4.03e5)
49
50
51 printf ("\n\n Speed of water at point A vA = \n\n %
     .1e m/s",vA)
52 printf ("\n\n Speed of water at point B vB = \n\n %
     .1f m/s",vB)
53 printf ("\n\n Pressure in pipe at B pB = \n\n %.2e
     Pa",pB)
54 printf ("\n\n Speed of water at point C vC = \n\n %
     .1f m/s",vC)
55 printf ("\n\n Pressure in pipe at C pC = \n\n %.2e
     Pa",pC)
```

Scilab code Exa 16.3 C16P3

```
clear
clc
//to find coefficient of viscocity of castor oil

// GIVEN:
//density of castor oil
rho = 0.96e3//in Kg/m^3
//gauge pressure of pump
delta_p = 950//in Pa
//diameter of pipe
D =  2.6//in cm
//length of pipe
L = 65//in cm
//time interval in which oil is collected
dt = 90//in seconds
//mass of oil collected in dt time interval
dm = 1.23//in Kg
// SOLUTION:
//radius of pipe
R = (D*10^-2)/2//in meters
//mass flux
dm_by_dt = (dm/dt)//in Kg/s
//coefficient of viscocity of castor oil
eta = (rho*%pi*(R^4)*delta_p)/(8*(dm/dt)*(L*10^-2))
   //in N.s/m^2 //taking Lin meters

printf ("\n\n Mass flux dm_by_dt = \n\n %.4f Kg/s",
   dm_by_dt)
printf ("\n\n Coefficient of viscocity of castor oil
    eta = \n\n %.2f N.s/m^2",eta)
```

Chapter 17

OSILLATIONS

Scilab code Exa 17.1 C17P1

```
clear
clc
//to find force constant k of spring
//to find magnitude of horizontal force and period
   of oscillation

// GIVEN:
//refer to figure 17-5 from page no. 375
//mass of boby
M = 1.65//in Kg
//increase in length
y = 7.33//in cm
//mass of block
m = 2.43//in Kg
//distance by which spring is streched
x = 11.6//in cm
//acceleration due to gravity
g = 9.81//in m/s^2

// SOLUTION:
//applying simple harmonic motion equation
```

```
21 //equating forces in y direction
22 //force constant k of spring
23 k = (-M*g)/(-y*10^-2)//in N/m //taking y in meters
24 //magnitude of horizontal force
25 F = k*(x*10^-2)//in N //taking x in meters
26 //period of oscillation
27 T = (2*%pi*(sqrt(m/k)))*10^3//in miliseconds
28 k = round(k)
29
30 printf ("\n\n Force constant k of spring k = \n\n
     %3i N/m",k)
31 printf ("\n\n Magnitude of horizontal force F = \n\n
     %.1f N",F)
32 printf ("\n\n Period of oscillation T = \n\n %3i ms"
     ,T)
```

Scilab code Exa 17.2 C17P2

```
1 clear
2 clc
3 //to find total energy stored in the system
4 //to find maximum speed and magnitude of maximum
     acceleration of block
5 //to find position,velocity and acceleration of
     block at t = 0.215s
6
7 // GIVEN:
8 //refer to problem 17-1
9 //mass of boby
10 M = 1.65//in Kg
11 //increase in length
12 y = 7.33//in cm
13 //mass of block
14 m = 2.43//in Kg
15 //distance by which spring is streched
```

```
16 x_m = 11.6//in cm
17 //time
18 t = 0.215//seconds
19 //acceleration due to gravity
20 g = 9.81//in m/s^2
21
22 // SOLUTION:
23 //applying simple harmonic motion equation
24 //equating forces in y direction
25 //force constant k of spring
26 k = (-M*g)/(-y*10^-2)//in N/m //taking y in meters
27 //total energy stored in the system
28 E = (1/2)*k*((x_m*10^-2)^2)//in J
29 //magnitude of kinetic energy
30 K_max = E//in J
31 //maximum speed of block
32 v_max = sqrt((2*K_max)/m)//in m/s
33 //maximum acceleration of block
34 a_max = (k*(x_m*10^-2))/m//in m/s^2
35 //period of oscillation
36 T = (2*%pi*(sqrt(m/k)))*10^3//in miliseconds
37 //angular frequency
38 omega = (2*%pi)/(T*10^-3)//in rad/s
39 z = omega*t
40 //position of block at t = 0.215s
41 x = (x_m*10^-2)*(cos(z))//in m
42 //velocity of block at t = 0.215s
43 vx = -(omega*(x_m*10^-2))*(sin(z))//in m/s
44 //acceleration of block at t = 0.215s
45 ax = -(omega^2)*x//in m/s^2
46 omega = nearfloat("succ",9.536)
47 a_max = nearfloat("succ",10.6)
48 x = nearfloat("succ",-0.0535)
49 ax = nearfloat("succ",4.87)
50
51 printf ("\n\n Total energy stored in the system E =
     \n\n %.2f J",E)
52 printf ("\n\n Maximum speed of block v_max = \n\n %
```

```
    .2f m/s",v_max)
53 printf ("\n\n Maximum acceleration of block a_max =
    \n\n %.1f m/s^2",a_max)
54 printf ("\n\n Angular frequency omega = \n\n %.3f
    rad/s",omega)
55 printf ("\n\n Position of block at t = 0.215s x = \n
    \n %.4f m",x)
56 printf ("\n\n Velocity of block at t = 0.215s vx = \
    n\n %.3f m/s",vx)
57 printf ("\n\n acceleration of block at t = 0.215s ax
    = \n\n %.2f m/s^2",ax)
```

Scilab code Exa 17.3 C17P3

```
1  clear
2  clc
3  //to find equation for x(t)
4
5  // GIVEN:
6  //refer to problem 17-1
7  //mass of boby
8  M = 1.65//in Kg
9  //increase in length
10 y = 7.33//in cm
11 //mass of block
12 m = 2.43//in Kg
13 //distance by which spring is streched
14 x_m = 11.6//in cm
15 //displacement of block
16 x = 0.0624//in meters
17 //velocity of block
18 vx = 0.847//in m/s
19 //acceleration due to gravity
20 g = 9.81//in m/s^2
21
```

```
22 // SOLUTION:
23 //applying simple harmonic motion equation
24 //equating forces in y direction
25 //force constant k of spring
26 k = (-M*g)/(-y*10^-2)//in N/m //taking y in meters
27 //total energy of system
28 E = ((1/2)*m*(vx^2))+((1/2)*k*(x^2))//in J
29 //maximum amplitude of motion
30 xm = sqrt((2*E)/k)//in meters
31 //using cosin equation of x
32 //value of cos(fi)
33 cos_fi = x/xm
34 //phase constant
35 fi1 = acosd(cos_fi)
36 fi2 = 360-(fi1)
37 fi = fi2*(%pi/180)//in rad
38 //period of oscillation
39 T = (2*%pi*(sqrt(m/k)))*10^3//in miliseconds
40 //angular frequency
41 omega = (2*%pi)/(T*10^-3)//in rad/s
42 //initial velocity
43 v_x1 = -(omega*xm)*sind(fi1)//in m/s
44 v_x2 = -(omega*xm)*sind(fi2)//in m/s
45 xm = nearfloat("pred",0.1085)
46 cos_fi = nearfloat("succ",0.5751)
47 omega = nearfloat("succ",9.54)
48 fi = nearfloat("succ",5.33)
49
50 printf ("\n\n Total energy of system E = \n\n %.3f J
     ",E)
51 printf ("\n\n Maximum amplitude of motion xm = \n\n
     %.4f m",xm)
52 printf ("\n\n Value of cos(fi) = \n\n %.4f",cos_fi)
53 printf ("\n\n Initial velocity = \n\n %.3f for fi =
     %.1f degree \n 0r \n %.3f for fi = %.1f degree",
     v_x1,fi1,v_x2,fi2)
54 printf ("\n\n Equation for x(t) = \n\n (%.3f m)*(cos
     (%.2f rad/s)t + %.2f rad)",xm,omega,fi)
```

Scilab code Exa 17.4 C17P4

```
clear
clc
//to find rotaional inertia of traingle

// GIVEN:
//mass of rod
M = 0.112//in Kg
//length of rod
L = 0.096//in m
//period of oscillations of rod
T_rod = 2.14//in seconds
//period of oscillations of traingular shape body
T_triangle = 5.83//in seconds


// SOLUTION:
//using equation of physical pendulum
//rotational inertia of body
I_rod = (M*L^2)/12//in Kg.m^2
//rotaional inertia of traingle
I_triangle = I_rod*(T_triangle/T_rod)^2//in Kg.m^2

printf ("\n\n Rotational inertia of body I_rod = \n\
    n %.2e Kg.m^2",I_rod)
printf ("\n\n Rotaional inertia of traingle
    I_triangle = \n\n %.2e Kg.m^2",I_triangle)
```

Scilab code Exa 17.6 C17P6

```
clear
clc
//to find value of acceleration due to gravity

// GIVEN:
//radius of disk
R = 10.2//in cm
//period
T = 0.784//in seconds

// SOLUTION
//refer to problem 17-5
//acceleration due to gravity
g = (6*(%pi^2)*(R*10^-2))/(T^2)//in m/s^2

printf ("\n\n Value of acceleration due to gravity g
    = \n\n %.2f m/s^2",g)
```

Scilab code Exa 17.7 C17P7

```
clear
clc
//to find time required by body to come halfway

// GIVEN:
//refer to figure 17-15 from page no. 385
//from the equation given
//radius of reference circle
r = 0.35//in m
//angular speed
omega = 8.3//in rad/s

