Scilab Textbook Companion for Engineering Mechanics by S S Bhavikatti[1]

Created by
Sumit Shivaji Metakari
B.E. Mechanical
Mechanical Engineering
Government College Of Engineering Karad
College Teacher
None
Cross-Checked by
None

August 9, 2018

[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: Engineering Mechanics

Author: S S Bhavikatti

Publisher: New Age International (p) Limited,new Delhi

Edition: 5

Year: 2015

ISBN: 978-81-224-3798-0

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 2

Resultant and equilibrium of system of coplanar concurrent forces

Scilab code Exa 2.1 Determination of magnitude of forces and angle between them

```
//Determination of magnitude of forces and angle
   between them
//Assume F1=F then F2=2*F1
//condition 1 gives
//5*F^2+4*F^2*cosd(theta)=67600...(1)
//condition 2 gives
//5*F^2-4*F^2*cosd(theta)=32400...(2)
//Adding (1) and (2)
F=sqrt(10000)  //N
F1=F  //N
F2=2*F  //N
//put F1 and F2 in Equation (1)
theta=acosd(0.44)  //degree
printf("Magnitude of forces are :-\n F1=%.0f N\n F2=
   %.0f N",F1,F2)
```

```
printf("\nAngle between the forces is :-\n theta=%.1f degree",theta)
```

Scilab code Exa 2.2 Horizontal and Vertical components of Force

```
//Horizontal and Vertical components of Force
//Resolving 20 kN force we get
Fx=20*cosd(60)  //kN  (towards left)
Fy=20*sind(60)  //kN  (Downward)
printf("Horizontal and vertical components respectively are:-\n Fx=%.2f kN (towards left)\n Fy=%.2f kN (Downward)",Fx,Fy)
```

Scilab code Exa 2.3 Components of block normal to and parallel to inclined plane

```
//Components of block normal to and parallel to inclined plane
//Let Wn be the normal component and Wp be the parallel component
//Refer fig. 2.5(b),triangle ABC
Wn=10*cosd(20)  //kN
Wp=10*sind(20)  //kN
printf("The normal and parallel components respectively are :-\n Wn=%.2f kN\n Wp=%.2f kN",Wn,Wp)
```

Scilab code Exa 2.4 Resultant of three forces that are acting on a hook

```
//Resultant of three forces that are acting on a
   hook
//Resolving all forces along x and y axis gives
Fx=70*cosd(50)+80*cosd(25)+50*cosd(45)   //N
Fy=70*sind(50)+80*sind(25)-50*sind(45)   //N
R=sqrt(Fx^2+Fy^2)   //N
alpha=atand(Fy/Fx)   //degree
printf("\nThe resultant is R=%.1f N \nThe
   inclination of resultant w.r.t. positive x-axis
   is alpha=%.2f degree",R,alpha)
//The answers vary due to round off error
```

Scilab code Exa 2.5 Determining the resultant

```
//Determining the resultant
clc
//Given-
//inclination of 200N force with x axis calculated
   by using slope of 200N force as shown in fig.2.8
theta1=atand(1/2)    //degree
//inclination of 120N force with x axis calculated
   by using slope of 120N force as shown in fig.2.8
theta2=atand(4/3)    //degree
//summation of forces in X direction
Fx=200*cosd(26.565)-120*cosd(53.13)-50*cosd(60)+100*
   sind(40) //N
```

```
//summation of forces in Y direction
Fy=200*sind(26.565)+120*sind(53.13)-50*sind(60)-100*cosd(40) //N
//Resultant
R=sqrt((Fx)^2+(Fy)^2)    //N
//inclination of resultant w.r.t X axis
alpha=atand(65.5/146.2)    //degree
printf("The resultant of given forces :-\nR=%.1f N.\n",R)
printf("Inclination of resultant w.r.t X-axis :-\nalpha=%.1f degree.",alpha)
```