// SOLUTION
//refer to problem 17-5
//angle turned to come halfway
```

```
16 wt = 60//in degree
17 //time required by body to come halfway
18 t = ((wt*%pi)/180)/omega//in seconds //taking angle
      in radians
19
20 printf ("\n\n Angle turned to come halfway wt = \n\n
       %2i degree",wt)
21 printf ("\n\n Time required by body to come halfway
      t = \n\n %.2f seconds",t)
```

Scilab code Exa 17.8 C17P8

```
1 clear
2 clc
3 //to find periods of oscillations
4
5 // GIVEN:
6 //refer to figure 17-11 from page no. 386
7 //mass of block
8 m = 250//in gram
9 //force constant
10 k = 85//in N/m
11 //damping constant
12 b = 0.070//in Kg/s
13
14 // SOLUTION
15 //using equation of damped oscillatory motion
16 //for small damping period
17 T = 2*%pi*(sqrt((m*10^-3)/k))//in seconds //taking m
       in Kg
18 //periods of oscillations
19 t = ((m*10^-3)*(log(2)))/b//in seconds //taking m in
       Kg
20
21 printf ("\n\n For small damping period T = \n\n %.2f
```

```
    seconds",T)
22 printf ("\n\n Periods of oscillations t = \n\n %.1f
    seconds",t)
```

Scilab code Exa 17.9 C17P9

```
1  clear
2  clc
3  //to find reduced mass of molecule
4  //to find effective force constant
5
6  // GIVEN:
7  //refer to figure 17-22 from page no. 390
8  //mass of hydrogen atom
9  m1 = 1.007825//in u
10 //mass of isotop cl-35
11 m2 = 34.968853//in u
12 //mass of isotop cl-37
13 m3 = 36.965903//in u
14 //vibrational frequency
15 f = 8.5e13//in Hz
16 //mass
17 M = 1.66e-27//in Kg/u
18
19 // SOLUTION
20 //reduced mass of H35cl
21 m = (m1*m2)/(m1+m2)//in u
22 //reduced mass of H37cl
23 m_1 = (m1*m3)/(m1+m3)//in u
24 //effective force constant
25 k = 4*(%pi^2)*(f^2)*m_1*M//in N/m
26
27 printf ("\n\n Reduced mass of H35cl m = \n\n %.6f u"
    ,m)
28 printf ("\n\n Reduced mass of H37cl m_1 = \n\n %.6f
```

```
    u",m_1)
29 //answer is slightly different than ans. in book.But
       ans. by scilab program is same as that of
    calculator.
30 printf ("\n\n Effective force constant k = \n\n %3i
    N/m",k)
```

Chapter 18

WAVE MOTION

Scilab code Exa 18.1 C18P1

```
1 clear
2 clc
3 //to find amplitude,frequency,speed and wave length
     of the wave motion
4 //to find equation of wave
5
6 // GIVEN:
7 //distance moved up and down
8 x = 1.30//in cm
9 //frequency
10 f = 125//in per second
11 //wavelength
12 lambda = 15.6//in cm
13
14 // SOLUTION
15 //using equations of sinusoidal wave motion
16 //amplitude of wave motion
17 ym = x/2//in cm
18 //wave speed
19 v = (lambda*10^-2)*f//in m/s //taking lambda in
     meters
```

```
20 //wave number
21 k = (2*%pi)/(lambda*10^-2)//in rad/m //taking lambda
       in meters
22 //angular frequency
23 omega = v*k//in rad/s
24 omega = nearfloat("succ",786)
25
26 printf ("\n\n Amplitude of wave motion ym = \n\n %.2
       f cm",ym)
27 printf ("\n\n Wave speed v = \n\n %.1f m/s",v)
28 printf ("\n\n Wave number k = \n\n %.1f rad/m",k)
29 printf ("\n\n Angular frequency omega = \n\n %3i rad
       /s",omega)
30 printf ("\n\n Equation of wave is \n\n y(x,t) = (%.2
       f cm)*sin[(%.1f rad/m)x - (%3i rad/s)t] ",ym,k,
       omega)
```

Scilab code Exa 18.2 C18P2

```
1 clear
2 clc
3 //to find expression of velocity and acceleration of
      partical p
4 //to find displacement,velocity and accleration of
      partical
5
6 // GIVEN:
7 //refer to problem 18-1
8 //distance moved up and down
9 x = 1.30//in cm
10 //frequency
11 f = 125//in per second
12 //wavelength
13 lambda = 15.6//in cm
14 //location of partical p
```

```
15 xp = 0.245//in meters
16 //time
17 t = 15.0//in ms
18
19 // SOLUTION
20 //using equations of sinusoidal wave motion
21 //amplitude of wave motion
22 ym = x/2//in cm
23 //wave speed
24 v = (lambda*10^-2)*f//in m/s //taking lambda in
      meters
25 //wave number
26 k = (2*%pi)/(lambda*10^-2)//in rad/m //taking lambda
       in meters
27 //angular frequency
28 omega = v*k//in rad/s
29 omega = nearfloat("succ",786)
30 //value of constant
31 ym_into_omega = ym*omega//in cm/s
32 k_into_x = k*xp//in rad
33 omega2_into_ym = (omega^2)*ym//in cm/s^2
34 //displacement of partical at t
35 y = (ym)*(sin((k_into_x) - (omega*(t*10^-3))))//in
      cm/s
36 //velocity of partical at t
37 uy = -(ym_into_omega)*(cos((k_into_x) - (omega*(t
      *10^-3))))//in cm/s
38 //acceleration of partical at t
39 ay = -(omega2_into_ym)*(sin((k_into_x) - (omega*(t
      *10^-3))))//in cm/s^2
40 ym_into_omega = round(ym_into_omega)
41
42 printf ("\n\n Expression of velocity of partical p
      is \n\n uy(xp,t) = -(%3i cm/s)*cos[(%.2f rad) - (
      %3i rad/s)t] ",ym_into_omega,k_into_x,omega)
43 printf ("\n\n Expression of accleration of partical
      p is \n\n ay(xp,t) = -(%.2e cm/s^2)*sin[(%.2f rad
      ) - (%3i rad/s)t] ",omega2_into_ym,k_into_x,omega
```

```
    )
44 printf ("\n\n Value of omega^2*ym = \n\n %.2e cm/s^2
    ",omega2_into_ym)
45 printf ("\n\n Displacement of partical at t y = \n\n
    %.2f cm",y)
46 //answer of uy is slightly different than book.
    answer of scilab program is same as that of
    calculator.
47 printf ("\n\n Velocity of partical at t uy = \n\n %
    .2f cm/s",uy)
48 printf ("\n\n Acceleration of partical at t ay = \n\
    n %.1e cm/s^2",ay)
```

Scilab code Exa 18.3 C18P3

```
1  clear
2  clc
3  //to find amplitude of combined wave
4  //to find value by which phase difference be changed
5
6  // GIVEN:
7  //amplitude of each wave
8  ym = 9.7//in mm
9  //phase difference
10 fi = 110//in degree
11
12 // SOLUTION
13 //using equations of interference of waves
14 //amplitude of combined wave
15 y = 2*ym*(cosd(fi/2))//in mm
16 //value by which phase difference be changed
17 delta_fi = 2*(acosd(1/2))//in degree
18 delta_fi1 = -(delta_fi)//in degree
19
20 printf ("\n\n Amplitude of combined wave y = \n\n %
```

```
    .1f mm",y)
21 printf ("\n\n Value by which phase difference be
    changed delta_fi = \n\n %3i degree or %3i degree
    ",delta_fi,delta_fi1)
```

Scilab code Exa 18.4 C18P4

```
1  clear
2  clc
3  //to find tension in string to get 4 loops
4
5  // GIVEN:
6  //refer to figure 18-23 from page no. 418
7  //freqyuency
8  fn = 120//in Hz
9  //length of string
10 L = 1.2//in meters
11 //linear mass density of string
12 mew = 1.6//in g/m
13 //no. of loops
14 n = 4
15
16 // SOLUTION
17 //using equation of wave motion
18 //tension in string to get 4 loops
19 F = (4*(L^2)*(fn^2)*(mew*10^-3))/(n^2)//in N //
    taking mew in Kg/m
20
21 printf ("\n\n Tension in string to get 4 loops F = \
    n\n %.1f N",F)
```

Scilab code Exa 18.5 C18P5

```
clear
clc
//to find longest wavelengths of resonance of the
   string
//to fing corresponding wavelengths that reach the
   ear of the listener

// GIVEN:
//freqyuency
f = 440//in Hz
//length of string
L = 0.34//in meters
//wave speed in air
v_air = 343//in m/s

// SOLUTION
//using equation of wave for resonance condition
//longest wavelengths of resonance of the string
lambda1 = (2*L)/1//in meters
lambda2 = (2*L)/2//in meters
lambda3 = (2*L)/3//in meters
//wave speed
v_string = f*lambda1//in m/s
//multiplication factor
v_air_by_v_string = (v_air/v_string)
//corresponding wavelengths that reach the ear of
   the listener
lambda_1 = (lambda1)*(v_air/v_string)//in meters
lambda_2 = (lambda2)*(v_air/v_string)//in meters
lambda_3 = (lambda3)*(v_air/v_string)//in meters

printf ("\n\n Longest wavelengths of resonance of
   the string \n lamda1 = %.2f m \n lamda2 = %.2f m
   \n lamda3 = %.2f m ",lambda1,lambda2,lambda3)
printf ("\n\n Wave speed v_string = \n\n %3i m/s ",
   v_string)
printf ("\n\n Relation between lambda_air and
   lambda_string is \n\n lambda_air = %.2f(
```

```
    lambda_string) ",v_air_by_v_string)
32 printf ("\n\n Corresponding wavelengths that reach
    the ear of the listener \n lamda_1 = %.2f m \n
    lamda_2 = %.2f m \n lamda_3 = %.2f m ",lambda_1,
    lambda_2,lambda_3)
```

Chapter 19

SOUND WAVES

Scilab code Exa 19.1 C19P1

```
1 clear
2 clc
3 //to find density and displacement amplitude
4
5 // GIVEN:
6 //maximum pressure variation
7 delta_pm = 28//in Pa
8 //frequency
9 f = 1000//in Hz
10 //pressure amplitude
11 delta_p1 = 2.8e-5//in Pa
12 //bulk modulus of air
13 B = 1.4e5//in Pa
14 //speed of sound in air
15 v = 343//in m/s
16 //density of air
17 rho_0 = 1.21//in Kg/m^3
18
19 // SOLUTION
20 //using equation of sound wave
21 //wave number
```

```
22 k = (2*%pi*f)/v//in rad/m
23 //density amplitude
24 delta_rho_m = delta_pm*(rho_0/B)//in Kg/m^3
25 //displacement amplitude
26 s_m = delta_pm/(k*B)//in meters
27 //for faintest sounds
28 //density amplitude
29 delta_rhom = delta_p1*(rho_0/B)//in Kg/m^3
30 //displacement amplitude
31 sm = delta_p1/(k*B)//in meters
32
33 printf ("\n\n Wave number k = \n\n %.1f rad/m ",k)
34 printf ("\n\n Density amplitude delta_rho_m = \n\n %
   .1e Kg/m^3 ",delta_rho_m)
35 printf ("\n\n Displacement amplitude s_m = \n\n %.1e
    m ",s_m)
36 printf ("\n\n Density amplitude for faintest sounds
   delta_rhom = \n\n %.1e Kg/m^3 ",delta_rhom)
37 printf ("\n\n Displacement amplitude for faintest
   sounds sm = \n\n %.1e m ",sm)
```

Scilab code Exa 19.2 C19P2

```
1 clear
2 clc
3 //to find intensity  and sound level of sound wave
4
5 // GIVEN:
6 //radiated power
7 p = 25//in W
8 //distance from source
9 r = 2.5//in meters
10 //intensity of sound having sound level 0 dB
11 I0 = 1*10^-12//in W/m^2
12
```

```
13 // SOLUTION
14 //using equation of sound wave
15 //intensity of sound wave
16 I = p/(4*%pi*r^2)//in W/m^2
17 //sound level of sound wave
18 SL = 10*(log10(I/I0))//in dB
19
20 printf ("\n\n Intensity of sound wave I = \n\n %.2f
     W/m^2 ",I)
21 printf ("\n\n Sound level of sound wave SL = \n\n
     %3i dB ",SL)
```

Scilab code Exa 19.3 C19P3

```
1 clear
2 clc
3 //to find wavelength for minimum sound intensity
4
5 // GIVEN:
6 //refer figure 19-6 from page no. 433
7 //distance of listener
8 r2 = 1.2//in meter
9 //distance between two speaker
10 D = 2.3//in meters
11
12 // SOLUTION
13 //using equation of interference of sound wave
14 //using pythagorean formula
15 //distance from speaker 1
16 r1 = sqrt((r2^2)+(D^2))//in meters
17 //difference between distance from two sources
18 r1_minus_r2 = r1-r2//in meters
19 //wavelengths for minimum sound intensity
20 lambda1 = r1_minus_r2*2//in meters
21 lambda2 = (r1_minus_r2*2)/3//in meters
```

```
22 lambda3 = (r1_minus_r2*2)/5//in meters
23
24 printf ("\n\n Distance from speaker 1 r1 = \n\n %.1f
    m ",r1)
25 printf ("\n\n Difference between distance from two
   sources r1_minus_r2 = \n\n %.1f m ",r1_minus_r2)
26 printf ("\n\n Wavelengths for minimum sound
   intensity \n\n lambda = %.1f m,%.2f m,%.2f m ",
   lambda1,lambda2,lambda3)
```

Scilab code Exa 19.4 C19P4

```
1 clear
2 clc
3 //to find speed of sound
4
5 // GIVEN:
6 //refer figure 19-8 from page no. 436
7 //frequeny
8 f = 1080//in Hz
9 //distances of water level at resonance
10 x1 = 6.5//in cm
11 x2 = 22.2//in cm
12 x3 = 37.7//in cm
13
14 // SOLUTION
15 //using equation of sound wave for resonance
16 //from first two resonances
17 half_lambda = x2-x1//in cm
18 //from second and third resonance
19 halflambda = x3-x2//in cm
20 //average of both lambda values
21 half_lambda1 = (half_lambda+halflambda)/2//in cm
22 //wavelength of sound wave
23 lambda = 2*(half_lambda1)//in cm
```

```
24 //speed of sound
25 v = (lambda*10^-2)*f//in m/s //taking lambda in
      meters
26 v = round(v)
27
28 printf ("\n\n From first two resonances half_lambda
      = \n\n %.1f cm ",half_lambda)
29 printf ("\n\n From second and third resonance
      halflambda = \n\n %.1f cm ",halflambda)
30 printf ("\n\n Wavelength of sound wave lambda = \n\n
      %.1f cm ",lambda)
31 printf ("\n\n Speed of sound v = \n\n %3i m/s ",v)
```

Scilab code Exa 19.5 C19P5

```
1 clear
2 clc
3 //to find fundamental frequncy of string
4 //to find fundamental frequncy of string for first
      overtone
5 //to find original frequency
6
7 // GIVEN:
8 //refer figure 19-8 from page no. 436
9 //frequeny
10 f = 440//in Hz
11 //frequency of tuning fork
12 f2 = 3//in Hz
13 //frequency of tuning fork for first overtone
14 f3 = 880//in Hz
15
16 // SOLUTION
17 //using equation of sound wave
18 //fundamental frequncy of string
19 f1 = f+f2//in Hz
```

```
20 f_1 = f-f2//in Hz
21 //frequncy of string for first overtone frequency
22 f4 = f3+(2*f2)//in Hz
23 f_4 = f3-(2*f2)//in Hz
24 //original frequency
25 f5 = f1//in Hz
26
27 printf ("\n\n Fundamental frequncy of string \n\n f1
       = %3i Hz or %3i Hz ",f1,f_1)
28 printf ("\n\n Frequncy of string for first overtone
     frequency \n\n %3i Hz or %3i Hz ",f4,f_4)
29 printf ("\n\n Original frequency = \n\n %3i Hz",f5)
```

Scilab code Exa 19.6 C19P6

```
1 clear
2 clc
3 //to find frequency we would perceive
4
5 // GIVEN:
6 //frequency of siren
7 f = 1125//in Hz
8 //speed of car
9 vs = 29//in m/s
10 //speed of car and your speed
11 v_0 = 14.5//in m/s
12 //speed of sound
13 v = 343//in m/s
14
15
16 // SOLUTION
17 //using equation of sound wave
18 //frequency we would perceiv when police car is
     moving
19 f_dash = f*(v/(v-vs))//in Hz
```

```
f_dash = round(f_dash)
//frequency we would perceiv when your car is moving
v0 = vs//in m/s
fdash = f*((v+v0)/v)//in Hz
//frequency we would perceiv when both police car
   and your car is moving
v0 = v_0
F_dash = f*((v+v0)/(v-v0))//in Hz
//frequency we would perceiv when your car moving at
    9m/s and police car is behind you with 38m/s
v0 = 9//in m/s
vs = 38//in m/s
Fdash = f*((v-v0)/(v-vs))//in Hz
Fdash = round(Fdash)
printf ("\n\n Frequency we would perceiv when police
    car is moving f_dash = \n\n %4i Hz",f_dash)
printf ("\n\n Frequency we would perceiv when your
   car is moving fdash = \n\n %4i Hz",fdash)
printf ("\n\n Frequency we would perceiv when both
   police car and your car is moving F_dash = \n\n
   %4i Hz",F_dash)
printf ("\n\n Frequency we would perceiv when your
   car moving at 9m/s and police car is behind you
   with 38m/s Fdash = \n\n %4i Hz",Fdash)
```

Chapter 20

THE SPECIAL THEORY OF RELATIVITY

Scilab code Exa 20.1 C20P1

```
1 clear
2 clc
3 //to find minimum speed of muon in the Earth's fram
     of reference
4 //to find minimum speed of muon in the muon's fram
     of reference
5
6 //Given:
7 //refer to figure 20-8(a)and (b) from page no. 457
8 //lifetime of muon
9 delta_t0 = 2.2//in microsesonds
10 //height of atmosphere
11 L0 = 100//in Km
12 //speed of light
13 c = 3.00e8//in m/s
14
15 //Solution:
16 //appiying Einstein's posulates
17 //in the Earth's fram of reference
```

```
18 //time of travel
19 delta_t = (L0*10^3)/c//in microseconds
20 //minimum speed of muon
21 u = sqrt((1-((delta_t0/(delta_t)*10^-6)^2)))//in m/s
22
23 //in the muon's fram of reference
24 //height of atmosphere
25 L = c*(delta_t0*10^-6)//in meters
26 //minimum speed of muon
27 u1 = sqrt((1-(((L)/(L0*1000))^2)))//in m/s
28
29 printf ("\n\n Time of travel in the Earth fram of
      reference delta_t = \n\n %.2e seconds" ,delta_t);
30 printf ("\n\n Minimum speed of muon in the Earth
      fram of reference u = \n\n %.6fc" ,u);
31 printf ("\n\n Height of atmosphere in the muon fram
      of reference L = \n\n %3i meters" ,L);
32 printf ("\n\n Minimum speed of muon in the muon fram
       of reference u = \n\n %.6fc" ,u1);
```

Scilab code Exa 20.2 C20P2

```
1 clear
2 clc
3 //to find speed of missile measured by observer on
     the Earth
4
5 //Given:
6 //refer to figure 20-9 from page no. 457
7 //speed of spaceship
8 u = 0.80// times c
9 //speed of missile
10 v0 = 0.60//times c
11
12
```