Scilab code Exa 2.6 Determine resultant force

```
//Determine Resultant Force
//From given data
T=1200   //N
F=100   //N
N=500   //N
W=1000   //N
theta=60   //degree
//Taking co-ordinate system parallel and perpendicular to plane as x and y axis and resolving the forces
Fx=T-F-W*sind(theta)   //N
Fy=N-W*cosd(theta)   //N
R=sqrt(Fx^2+Fy^2)
printf("The resultant has magnitude R=%.0f N directed up the plane",R)
```

Scilab code Exa 2.7 Finding the third force F

```
//Finding the third force F
//Assume that the third force F makes an angle theta
    with x-axis
//Resolving the forces we get
//F*cosd(theta)=-225.9...(1)  //N
//F*sind(theta)=-408.9...(2)  //N
//Then (2)/(1) gives
theta=atand(-408.9/-225.9)  //degree
F=sqrt(225.9^2+408.9^2)  //N
printf("The third force is F=%.1f N and makes an
   angle of theta=%.2f degree",F,theta)
```

Scilab code Exa 2.8 Determining the value of theta

```
//Determining the value of theta
//x and y axes are selected as shown in fig. 2.11
//As the resultant is directed along x-axis,
   component of resultant in y-direction is zero
//Fy=0 gives
theta=(asind(0.833/(2*cosd(20))))-20  //degree
printf("Required value of theta=%.2f degree",theta)
```

Scilab code Exa 2.9 Finding T and R

```
//Finding T and R
//applying Lami's Theorem we get
T=(100*sind(90))/sind(90+15)   //N
R=(100*sind(180-15))/sind(90+15)   //N
printf("\nThe required values are:-\nT=%.1f N \nR=%
   .1f N",T,R)
```

Scilab code Exa 2.10 Determining horizontal force F

```
//Determining horizontal force F
//From fig 2.14(b)
//Resolving the forces
//Fy=0 gives
R=1500/cosd(30)   //N
//Fx=0 gives
F=R*sind(30)   //N
printf("Horizontal force of F=%.0f N is required to
   be applied",F)
```

Scilab code Exa 2.11 Finding forces developed in wires

```
//Finding forces developed in wires
//applying Lami's theorem
T1=150*sind(90+60)/sind(45+30)   //N
T2=150*sind(180-45)/sind(45+30)   //N
printf("The forces in the wires are:-\nT1=%.1f N \
   nT2=%.1f N",T1,T2)
```

Scilab code Exa 2.12 Determine reactions at contact

```
//Determine reactions at contact
//Refer fig.2.16(b)
//applying Lami's Theorem
R1=400*sind(180-45)/sind(60+45)   //N
R2=400*sind(180-60)/sind(60+45)   //N
printf("The reactions developed are:-\nR1=%.1f N \
   nR2=%.1f N",R1,R2)
```

Scilab code Exa 2.13 To determine force in the bar and floor reaction

```
//To determine force in the bar and floor reaction
//Refer Fig. 2.17(b)
//Equilibrium equation gives
S=(7*cosd(45)-5)/cosd(30)   //kN
R=10+7*sind(45)-S*sind(30)   //kN
printf("Tensile force in the bar has magnitude %.3f
   kN and Reaction from floor is R=%.3f kN",-S,R)
```

Scilab code Exa 2.14 Finding magnitude of F

```
// Finding magnitude of F
```

```
//When F is applied at point B,refer fig 2.18(a)and(
    b)
//From triangle AOC
OC=300-150
AO=300
alpha=acosd(OC/AO)  //degree
//from triangle AOB using geometry we get angle OBA
    =30 degree
//Resolving the forces we get
R=2000/cosd(30)  //N
F=R*sind(30)  //N
printf("Least force through point B is F=%.1f N",F)
//Least force required through the centre of roller
//Assume that F makes an angle theta with the
    horizontal
//Refer fig. 2.19 (a) and (b)
//Resolving the forces we get
//F*cosd(theta)=R*sind(60)...(1)
//F*sind(theta)+R*cosd(60)=W...(2)
//Solving (1) and (2) we get
//sind(theta)+cotd(60)*cosd(theta)=W/F
//For obtaining maximum value of W/F we
    differentiate W/F w.r.t. theta and we get
theta=acotd(cotd(60))  //degree
//Least value of F is observed when it is at right
    angle to reaction R
Fmin=2000*sind(60)  //N
printf("\nLeast force through the centre of roller
    is Fmin=%.0f N",Fmin)
```