```
13 //Solution:
14 //appiying formule for relativistic addition of
      velocities
15 //speed of missile measured by observer on Earth
16 v = (v0+u)/(1+(v0*u))//times c
17
18 printf ("\n\n Speed of missile measured by observer
      on the Earth v = \n\n %.2fc" ,v);
```

Scilab code Exa 20.3 C20P3

```
1 clear
2 clc
3 //to find distance between two flashes and time
      between two flashes
4
5 //Given:
6 //seperated distance
7 delta_x = 2.45//in Km
8 //time intervel
9 delta_t = 5.35//in microseconds
10 //speed of frame S'
11 u = 0.855//times c
12 //speed of light
13 c = 3.00e8//in m/s
14
15 //Solution:
16 //appiying Lorentz transformations
17 //Lorentz parameters
18 gama = 1/(sqrt(1-u^2))
19 //refer to table 20-2
20 //using interval transformations
21 //distance between two flashe
22 delta_x_dash = gama*((delta_x*1000)-(u*c*(delta_t
      *10^-6)))//in meters //taking delta_t in seconds
```

```
    and delta_x in meters
23 //time between two flashes
24 delta_t_dash = gama*((delta_t*10^-6)-(u*c*(((delta_x
    *1000))/(c^2))))//in seconds //taking delta_t in
    seconds and delta_x in meters
25 delta_t_dash = nearfloat("succ",-3.147e-6)
26
27 printf ("\n\n Lorentz parameters gama = \n\n %.3f" ,
    gama);
28 printf ("\n\n Distance between two flashe
    delta_x_dash = \n\n %4i meters" ,delta_x_dash);
29 printf ("\n\n Time between two flashes delta_t_dash
    = \n\n %.3e seconds" ,delta_t_dash);
```

Scilab code Exa 20.4 C20P4

```
1 clear
2 clc
3 //to find final velocity of particalas measured in
    the laboratory frame
4
5 //Given:
6 //refer to figure 20-14 from page no. 461
7 //velocity of partical
8 vx_dash = 0.60//times c
9 ////velocity of partical w.r.t. frame moving with it
10 u = 0.60//times c
11 //speed of light
12 c = 3.00e8//in m/s
13
14 //Solution:
15 //appiying transformations of velocities
16 //final velocity of particalas measured in the
    laboratory frame
17 vx = (vx_dash+u)/(1+(u*vx_dash))//times c
```

```
18
19 printf ("\n\n Final velocity of particalas measured
     in the laboratory frame vx = \n\n %.2fc" ,vx);
```

Scilab code Exa 20.5 C20P5

```
1  clear
2  clc
3  //to find time necessary for the rocket to pass
     particular point
4  //to find rest length for the rocket
5  //to find length of D of platform according to
     observer S'
6  //to find time required for S to pass entire length
     according to observer S'
7  //to find //time interval between two events
8  //Given:
9  //refer to figure 20-19(a),(b),(c) from page no. 465
10 //llength of platform
11 L = 65//in meters
12 //relative speed of rocket
13 u = 0.80//times c
14 //speed of light
15 c = 3.00e8//in m/s
16
17 //Solution:
18 //appiying formule for relativity of length
19 //time necessary for the rocket to pass particular
     point
20 delta_t0 = L*10^6/(u*c)//in microseconds
21 //rest length for the rocket
22 L0 = L/(sqrt(1-(u^2)))//in meters
23 //length of D of platform according to observer S'
24 D0 = L
25 D = D0*(sqrt(1-(u^2)))//in meters
```

```
26 D = round(D)
27 //time required for S to pass entire length
      according to observer S'
28 delta_t_dash = L0*10^6/(u*c)//in microseconds
29 //time measured by S and S' usind time dilation
      formula
30 delta_tdash = delta_t0/(sqrt(1-(u^2)))//in
      microseconds
31 //refer to table 20-2
32 //time interval between two events
33 deltat_dash = -(u*c*(-L))*10^6/((c^2)*(sqrt(1-(u^2))
      ))//in microseconds
34 //time interval between two events according to S'
35 deltatdash = (L0-D)*10^6/(u*c)//in microseconds
36
37 printf ("\n\n Time necessary for the rocket to pass
      particular point delta_t0 = \n\n %.2f
      microseconds" ,delta_t0);
38 printf ("\n\n Rest length for the rocket L0 = \n\n
      %3i meters" ,L0);
39 printf ("\n\n Length of D of platform according to
      observer S-dash D = \n\n %2i meters" ,D);
40 printf ("\n\n Time required for S to pass entire
      length according to observer S-dash delta_t_dash
      = \n\n %.2f microseconds" ,delta_t_dash);
41 printf ("\n\n Time measured by S and S-dash usind
      time dilation formula delta_tdash = \n\n %.2f
      microseconds" ,delta_tdash);
42 printf ("\n\n Time interval between two events
      deltat_dash = \n\n %.2f microseconds" ,
      deltat_dash);
43 printf ("\n\n Time interval between two events
      according to S-dash deltatdash = \n\n %.2f
      microseconds" ,deltatdash);
```

Scilab code Exa 20.6 C20P6

```
clear
clc
//to find momentum of proton

//Given:
//speed of proton
v = 0.86//times c
//speed of light
c = 3.00e8//in m/s
//mass of proton
m = 1.67e-27//in Kg

//Solution:
//appiying fomule for relativistic momentum
//momentum of proton
P = (m*v*c)/(sqrt(1-(v^2)))//in Kg.m/s
//value of pc
Pc = P*c*(6.24e12)//in MeV //(6.24e12) is conversion
    factor between J and MeV

printf ("\n\n Momentum of proton P = \n\n %.2e Kg.m/
   s" ,P);
printf ("\n\n Value of pc = \n\n %4i MeV" ,Pc);
printf ("\n\n Momentum of proton p = \n\n %4i MeV/c"
    ,Pc);
```

Scilab code Exa 20.7 C20P7

```
clear
clc
//to find speed of electron as fraction of c and as
   difference from c

```

```
5
6 //Given:
7 //kinetic energy of electron
8 K = 50//in GeV
9 //value of mc_square
10 mc_square = 0.511e-3//in GeV
11 //speed of light
12 c = 3.00e8//in m/s
13
14 //Solution:
15 //appiying fomule for relativistic energy
16 //speed of electron as fraction of c
17 v = sqrt(1-(1/(1+(K/mc_square)^2)))//times c
18 //speed of electron as difference from c
19 c_minus_v = (5.2e-11)*c//in m/s
20
21 printf ("\n\n Speed of electron as fraction of c v =
       \n\n %.12fc" ,v);
22 printf ("\n\n Speed of electron as difference from c
       c_minus_v = \n\n %.3f m/s" ,c_minus_v);
```

Scilab code Exa 20.8 C20P8

```
1 clear
2 clc
3 //to find difference between masses of combined ball
      from sum of masses of original balls
4
5 //Given:
6 //mass of ball
7 m = 35//in gram
8 //speed of ball
9 v = 1.7//in m/s
10 //speed of light
11 c = 3.00e8//in m/s
```

```
12
13 //Solution:
14 //appiying fomule for energy and mass in special
      relativity
15 //applying conservation of energy
16 //increase in rest energy
17 delta_E0 = 2*((1/2)*(m*10^-3)*(v^2))//in J //taking
      mass in Kg
18 //increase in mass
19 delta_m = delta_E0/(c^2)//in Kg
20
21 printf ("\n\n Increase in rest energy delta_E0 = \n\
      n %.3f J" ,delta_E0);
22 printf ("\n\n Difference between masses of combined
      ball from sum of masses of original balls delta_m
       = \n\n %.1e Kg" ,delta_m);
```

Scilab code Exa 20.9 C20P9

```
1 clear
2 clc
3 //to find kinetic energy needed to produce Z0
4
5 //Given:
6 //refer to sample problem 20-8
7 //rest energy
8 E0 = 91.2//in GeV
9 //rest energy of electron and positron
10 E = 0.511//in MeV
11 //speed of light
12 c = 3.00e8//in m/s
13
14 //Solution:
15 //appiying fomule for energy and mass in special
      relativity
```

```
16 //change in rest energy
17 delta_E0 = E0-(2*(E*10^-3))//in GeV //coveting E
      into GeV
18 //applying conservation of energy
19 //kinetic energy needed to produce Z0
20 delta_K = -(delta_E0)//in GeV
21
22 printf ("\n\n Change in rest energy delta_E0 = \n\n
      %.1f GeV" ,delta_E0);
23 printf ("\n\n Kinetic energy needed to produce Z0
      delta_K = \n\n %.1f GeV" ,delta_K);
```

Scilab code Exa 20.10 C20P10

```
1 clear
2 clc
3 //to find kinetic energy of each pion
4
5 //Given:
6 //value of mc^2 for Kaon
7 mk_c_square = 498//in MeV
8 //kinetic energy of Kaon
9 K = 325//in MeV
10 ////value of mc^2 for pion
11 mpi_c_square = 140//in MeV
12 //speed of light
13 c = 3.00e8//in m/s
14
15 //Solution:
16 //appiying fomule for coservation of total
      relativistic energy
17 //applying conservation of energy
18 //initial total relativistic energy
19 Ek = K+mk_c_square//in MeV
20 //total initial momentum
```

```
21 pk_c = sqrt((Ek^2)-(mk_c_square)^2)//in MeV
22 //total energy of final system
23 E = Ek//in MeV
24 //applying conservation of momentum
25 //value of p1c
26 p1c = 668//in MeV
27 p_1c = -13//in MeV
28 //kinetic energy of each pion
29 //kinetic energy of first pion
30 K1 = (sqrt((p1c^2)+(mpi_c_square^2)))-mpi_c_square//
   in MeV
31 //kinetic energy of second pion
32 K2 = (sqrt((p_1c^2)+(mpi_c_square^2)))-mpi_c_square
   //in MeV
33 K1 = round(K1)
34
35 printf ("\n\n Initial total relativistic energy Ek =
    \n\n %3i MeV" ,Ek);
36 printf ("\n\n Total initial momentum pk_c = \n\n %3i
    MeV" ,pk_c);
37 printf ("\n\n Total energy of final system E = \n\n
   %3i MeV" ,E);
38 printf ("\n\n Value of p1c = \n\n %3i MeV or %3i MeV
   " ,p1c,p_1c);
39 printf ("\n\n Kinetic energy of first pion K1 = \n\n
    %3i MeV" ,K1);
40 printf ("\n\n Kinetic energy of second pion K2 = \n\
   n %.1f MeV" ,K2);
```

Scilab code Exa 20.11 C20P11

```
1 clear
2 clc
3 //to find threshold kinetic energy to produce
   antiproton
```

```
4
5  //Given:
6  //refer to figure 20-23 from page no. 470
7  //rest energy of proton
8  mp_c_square = 938//in MeV
9  //speed of light
10 c = 3.00e8//in m/s
11
12 //Solution:
13 //appiying fomule for relativistic momentum
14 //applying conservation of energy
15 //value of mpc^2/E1'
16 mpc_square_by_E1dash = 1/2
17 //value of v1'/c
18 v1_dash_by_c = sqrt(1-(mpc_square_by_E1dash)^2)
19 //refer to table 20-3
20 //speed of incident proton
21 v_dash = v1_dash_by_c//times c
22 u = v1_dash_by_c//times c
23 v = (v_dash+u)/(1+(v1_dash_by_c)^2)//times c
24 //total energy of incident proton
25 E = 1/(sqrt(1-(v^2)))//times mp_c_square
26 E = round(E)
27 //threshold kinetic energy to produce antiproton
28 K = (E*mp_c_square)-mp_c_square//in MeV
29
30 printf ("\n\n Value of v1_dash/c = \n\n %.3f" ,
      v1_dash_by_c);
31 printf ("\n\n Speed of incident proton v = \n\n %.3
      fc" ,v);
32 printf ("\n\n Total energy of incident proton E = \n
      \n %1imp_c_square ",E);
33 printf ("\n\n Threshold kinetic energy to produce
      antiproton K = \n\n %4i MeV",K);
```

Chapter 21

TEMPERATURE

Scilab code Exa 21.1 C21P1

```
clear
clc
//to find temperature measured by thermometer

//Given:
//factor by which resistance is increased
R_by_Rtr = 1.392
//temperature of triple point of water
Ttr = 273.16//in K

//Solution:
//using formula for measuring temperatures
//temperature measured by thermometer
T_R = Ttr*R_by_Rtr//in K

printf ("\n\n Temperature measured by thermometer
   T_R = \n\n %.1f K" ,T_R);
```

Scilab code Exa 21.2 C21P2

```
clear
clc
//to find maximum temperature variation allowable
   during ruling

//Given:
//refer to table 21-3
//accuracy for milimeter interval
delta_L = 5e-5//in mm
//coefficient of linear expansion
alpha = 11e-6//in per degree celsius
//consider length of steel
L = 1//in mm

//Solution:
//using formula for temperature expansion
//maximum temperature variation allowable during
   ruling
delta_T = delta_L/(alpha*L)//in degree celsius

printf ("\n\n Maximum temperature variation
   allowable during ruling delta_T = \n\n %.1f
   degree celsius" ,delta_T);
```

Scilab code Exa 21.3 C21P3

```
clear
clc
//to find final pressure of gas

//Given:
//refer to figure 21-13 from page 488
//initial temperature of oxygen
Ti = 20//in degree celsius
//initial pressure of oxygen
```

```
10 pi = 15//in atm
11 //initial volume of oxygen
12 vi = 22//in liters
13 //final temperature of oxygen
14 Tf = 25//in degree celsius
15 //final volume of oxygen
16 vf = 16//in liters
17
18 //Solution:
19 //consider oxygen as ideal gas and applying
      equations of ideal gas
20 //final pressure of gas
21 pf = pi*((Tf+273)/(Ti+273))*(vi/vf)//in atm //taking
      temp. in kelvin
22 pf = round(pf)
23
24 printf ("\n\n Final pressure of gas pf = \n\n %2i
      atm" ,pf);
```

Chapter 22

MOLECULAR PROPERTIES OF GASES

Scilab code Exa 22.1 C22P1

```
1 clear
2 clc
3 //to find root mean square speed of hydrogen
     molecule
4
5 //Given:
6 //pressure
7 p = 1//in atm
8 //density of hydrogen
9 rho = 8.99e-2//in Kg/m^3
10
11 //Solution:
12 //assume hydron as ideal gas
13 //applying formula of root mean square speed for
     ideal gas
14 //root mean square speed of hydrogen molecule
15 vrms = sqrt((3*p*1.01e5)/(rho))//in m/s //taking
     pressure in Pa
16
```

```
17 //answer of vrms is slightly different than book
     answer.But ans. by scilab program is same as that
     of calculator
18 printf ("\n\n Root mean square speed of hydrogen
     molecule vrms = \n\n %4i m/s" ,vrms);
```

Scilab code Exa 22.2 C22P2

```
1  clear
2  clc
3  //to find number of moles of oxygen
4  //to find number  of molecules of oxygen
5  //to find approximate rate at which oxygen molecule
     strike one face of the box
6
7  //Given:
8  //refer to figure 22-2 from page no. 499
9  //length of edge of cubical box
10 L = 10//in cm
11 //pressure of oxygen
12 p = 1.0//in atm
13 //temperature of oxygen
14 T = 300//in K
15 //molar gas constant
16 R = 8.31//in J/mol.K
17 //Avogadro constant
18 NA = 6.02e23//in molecules/mol
19
20 //Solution:
21 ////assumong oxygen as ideal gas
22 //applying ideal gas equations
23 //volume of box
24 V = ((L*10^-2)^3)//in m^3
25 //number  of moles of oxygen
26 n = ((p*1.01*10^5)*V)/(R*T)//taking p into Pa
```

```
27 //number  of molecules of oxygen
28 N = n*NA
29 N = nearfloat("succ",2.5e22)
30 //refer to table 22-1
31 //root mean square speed of oxygen
32 vrms = 483//in m/s
33 //approximate rate at which oxygen molecule strike
      one face of the box
34 Rate = (N*vrms)/(6*(L*10^-2))//in collisions/s
35
36 printf ("\n\n Number  of moles of oxygen n = \n\n %
      .3f mol" ,n);
37 printf ("\n\n Number  of molecules of oxygen N = \n\
      n %.1e molecules" ,N);
38 printf ("\n\n Root mean square speed of oxygen vrms
      = \n\n %3i m/s" ,vrms);
39 printf ("\n\n Approximate rate at which oxygen
      molecule strike one face of the box Rate = \n\n %
      .1e collisions/s" ,Rate);
```

Scilab code Exa 22.3 C22P3

```
1 clear
2 clc
3 //to find ratio of rms speed of gas molecules
      containing 235-U and gas molecules containing
      238-U
4 //to find relative abundance of gas molecules
      containing 235-U
5 //to find number of times gas molecule should be
      passed through barrier
6
7 //Given:
8 //abundance of 235-U
9 a1 = 0.7//in percentage
```

```
10 //abundance of 238-U
11 a2 = 99.3//in percentage
12 //final abundance of 235-U
13 a3 = 3//in percentage
14
15 //Solution:
16 //applying equations for root mean square speed
17 //molecular mass of 235-U
18 m_235 = 235+6*(19)//in u
19 //molecular mass of 238-U
20 m_238 = 238+6*(19)//in u
21 //ratio of rms speed of gas molecules containing
     235-U and gas molecules containing 238-U
22 vrms_235_by_vrms_238 = sqrt(m_238/m_235)
23 //ratio of abundances
24 r = a1/a2
25 //relative abundance of gas molecules containing
     235-U
26 ratio_1_pass = r*vrms_235_by_vrms_238
27 //isotope ratio
28 i = (a3)/(100-(a3))
29 //number of times gas molecule should be passed
     through barrier
30 n = (log(i/r))/(log(vrms_235_by_vrms_238))
31
32 printf ("\n\n Molecular mass of 235-U m_235 = \n\n
     %3i u" ,m_235);
33 printf ("\n\n Molecular mass of 238-U m_238 = \n\n
     %3i u" ,m_238);
34 printf ("\n\n Ratio of rms speed of gas molecules
     containing 235-U and gas molecules containing
     238-U vrms_235_by_vrms_238 = \n\n %.4f" ,
     vrms_235_by_vrms_238);
35 printf ("\n\n Ratio of abundances = \n\n %.5f" ,r);
36 printf ("\n\n Relative abundance of gas molecules
     containing 235-U ratio_1_pass = \n\n %.5f" ,
     ratio_1_pass);
37 printf ("\n\n Isotope ratio = \n\n %.5f" ,i);
```

```
38 //answer of n slightly changes than book.But answer
     by scilab answer is same as that of answer by
     calculator
39 printf ("\n\n Number of times gas molecule should be
     passed through barrier = \n\n %3i" ,n);
```

Scilab code Exa 22.4 C22P4

```
1  clear
2  clc
3  //to find mean free path and average collision rate
     of nytrogen at room temperature
4
5  //Given:
6  //room temperature
7  T = 300//in K
8  //atmospheric pressure
9  p = 1.01e5//in Pa
10 //effective diameter of nytrogen
11 d = 3.15e-10//in meters
12 //average speed
13 vav = 478//in m/s
14 //Boltzmann constant
15 k = 1.38e-23//in J/K
16
17 //Solution:
18 //applying formula of mean path
19 //mean free path of nytrogen at room temperature
20 lambda = (k*T)/(sqrt(2)*%pi*(d^2)*p)//in meters
21 //average collision rate of nytrogen at room
     temperature
22 rate = vav/lambda//in collisions/second
23
24 printf ("\n\n Mean free path of nytrogen at room
     temperature lambda = \n\n %.1e meters" ,lambda);
```