Scilab code Exa 2.15 Determining the forces in bars AB and AC

```
//Determining the forces in bars AB and AC
```

```
//Refer fig 2.20(a) and (b)
//Select AB and AC as x and y axes
//Resolving the forces we get
F1=0  //N
F2=40*cosd(30)  //N
printf("Force in bar AB is F1=%.0f N and force in bar AC is F2=%.1f N",F1,F2)
```

Scilab code Exa 2.16 Forces in various segments of cable

```
//Forces in various segments of cable
//Refer fig. 2.21 (a) and (b)
//Apply Lami's theorem at point D
T1=250*sind(180-60)/sind(60+45)  //N
T2=250*sind(90+45)/sind(60+45)  //N
//Now consider system of forces acting at B
//Resolving the forces we get
T3=(T2*cosd(60)+200)/cosd(30)  //N
T4=T3*sind(30)+T2*sind(60)  //N
printf("\nthe various forces are:-\nT1=%.1f N\nT2=%.1f N\nT3=%.1f N\nT4=%.1f N",T1,T2,T3,T4)
```

Scilab code Exa 2.17 Load required at point D

```
//Load required at point D
//Refer fig. 2.22 (a),(b) and (c)
//Using simple geometry we have
alpha=acosd(1.3125/1.5)  //degree
Beta=acosd(2-1.3125)  //degree
```

```
//Applying Lami's Theorem at point C
T1=1500*sind(90)/sind(180-alpha)  //N
T2=1500*sind(90+alpha)/sind(180-alpha)  //N
//Applying Lami's Theorem at point B
T3=T2*sind(90)/sind(90+Beta)  //N
W=T2*sind(180-Beta)/sind(90+Beta)  //N
printf("The load required to be connected at point D is W=%.1f N",W)
```

Scilab code Exa 2.18 finding tension and inclination

```
//finding tension and inclination
//refer fig. 2.23 (a),(b) and (c)
//Applying Lami's theorem at B
T1=20*sind(50)/sind(180+30-50)  //kN
T2=20*sind(180-30)/sind(180+30-50)  //kN
//now consider equilibrium of forces at point C we get
//T3*sind(theta)=22.4...(1)
//T3*cosd(theta)=11.20...(2)
//from (1) and (2) we get
theta=atand(2)  //degree
//then (1) gives
T3=T2*sind(50)/sind(theta)  //kN
printf("\nThe required values are:-\nT1=%.2f kN\nT2=%.2f kN\nT3=%.2f kN",T1,T2,T3)
```

Scilab code Exa 2.19 determine tension and inclination

```
//determine tension and inclination
//Refer fig. 2.24 (a),(b) and (c)
//consider equilibrium at point B,we get
//T2*sind(theta)=T1*sind(30)...(1)
//T2*cosd(theta)=T1*sind(30)-20...(2)
//consider equilibrium at point C,we get
//T2*sind(theta)=T3*sind(60)...(3)
//T2*cosd(theta)=-T3*cosd(60)+25...(4)
//solving (1) and (3) we get
//T1=T3*sqrt(3)...(5)
//solving (2) and (4) and substituting (5) we get
T3=45/2  //kN
T1=T3*sqrt(3)  //kN
//then (1)/(2) gives
theta=atand(1.416)  //degree
T2=19.48/sind(theta)  //kN
printf("\nThe required values are:-\nT1=%.2f kN\nT2=
   %.2f kN\nT3=%.2f kN\ntheta=%.2f degree",T1,T2,T3,
   theta)
```

Scilab code Exa 2.20 Reactions developed at contacts

```
//Reactions developed at contacts
//Refer fig. 2.25(a),(b and (c)
//consider equilibrium of cylinder 1
//using conditions of equilibrium we get
RA=500*cosd(30)  //N
RB=500*sind(30)  //N
//Consider equilibrium of cylinder 2
//using conditions of equilibrium we get
RC=(500+250*sind(30))/cosd(30)  //N
RD=RC*sind(30)+250*cosd(30)  //N
printf("\nThe reactions are:-\nRA=%.1f N\nRB=%.1f N\
```

```
   nRC=%.1f N\nRD=%.1f N",RA,RB,RC,RD)
```

Scilab code Exa 2.21 reactions developed at contact surfaces

```
//reactions developed at contact surfaces
//Refer fig. 2.26 (a),(b) and (c)
//using geometry
theta=acosd(0.8)  //degree
//consider equilibrium of cylinder 1
//Using equilibrium conditions
RB=800/sind(theta)  //N
RA=RB*cosd(theta)  //N
//consider equilibrium of cylinder 2
//Using equilibrium conditions
RD=((RB*sind(theta))+1200)/cosd(45)  //N
RC=RD*sind(45)+RB*cosd(theta)  //N
printf("\nThe reactions are:-\nRA=%.1f N\nRB=%.1f N\
   nRC=%.1f N\nRD=%.1f N",RA,RB,RC,RD)
```

Scilab code Exa 2.22 determine the reactions developed at contact points

```
//determine the reactions developed at contact
   points
//refer fig. 2.27 (a),(b) and (c)
//considering the equilibrium conditions of
   cylinders we have
RB=600  //N
alpha=atand(450/150)  //degree
RD=RB/sind(alpha)  //N
```

```
RC=RD*cosd(alpha)   //N
RA=RC   //N
printf("\nThe reactions are:-\nRA=%.1f N\nRB=%.1f N\nRC=%.1f N\nRD=%.1f N",RA,RB,RC,RD)
//The answers vary due to round off error
```

Scilab code Exa 2.23 Force P required to hold the system

```
//Force P required to hold the system
//Refer to fig. 2.28 (a),(b)&(c)
//Applying Lami's Theorem at A we get
C=4000*sind(180-60)/sind(60+90-15)   //N
//Applying equilibrium conditions B
P=(-2000*cosd(45)+C*cosd(60))/cosd(15)   //N
printf("P=%.1f N is required to hold the system in given position.",P)
```

Chapter 3

Resultant and equilibrium of system of coplanar non concurrent forces

Scilab code Exa 3.1 `Determine the moment`

```
//Determine the moment
//Refer fig. 3.5
//Take clockwise moment as positive
//Apply Varignon's Theorem
MA=100*300*cosd(60)-100*500*sind(60)   //N-mm
printf("MA=%f N-mm, Anticlockwise",-MA)
```

Scilab code Exa 3.2 `Finding y intercept`

```
//Finding y-intercept
```

```
//Apply law of Transmissibility and resolve 5000 N force at B
Fx=5000*4/5   //N
Fy=5000*3/5   //N
//Apply Varignon's Theorem
y=8000/4000   //m
printf("y-intercept will be y=%.0d m",y)
```

Scilab code Exa 3.3 Determine Resultant

```
//Determine Resultant
//horizontal direction is assumed as x-axis and vertical as y-axis
Rx=-20*cosd(60)   //kN  (towards left)
Ry=-20-30-20*sind(60)   //kN  (downwards)
R=sqrt(Rx^2+Ry^2)   //kN
alpha=atand(Ry/Rx)   //degree  (as shown in fig. 3.12(b))
//Taking moment about A
MA=20*1.5+30*3+20*6*sind(60)   //kN-m
//x-intercept of the resultant is
x=MA/Ry   //m (shown in fig.)
printf("R=%.2f kN as shown in fig. 3.12(a)",R)
```

Scilab code Exa 3.4 Find resultant

```
//Find resultant
Rx=60-100*cosd(60)-120*cosd(30)   //kN  (towards left)
```

```
Ry=-80+100*sind(60)-120*sind(30)  //kN  (downwards)
R=sqrt(Rx^2+Ry^2)  //kN
alpha=atand(Ry/Rx)  //degree  (shown in fig. 3.13(b))
MA=(80*100*cosd(60)+60*100*sind(60)+120*100*sind(30))  //kN-mm
//intercept on x-axis is
x=MA/Ry  //mm  (as shown in fig. 3.13(a))
printf("R=%.0f kN is the resultant as shown in fig. 3.13 (a)",R)
```