```
25 printf ("\n\n Average collision rate of nytrogen at
     room temperature rate = \n\n %.1e collisions/
     second" ,rate);
```

Scilab code Exa 22.5 C22P5

```
1  clear
2  clc
3  //to find average speed,root-mean speed,root-mean
     square speed and most probable speed of particals
4
5  //Given:
6  //number of particals
7  N = 10
8  //speed of particals
9  v1 = 0.0//in m/s
10 v2 = 1.0//in m/s
11 v3 = 2.0//in m/s
12 v4 = 3.0//in m/s
13 v5 = 3.0//in m/s
14 v6 = 3.0//in m/s
15 v7 = 4.0//in m/s
16 v8 = 4.0//in m/s
17 v9 = 5.0//in m/s
18 v10 = 6.0//in m/s
19
20 //Solution:
21 //applying formula for average speed
22 //average speed of particals
23 vav = (1/N)*(v1+v2+v3+v4+v5+v6+v7+v8+v9+v10)//in m/s
24 //applying formula for root-mean speed
25 //root-mean speed of particals
26 v_square_av = (1/N)*(v1^2+v2^2+v3^2+v4^2+v5^2+v6^2+
     v7^2+v8^2+v9^2+v10^2)//in m^2/s^2
27 //applying formula for root-mean square speed
```

```
28 //root-mean square speed of particals
29 vrms = sqrt(v_square_av)//in m/s
30 //most probable speed of particals
31 //taking into consideration all speeds of particals
32 vp = v4//in m/s
33 printf ("\n\n Average speed of particals vav = \n\n
   %.1f m/s" ,vav);
34 printf ("\n\n Root-mean speed of particals
   v_square_av = \n\n %.1f m^2/s^2" ,v_square_av);
35 printf ("\n\n Root-mean square speed of particals
   vrms = \n\n %.1f m/s" ,vrms);
36 printf ("\n\n Most probable speed of particals vp =
   \n\n %.1f m/s" ,vp);
```

Scilab code Exa 22.6 C22P6

```
1 clear
2 clc
3 //to find fraction of molecules having speed in
   range 599-601m/s
4
5 //Given:
6 //temperature
7 T = 300//in K
8 //molar mass of oxygen
9 M = 0.032//in Kg/mol
10 //molar gas constant
11 R = 8.31//in J/mol.K
12 //velocity
13 v = 600//in m/s
14
15 //Solution:
16 //fraction of molecules having speed in range
   599-601m/s
17 //difference in speed
```

```
18 dv = 2//in m/s
19 f = 4*%pi*((M/(2*%pi*R*T))^(3/2))*(v^2)*%e^((-M*(v
      ^2)/(2*R*T)))*dv
20 f1 = f*100//in percent
21
22 printf ("\n\n Fraction of molecules having speed in
      range 599-601m/s f = \n\n %.1e" ,f);
23 printf ("\n\n Percentage of molecules having speed
      in range 599-601m/s f = \n\n %.2f percent" ,f1);
```

Scilab code Exa 22.7 C22P7

```
1 clear
2 clc
3 //to find most probable speed,average speed,root-
      mean square speed of oxygen
4
5 //Given:
6 //temperature
7 T = 300//in K
8 //molar gas constant
9 R = 8.31//in J/mol.K
10 //molar mass
11 M = 0.032//in Kg/mol
12
13 //Solution:
14 //applying formula for most probable speed
15 //most probable speed of oxygen
16 vp = sqrt((2*R*T)/(M))//in m/s
17 //applying formula for average speed
18 //average speed of oxygen
19 vav = sqrt((8*R*T)/(%pi*M))//in m/s
20 //applying formula for root-mean square speed
21 //root-mean square speed of oxygen
22 vrms = sqrt((3*R*T)/(M))//in m/s
```

```
23 vp = round(vp)
24 a = vav/vp
25 a1 = vrms/vp
26 a1 = nearfloat("succ",1.225)
27
28 printf ("\n\n Most probable speed of oxygen vp = \n\
   n %3i m/s" ,vp);
29 printf ("\n\n Average speed of oxygen vav = \n\n %3i
    m/s" ,vav);
30 printf ("\n\n Root-mean square speed of oxygen vrms
   = \n\n %3i m/s" ,vrms);
31 printf ("\n\n For any gas vp:vav:vrms = 1:%.3f:%.3f"
    ,a,a1);
```

Scilab code Exa 22.9 C22P9

```
1 clear
2 clc
3 //to find pressure according to ideal gas law
4 //to find pressure according to van der Waals
   equations
5
6 //Given:
7 //for qxygen van der Waals coefficients
8 a = 0.138//in J.m^3/mol^2
9 b = 3.18e-5//in m^3/mol
10 //number mol of oxygen
11 n = 1//in mol
12 //volume of box
13 V = 0.0224//in m^3
14 //molar gas constant
15 R = 8.31//in J/mol.K
16 //molar mass
17 M = 0.032//in Kg/mol
18 //temperature
```

```
19 T = 50//in K
20
21 //Solution:
22 //applying ideal gas equation
23 //pressure according to ideal gas law
24 p = (n*R*T)/V//IN Pa
25 //applying van der Waals equations
26 //pressure according to van der Waals equations
27 P = ((n*R*T)/(V-(n*b)) )-((a*n^2)/V^2)//in Pa
28
29 printf ("\n\n Pressure according to ideal gas law p
     = \n\n %.2e Pa" ,p);
30 printf ("\n\n Pressure according to van der Waals
     equations P = \n\n %.2e Pa" ,P);
```

Chapter 23

THE FIRST LAW OF THERMODYNAMICS

Scilab code Exa 23.2 C23P2

```
1 clear
2 clc
3 //to find rate of heat energy pass through the
     insulation
4 //to find additional insulation required to reduce
     heat transfer rate by half
5
6 //Given:
7 //refer to figure 23-6 from page no. 520
8 ////temperature of steam
9 TS = 100//in degree celsius
10 //diameter of pipe
11 d = 5.4//in cm
12 //thickness of insulation
13 t = 5.2//in cm
14 //length of pipe
15 D = 6.2//in meters
16 //temperature of room
17 TR = 11//in degree celsius
```

```
18 //thermal conductivity
19 k = 0.048//in W/m.K
20
21 //Solution:
22 //radius of cylinder
23 r1 = d/2//in cm
24 //radius of cylinder with insulation
25 r2 = r1+t//in cm
26 //applying fourier's law of heat conduction
27 //rate of heat energy pass through the insulation
28 H = (2*%pi*k*D*(TS-TR))/(log(r2/r1))//in W
29 //additional insulation required to reduce heat
      transfer rate by half
30 r2_dash = (r2^2)/r1//in cm
31
32 printf ("\n\n Rate of heat energy pass through the
      insulation H = \n\n %3i W" ,H);
33 printf ("\n\n Additional insulation required to
      reduce heat transfer rate by half r2_dash = \n\n
      %2i cm" ,r2_dash);
```

Scilab code Exa 23.3 C23P3

```
1 clear
2 clc
3 //to find final equilibrium temperature of the
      system
4
5 //Given:
6 //mass of copper cube
7 mc = 75//in gram
8 //temperature of oven
9 T0 = 312//in degree celsius
10 //mass of water
11 mw = 220//in gram
```

```
12 //heat capacity of beaker
13 Cb = 190//in J/K
14 //intial temperature of water and beaker
15 Ti = 12.0//in degree celsius
16 //heat capacity of water
17 Cw = 4190//in J/Kg.K
18 //heat capacity of copper cube
19 Cc = 387//in J/Kg.K
20
21 //Solution:
22 //applying laws of thermodynamics
23 //for equilibrium condition
24 //final equilibrium temperature of the system
25 Tf = (((mw*10^-3)*Cw*Ti)+(Cb*Ti)+((mc*10^-3)*Cc*T0))
      /(((mw*10^-3)*Cw)+(Cb)+((mc*10^-3)*Cc))//in
      degree celsius //taking masses in Kg
26 //heat transfer for water
27 Qw = (mw*10^-3)*Cw*(Tf-Ti)//in J
28 //heat transfer for beaker
29 Qb = Cb*(Tf-Ti)//in J
30 //heat transfer for copper
31 Qc = (mc*10^-3)*Cc*(Tf-T0)//in J
32 Qw = nearfloat("pred",7011)
33 Qb = nearfloat("pred",1441)
34 Qc = nearfloat("pred",-8450)
35
36 printf ("\n\n Final equilibrium temperature of the
      system Tf = \n\n %.1f degree celsius" ,Tf);
37 printf ("\n\n Heat transfer for water Qw = \n\n %4i
      J" ,Qw);
38 printf ("\n\n Heat transfer for beaker Qb = \n\n %4i
      J" ,Qb);
39 printf ("\n\n Heat transfer for copper Qc = \n\n %4i
      J" ,Qc);
```

Scilab code Exa 23.4 C23P4

```
clear
clc
//to find work done by three different paths

//Given:
//refer to figure 23-17 from page no. 529
//final volume
vf = 1.0//in m^3
//initial volume
vi = 4.0//in m^3
//final pressure
pf = 40//in Pa
//initialvolume
pi = 10//in Pa