Scilab code Exa 3.5 Find the resultant

```
//Find the resultant
//refer fig. 3.14 (a) and (b)
theta1=atand(10/10)
theta2=atand(30/40)
theta3=atand(10/20)
Rx=2*cosd(theta1)+5*cosd(theta2)-1.5*cosd(theta3)  //kN
Ry=2*sind(theta1)-5*sind(theta2)-1.5*sind(theta3)  //kN
R=sqrt(Rx^2+Ry^2)  //kN
alpha=atand(-Ry/Rx)  //Degree
//Moment of forces about O is
MO=2*30*cosd(45)+5*50*sind(theta2)+1.5*10*sind(theta3)  //kN-mm
//distance d of resultant R from O is given as
d=MO/R
printf("The resultant of the system is R=%.3f kN as shown in fig. 3.14(b)",R)
```

Scilab code Exa 3.6 Determine magnitude along with direction and point of applicat

```
//Determine magnitude,direction and point of application
//refer fig.3.15(a)&(b)
Rx=500*cosd(60)-700  //N (towards left)
Ry=-500*sind(60)-1000-1200  //N (Downwards)
R=sqrt((Rx^2)+(Ry^2))  //N
alpha=atand(-Ry/Rx)  //degree
//taking moment about O
MO=-500*300*sind(60)-1000*150+1200*150*cosd(60)-700*300*sind(60)
//let point of application of resultant be at a distance of x from point O along the horizontal then
x=MO/Ry  //mm
printf("The Resultant is R=%.1f N as shown in fig.3.15",R)
```

Scilab code Exa 3.7 Safety of dam

```
//Safety of dam
//refer fig. 3.16
Rx=300  //kN  (towards right)
Ry=100-1200-400  //kN  (Downwards)
//taking moment about O
MO=300*3-100*1+1200*2+400*5
```

```
//assume that the resultant cut the base at a
   distance of x from O
x=MO/Ry  //m
printf("x=%.3f m lies in the middle third of base.
   Hence dam is safe",-x)
```

Scilab code Exa 3.8 determine the resultant

```
//determine the resultant
//refer fig. 3.17 (a) and (b)
Rx=-400*cosd(45)-150*cosd(30)  //N  (towards left)
Ry=200+400*sind(45)-150*sind(30)
R=sqrt(Rx^2+Ry^2)  //N
alpha=atand(Ry/Rx)  //degree
//assume that the resultant intersects arm AB at a
   distance of x from A
//taking moment about A
MA=-400*3*sind(45)-400*0.6*cosd(45)+50+150*6*sind
   (30)+150*1*cosd(30)  //N-m  (anticlockwise)
x=MA/Ry  //m
printf("The resultant is R=%.1f N as shown in fig
   .3.17 (a)",R)
```

Scilab code Exa 3.9 determine equilibriant

```
//determine equilibriant
//two 40 kN forces have no moment about the pulley
   centre hence can be considered acting at pulley
   centre
```

```
//Accordingly
Rx=20*cosd(45)-30*cosd(60)-50*cosd(30)+40*cosd(20)-40*sind(30)  //kN  (towards left)
Ry=-20*sind(45)-20+20-30*sind(60)-50*sind(30)-40*sind(20)-40*cosd(30)  //kN  (Downwards)
R=sqrt(Rx^2+Ry^2)  //kN
alpha=atand(Ry/Rx)  //degree
//Taking moment about A
MA=20*4-20*4+30*6*sind(60)+50*2*sind(30)-50*2*cosd(30)+40*3*cosd(20)-40*3*sind(30)
//assume that the resultant intersects AB at a distance x from A,then
x=MA/Ry  //m
printf("Equilibriant is equal and opposite to resultant.\nR=%.2f kN\nalpha=%.2f degree\nx=%.3f m\nAs shown in fig.3.18 (a)",R,alpha,-x)
```

Scilab code Exa 3.10 tension in the cable and reaction at a point

```
//tension in the cable and reaction at a point
//refer fig. 3.21 (a),(b)&(c)
//Taking moment about A we get
T=((25*12*cosd(30))+(10*6*cosd(30)))/(12*sind(15))  //kN
//applying equilibrium conditions
HA=T*cosd(15)  //kN
VA=10+25+T*sind(15)  //kN
RA=sqrt(HA^2+VA^2)  //kN
alpha=atand(VA/HA)  //degree
printf("tension T in cable BC is T=%.2f kN.\nRA=%.3f kN\nalpha=%.2f degree",T,RA,alpha)
```

Scilab code Exa 3.11 Horizontal force required at bottom to avoid slipping

```
//Horizontal force required at bottom to avoid
   slipping
//refer fig 3.22 (a)&(b)
//Taking moment about A
RB=(700*2*cotd(60)+100*1.5*cotd(60))/3  //N
F=RB  //N
printf("The required force is F=%.1f N",F)
```

Scilab code Exa 3.12 Horizontal force required at certain height to avoid slipping

```
//Horizontal force required at certain height to
   avoid slipping
//Refer fig. 3.23
//applying equilibrium conditions we get
//F=RB...(1)
//-RB*3+700*2*cotd(60)+100*1.5*cotd(60)+F=0...(2)
//Solving (1)&(2) we get
F=(700*2+100*1.5)*cotd(60)/2  //N
printf("Required force is F=%.2f N",F)
```

Scilab code Exa 3.13 Reactions at supports C and D

```
//Reactions at supports C and D
//Refer fig.3.24(a),(b)&(c)
//applying equilibrium conditions for roller
R2=2000/cosd(30)  //N
//consider equilibrium of bar CD
//Taking moment about C
RD=((800*2.5*cosd(30))+(R2*2))/(5*cosd(30))  //N
VC=800+R2*cosd(30)-RD  //N
HC=R2*sind(30)  //N
printf("The reactions are:-\nRD=%.1f N\nHC=%.1f N",RD,HC)
```

Scilab code Exa 3.14 Tension in the cable and reaction at axles

```
//Tension in the cable and reaction at axles
//Refer fig.3.25 (a)&(b)
//assume T as tension in the rope parallel to track
//applying equilibrium conditions
T=60*sind(60)  //kN
//applying moment equilibrium condition about upper axle reaction point we get
R1=(-T*600+60*800*sind(60)+60*600*cosd(60))/1200  //kN
R2=60*cosd(60)-R1  //kN
printf("Required values are:-\nT=%.3f kN\nR1=%.3f kN\nR2=%.2f kN",T,R1,R2)
```

Scilab code Exa 3.15 Minimum weight of hollow cylinder

```
//Minimum weight of hollow cylinder
//Refer fig.3.26 (a),(b)&(c)
O1O2=400+600  //mm
O2D=1600-400-600  //mm
alpha=acosd(O2D/O1O2)  //degree
//considering equilibrium of spheres
//Taking moment about O2
R1=(600)/(1000*sind(alpha))  //kN
R2=R1  //kN
R3=1+3  //kN
//Assume minimum weight W.During tipping there will be no reaction at point B
//Taking moment about A
W=(0.75*1000*sind(53.13))/(800)  //kN
printf("Minimum weight of hollow cylinder is W=%.2f kN",W)
```

Scilab code Exa 3.16 `Tension in horizontal rope`

```
//Tension in horizontal rope
//Refer fig. 3.27 (a),(b)&(c)
//Considering equilibrium of the entire system
RB=500/2  //N
RA=RB  //N  (symmetry)
R1=500/(2*cosd(60))  //N
R2=R1  //N
//Taking moment about C
T=((500*0.866)+(250*1.2*0.5))/(1.8*sind(60))  //N
printf("The tension is:-\nT=%.1f N",T)
```