//Solution:
//applying laws of thermodynamics
//work done by constant pressure in path 1
W = -pi*(vf-vi)//in J
//work done in constant volume in path 1
w = 0//in J
//work done by path 1
W1 = W+w//in J
//work done by path 2
W2 = -pi*vi*(log(vf/vi))//in J
//work done by path 3
W3 = 0-(pf*(vf-vi))//in J

printf ("\n\n Work done by constant pressure in path
    1 W = \n\n %2i J" ,W);
printf ("\n\n Work done by path 1 W1 = \n\n %2i J" ,
   W1);
printf ("\n\n Work done by path 2 W2 = \n\n %2i J" ,
   W2);
printf ("\n\n Work done by path 3 W3 = \n\n %2i J" ,
   W3);
```

Scilab code Exa 23.5 C23P5

```
clear
clc
//to find speed of sound in the gas

//Given:
//room temperature
T = 20//in degree celsius
//parameter gama for air
gama = 1.4
//molar gas constant
R = 8.31//in J/mol.K
//molar mass for air
M = 0.0290//in Kg/mol

//Solution:
//applying laws of thermodynamics
//speed of sound in the gas
v = sqrt((gama*R*(T+273))/M)//in m/s
v = round(v)

printf ("\n\n Speed of sound in the gas v = \n\n %3i
    m/s" ,v);
```

Scilab code Exa 23.6 C23P6

```
clear
clc
//to find time required for room temperature to be
   21 degree celsius
```

```
4
5 //Given:
6 //room temperature
7 T = 0//in degree celsius
8 //length of room
9 l = 6//in meters
10 //breadth of room
11 b = 4//in meters
12 //height of room
13 h = 3//in meters
14 //power of heater
15 p = 2//in KW
16 //final air temperature
17 T1 = 21//in degree celsius
18
19 //Solution:
20 //applying laws of thermodynamics
21 //volume of room
22 V = (l*b*h)*1000//in L
23 //number of moles of gas
24 n = V/22.4//in mol //since 1 mol occupies 22.4L of
     volume
25 //refer to table 23-4
26 //molar heat capacity
27 Cv = 20.8//in J/mol.K
28 //using relation of heat capacity
29 //absorbtion of heat take place
30 Q = n*Cv*(T1-T)//in J
31 //time required for room temperature to be 21 degree
     celsius
32 t = Q/(p*10^3)//in seconds //taking power in W
33 t = nearfloat("pred",701)
34
35 printf ("\n\n Volume of room V = \n\n %5i L" ,V);
36 printf ("\n\n Number of moles of gas n  = \n\n %.1e
     mol" ,n);
37 printf ("\n\n Absorbtion of heat take place Q = \n\n
     %.1e J" ,Q);
```

```
38 printf ("\n\n Time required for room temperature to
      be 21 degree celsius t = \n\n %3i seconds" ,t);
```

Scilab code Exa 23.7 C23P7

```
1  clear
2  clc
3  //to find change in internal energy
4
5  //Given:
6  //refer to figure 23-17 from page no. 529
7  //refer to problem 23-4
8  //final volume
9  vf = 1.0//in m^3
10 //initial volume
11 vi = 4.0//in m^3
12 //initialvolume
13 pi = 10//in Pa
14 //value of constant for monoatomic gas
15 gama = 1.66
16 //number of moles of ideal gas
17 n = 0.11//in mol
18 //molar gas constant
19 R = 8.31//in J/mol.K
20
21 //Solution:
22 //applying laws of thermodynamics
23 //applying adiabatic relationship
24 //final pressure of gas
25 pf = (pi*(vi^gama))/(vf^gama)//in Pa
26 //initial temperature of gas
27 Ti = (pi*vi)/(n*R)//in K
28 //final temperature of gas
29 Tf = (pf*vf)/(n*R)//in K
30 //applying internal energy formula
```

```
31 //change in internal energy
32 delta_Eint = (3/2)*(n*R*(Tf-Ti))//in J
33 pf = round(pf)
34 Ti = round(Ti)
35
36 printf ("\n\n Final pressure of gas pf = \n\n %3i Pa
     " ,pf);
37 printf ("\n\n Initial temperature of gas Ti = \n\n
     %2i K" ,Ti);
38 printf ("\n\n Final temperature of gas Tf = \n\n %3i
     K" ,Tf);
39 printf ("\n\n Change in internal energy delta_Eint =
     \n\n %2i J" ,delta_Eint);
```

Scilab code Exa 23.8 C23P8

```
1 clear
2 clc
3 //to find work done on the system
4 //to find heat added to the system
5 //to find change in internal energy of the system
6
7 //Given:
8 //refer to figure 23-23 from page no. 535
9 //mass of water
10 m = 1.00//in Kg
11 //initial volume of liquid
12 vi = 1.00e-3//in m^3
13 //final volume of steam
14 vf = 1.671//in m^3
15 //atmospheric pressure
16 p = 1.01e5//in Pa
17 //molar gas constant
18 R = 8.31//in J/mol.K
19
```

```
20 //Solution:
21 //applying laws of thermodynamics
22 //applying constant pressure relationship
23 //work done on the system
24 W = (-p*(vf-vi))//in KJ
25 //latent heat of vaporization
26 L = 2256//in KJ/Kg
27 //heat added to the system
28 Q = L*m//in KJ
29 //change in internal energy of the system
30 delta_Eint = Q+W//in KJ
31
32 printf ("\n\n Work done on the system W = \n\n %.2e
     J" ,W);
33 //answer of Q and delta_Eint slightli changes.But
     answer by scilab program is same as that of
     calculator answer
34 printf ("\n\n Heat added to the system Q = \n\n %4i
     KJ" ,Q);
35 printf ("\n\n Change in internal energy of the
     system delta_Eint = \n\n %4i KJ" ,delta_Eint);
```

Scilab code Exa 23.9 C23P9

```
1 clear
2 clc
3 //to find work done on the system
4 //to find heat added to the system
5 //to find change in internal energy of the system
6
7 //Given:
8 //refer to figure 23-21 from page no. 534
9 //number of moles
10 n = 0.75//in mol
11 //pressures at corresponding points
```

```
PA = 3.2e3//in Pa
PB = 1.2e3//in Pa
//volume at corresponding point
VA = 0.21//in m^3
//molar gas constant
R = 8.31//in J/mol.K
//value of constants
Cv = 20.8//in J/mol.K
Cp = 29.1//in J/mol.K

//Solution:
//applying laws of thermodynamics
//using ideal gas law
//temperature at A
TA = (PA*VA)/(n*R)//in K
////temperature at B
TB = (PB*VA)/(n*R)//in K //since VA=VB
//volume at C
VC = (n*R*TA)/(PB)//in m^3 //since TC = TA and PC =
   PB
//during process A-B
//applying constant volume relationship
//heat added to the system
//redefining TA AND TB
TA = 108//in K
TB = 40//in K
Q1 = n*Cv*(TB-TA)//in J
//work done on the system
W1 = 0//in J
//change in internal energy of the system
delta_Eint1 = Q1+W1//in J

//during process B-C
//applying constant pressure relationship
//heat added to the system
Q2 = n*Cp*(TA-TB)//in J //since TC = TA
//work done on the system
W2 = -PB*(VC-VA)//in J //since VB = VA
```

```
49 //change in internal energy of the system
50 delta_Eint2 = Q2+W2//in J
51
52 //during process C-A
53 //applying isothermal relationship
54 //work done on the system
55 W3 = -n*R*TA*(log(VA/VC))//in J
56 //change in internal energy of the system
57 delta_Eint3 = 0//in J
58 //heat added to the system
59 Q3 = delta_Eint3-W3 //in J
60 //delta_Eint1 = nearfloat("succ",-1061)
61 //Q2 = nearfloat("succ",1480)
62 //delta_Eint2 = nearfloat("succ",1060)
63 //W3 = nearfloat("succ",660)
64 //Q3 = nearfloat("succ",-661)
65 //totol work done during process
66 W = W1+W2+W3//in J
67 //total change in internal energy during process
68 delta_Eint = delta_Eint1+delta_Eint2+delta_Eint3//in
       J
69 TA = round(TA)
70 //value of Q2,delta_Eint2,delta_E slightly varies
      than book.But answer by scilab is same as that of
       calculator answer
71
72 printf ("\n\n Temperature at A TA = \n\n %3i K" ,TA)
      ;
73 printf ("\n\n Temperature at B TB = \n\n %3i K" ,TB)
      ;
74 printf ("\n\n Volume at C VC = \n\n %.2f m^3" ,VC);
75 printf ("\n\n During process A-B");
76 printf ("\n\n Heat added to the system Q1 = \n\n %4i
       J" ,Q1);
77 printf ("\n\n Work done on the system W1 = \n\n %3i
      J" ,W1);
78 printf ("\n\n Change in internal energy of the
      system delta_Eint1 = \n\n %4i J" ,delta_Eint1);
```

```
79 printf ("\n\n During process B-C");
80 printf ("\n\n Heat added to the system Q2 = \n\n %4i
       J" ,Q2);
81 printf ("\n\n Work done on the system W2 = \n\n %3i
      J" ,W2);
82 printf ("\n\n Change in internal energy of the
      system delta_Eint2 = \n\n %4i J" ,delta_Eint2);
83 printf ("\n\n During process C-A");
84 printf ("\n\n Heat added to the system Q3 = \n\n %4i
       J" ,Q3);
85 printf ("\n\n Work done on the system W3 = \n\n %3i
      J" ,W3);
86 printf ("\n\n Change in internal energy of the
      system delta_Eint3 = \n\n %4i J" ,delta_Eint3);
87 printf ("\n\n Totol work done during process W = \n\
      n %3i J" ,W);
88 printf ("\n\n Total change in internal energy during
       process delta_Eint = \n\n %4i J" ,delta_Eint);
```

Chapter 24

ENTROPY AND THE SECOND LAW OF THERMODYNAMICS

Scilab code Exa 24.1 C24P1

```
clear
clc
//to find entropy change of water during process

//Given:
//mass of water
m = 1.8//in Kg
//initial temperature of water and hot plate
Ti = 20//in degree celsius
//final temperature of hot plate
Tf = 100//in degree celsius
//heat capacity of water
c = 4190//in J/Kg.K

//Solution:
//applying laws of thermodynamics
//applying formula for entropy change
```

```
18 //entropy change of water during process
19 delta_S = m*c*(log((Tf+273)/(Ti+273)))//in J/K //
      taking temperatures in K
20
21 printf ("\n\n Entropy change of water during process
      delta_S = \n\n %4i J/K" ,delta_S);
```

Scilab code Exa 24.2 C24P2

```
1 clear
2 clc
3 //to find temperature rise of system water+stone
4 //to find entropy change of system
5 //to find entropy change of reverse process
6
7 //Given:
8 //refer to figure 24-1 from page no. 548
9 //mass of stone
10 ms = 1.5//in Kg
11 //mass of water
12 mw = 4.5//in Kg
13 //vertical height
14 h = 2.5//in meters
15 //initial temperature of water and stone
16 T = 300//in K
17 //specific heat capacity of water
18 cw = 4190//in J/Kg.K
19 //specific heat capacity of stone material
20 cs = 790//in J/Kg.K
21 //acceleration due to gravity
22 g = 9.8//in m/s^2
23
24 //Solution:
25 //applying laws of thermodynamics
26 //applying formula for entropy change for
```

```
    irreversible process
//heat transfer
Q = mw*g*h//in J
//temperature rise of system water+stone
delta_T = Q/((mw*cw)+(ms*cs))//in K
//entropy change of system
delta_S = Q/(T)//on J/K
//entropy change of reverse process
delta_s = -Q/T//in J/k //since heat is extracted
    from system

printf ("\n\n Heat transfer Q = \n\n %3i J" ,Q);
printf ("\n\n Temperature rise of system water+stone
     delta_T = \n\n %.1e K" ,delta_T);
printf ("\n\n Entropy change of system delta_S = \n\
    n %.2f J/k" ,delta_S);
printf ("\n\n Entropy change of reverse process
    delta_s = \n\n %.2f J/K" ,delta_s);
```

Scilab code Exa 24.3 C24P3

```
clear
clc
//to find net entropy change of irreverse process

//Given:
//refer to figure 24-3(a)and (b) from page no. 549
//mass of hot water
m = 0.57//in Kg
//initial temperature of hot water
TiH = 363//in K
//initial temperature of cold water
TiC = 283//in K
//equilibrium temperature
Tf = 323//in K
```

```
15 //specific heat capacity of water
16 c = 4190//in J/Kg.K
17
18 //Solution:
19 //applying laws of thermodynamics
20 //applying formula for entropy change for
      irreversible process
21 //entropy change of hot water
22 delta_SH = m*c*log(Tf/TiH)//in J/K
23 //entropy change of cold water
24 delta_SC = m*c*log(Tf/TiC)//in J/K
25 //net entropy change of irreverse process
26 delta_S = delta_SH+delta_SC//in J/K
27 delta_SH  = round(delta_SH)
28 delta_SC  = round(delta_SC)
29 delta_S = round(delta_S)
30
31 printf ("\n\n Entropy change of hot water delta_SH =
       \n\n %3i J/K" ,delta_SH);
32 printf ("\n\n Entropy change of cold water delta_SC
     = \n\n %3i J/K" ,delta_SC);
33 printf ("\n\n Net entropy change of irreverse
     process delta_S = \n\n %3i J/K" ,delta_S);
```

Scilab code Exa 24.4 C24P4

```
1 clear
2 clc
3 //to find net entropy change of the gas for
     irreverse process
4
5 //Given:
6 //refer to figure 24-5(a)and (b) from page no. 550
7 //number of moles
8 n = 0.55//in mol
```

```
9 //room temperature
10 T = 293//in K
11 //molar gas constant
12 R = 8.31//in J/mol.K
13
14 //Solution:
15 //applying laws of thermodynamics
16 //applying formula for entropy change for isothermal
       expansion
17 //ratio of final to initial volumes //since both
      chamber are of same volumes
18 Vf_by_Vi = 2
19 //entropy change of the gas for irreverse process
20 delta_S = n*R*log(Vf_by_Vi)//in J/K
21
22 printf ("\n\n Ratio of final to initial volumes VF/
      Vi = \n\n %1i" ,Vf_by_Vi);
23 printf ("\n\n Entropy change of the gas for
      irreverse process delta_S = \n\n %.2f J/K" ,
      delta_S);
```

Scilab code Exa 24.5 C24P5

```
1 clear
2 clc
3 //to find maximum possible efficiency of turbine
4
5 //Given:
6 //temperature of steam in boiler
7 TH = 520//in degree celsius
8 ////temperature of steam in condenser
9 TL = 100//in degree celsius
10
11 //Solution:
12 //applying laws of thermodynamics
```

```
13 //applying formula for carnot cycle
14 //maximum possible efficiency of turbine
15 Emax = 1-((TL+273)/(TH+273))
16 Emax1 = Emax*100//in percent
17 Emax1 = round(Emax1)
18
19 printf ("\n\n Maximum possible efficiency of turbine
       Emax = \n\n %.2f" ,Emax);
20 printf ("\n\n Maximum possible efficiency of turbine
       Emax = \n\n %2i percent" ,Emax1);
```

Scilab code Exa 24.6 C24P6

```
1 clear
2 clc
3 //to find work per cycle required to operate
     refrigerator
4 //to find heat per cycle discharged to the room
5
6 //Given:
7 //coefficient of performance of refrigerator
8 K = 4.7
9 //rate of heat extraction
10 QL = 250//in J/cycle
11
12 //Solution:
13 //applying laws of thermodynamics
14 //applying formula for refrigeration cycle
15 //work per cycle required to operate refrigerator
16 W = QL/K//in J/cycle
17 //heat per cycle discharged to the room
18 QH = W+QL//in J/cycle
19
20 printf ("\n\n Work per cycle required to operate
     refrigerator W = \n\n %3i J/cycle" ,W);
```

```
21 printf ("\n\n Heat per cycle discharged to the room
     QH = \n\n %3i J/cycle" ,QH);
```

Scilab code Exa 24.7 C24P7

```
1  clear
2  clc
3  //to find minimum rate of energy to be supplied to
       the heat pump
4
5  //Given:
6  //outside temperature
7  TL = -10//in degree celsius
8  //interior temperature
9  TH = 22//in degree celsius
10 //heat transfer
11 QH = 16//in KW
12
13 //Solution:
14 //applying laws of thermodynamics
15 //applying formula for refrigeration cycle
16 //coefficient of performance
17 K = (TL+273)/((TH+273)-(TL+273))//taking temperature
       in K
18 //minimum rate of energy to be supplied to the heat
       pump
19 W_by_deltat = QH/(K+1)//in KW
20
21 printf ("\n\n Coefficient of performance K = \n\n %
     .2f" ,K);
22 printf ("\n\n Minimum rate of energy to be supplied
     to the heat pump W_by_deltat = \n\n %.1f KW" ,
     W_by_deltat);
```

Scilab code Exa 24.8 C24P8

```
clear
clc
//to find heat energy extracted from high
   temperature reservior per cycle
//to find heat energy discharge to low temperature
   reservior per cycle
//to find entropy change per cycle

//Given:
//work output
W = 120//in J per cycle
//efficiency
Ex = 75//in percent
//boiling point of water
TH = 100//in degree celsius
//freezing point of water
TL = 0//in degree celsius

//Solution:
//applying laws of thermodynamics
//applying formula for refrigeration cycle
//applying carnot cycle formula
//efficiency of carnot engine
Ec = 1-((273+TL)/(TH+273))//taking temperatures in K
Ec1 = Ec*100//in percent
//heat energy extracted from high temperature
   reservior per cycle
QH = W/(Ex*10^-2)//in J
//heat energy discharge to low temperature reservior
    per cycle
QL = QH-W//in J
delta_SH = -(QH)/(TH+273)//in J/K //taking
```

```
    temperatures in K
29 delta_SL = (QL)/(TL+273)//in J/K //taking
    temperatures in K
30 delta_SWS = 0//in J/K
31 //entropy change per cycle
32 delta_Sx = delta_SH+delta_SL+delta_SWS//in J/K
33 Ec1 = round(Ec1)
34
35 printf ("\n\n Efficiency of carnot engine Ec = \n\n
    %.3f" ,Ec);
36 printf ("\n\n Efficiency of carnot engine Ec = \n\n
    %2i percent" ,Ec1);
37 printf ("\n\n Heat energy extracted from high
    temperature reservior per cycle QH = \n\n %3i J"
    ,QH);
38 printf ("\n\n Heat energy discharge to low
    temperature reservior per cycle QL = \n\n %3i J"
    ,QL);
39 printf ("\n\n Entropy change per cycle delta_Sx = \n
    \n %.2f J/K" ,delta_Sx);
```

Scilab code Exa 24.9 C24P9

```
1 clear
2 clc
3 //to find number of independent ways
4 //to find number of microstates
5
6 //Given:
7 //number of molecules
8 N = 200//in molecules
9 //half number of molecules
10 N1 = 100//in molecules
11 //for 150 molecules in one box and 50 molecules in
    one box
```

```
12 n1 = 150
13 n2 = 50
14
15 //Solution:
16 //number of independent ways
17 w = factorial(N)/((factorial(N1))*(factorial(N1)))
18 //number of microstates
19 W = factorial(N)/((factorial(n1))*(factorial(n2)))
20 //answer is Nan.Because function factorial in scilab
       overflows as soon as N > 170 as here numerator
     is  N = 200 and answer of denominator is infinity
21
22 printf ("\n\n Number of independent ways w = \n\n %
     .2e" ,w);
23 printf ("\n\n Number of microstates W = \n\n %.2e" ,
     W);
```