Scilab code Exa 3.17 Reactions developed in cantilever beam

```
//Reactions developed in cantilever beam
//Refer fig. 3.44 (a)&(b)
//assumptions are made as shown in fig. 3.44 (a)&(b)
//applying equilibrium conditions
VA=15+(10*2)+(20*sind(60))  //kN
HA=20*cosd(60)  //kN
//Taking moment about A
MA=10*2*1+20*2*sind(60)+15*3  //kN-m
printf("Required values:-\nVA=%.2f kN\nHA=%.2f kN\
   nMA=%.2f kN-m",VA,HA,MA)
```

Scilab code Exa 3.18 Reactions developed in cantilever beam

```
//Reactions developed in cantilever beam
//Refer fig. 3.45 (a)&(b)
//Make assumptions as shown in fig. 3.45(a) and(b)
//applying equilibrium conditions
VA=60+45*2/2  //kN
HA=0  //kN
//Taking moment about A
MA=((45*2*2)/(3*2))+(60*2.5)  //kN-m
printf("Required values:-\nVA=%.2f kN\nHA=%.2f kN\
   nMA=%.2f kN-m",VA,HA,MA)
```

Scilab code Exa 3.19 Reactions developed in simply supported beam

```
//Reactions developed in simply supported beam
//Refer fig. 3.46 (a)&(b)
//make assumptions as shown in fig. 3.46 (a)&(b)
//Taking moment about B
RA=((20*4*2)+((4*40*4)/(3*2)))/(6) //kN
RB=80+80-RA  //kN
printf("The reactions are:-\nRA=%.2f kN\nRB=%.2f kN"
   ,RA,RB)
```

Scilab code Exa 3.20 Reactions developed at A and B

```
//Reactions developed at A and B
//Refer fig. 3.47 (a) and (b)
//Make proper assumptions from this fig.
//applying equilibrium conditions
HA=15*cosd(30)+20*cosd(45)  //kN
//Taking moment about A
RB=(10*4+15*6*sind(30)+20*10*sind(45))/12  //kN
VA=-RB+10+15*sind(30)+20*sind(45)  //kN
printf("The reactions developed are:-\nHA=%.2f kN\
   nVA=%.2f kN\nRB=%.2f kN",HA,VA,RB)
```

Scilab code Exa 3.21 determine magnitude and direction of support reactions

```
//determine magnitude and direction  of support
   reactions
//Refer fig. 3.48 (a)&(b)
//Taking moment about A
RB=((60*1*sind(60))+(80*3*sind(75))+(50*5.5*sind(60)
   ))/(6*sind(60))  //kN  (At 60 degree to the
   horizontal)
HA=(-60*cosd(60))+(80*cosd(75))-(50*cosd(60))
   +(100.45*cosd(60))  //kN
VA=(-100.45*sind(60))+(60*sind(60))+(80*sind(75))
   +(50*sind(60))  //kN
RA=sqrt((HA^2)+(VA^2))  //kN
alphaA=atand(VA/HA)  //Degree
printf("The reactions are:-\nRA=%.2f kN \nRB=%.2f kN
    \nAs shown in fig. 3.48",RA,RB)
```

Scilab code Exa 3.22 determine support reactions

```
//determine support reactions
//Refer fig. 3.49
//Taking moment about B
RA=(20*7+30*4*5+60*2*sind(45))/9  //kN
HB=60*cosd(45)  //kN
VB=20+30*4+60*sind(45)-RA  //kN
printf("The reactions are:-\nRA=%.2f kN \nHB=%.2f kN
    \nVB=%.2f kN \nAs shown in fig.3.49",RA,HB,VB)
```

Scilab code Exa 3.23 determine support reactions

```
//determine support reactions
//Refer fig. 3.50
//Taking moment about A
RB=(30*1+24*3*(2+1.5)+(1.5*40/2)*(5+1.5/3))/5  //kN
RA=30+72+30-RB  //kN
printf("The support reactions are:-\nRA=%.2f kN \nRB
   =%.2f kN \nAs shown in fig. 3.50",RA,RB)
```

Scilab code Exa 3.24 determine support reactions

```
//determine support reactions
//Refer fig. 3.51
//Taking moment about A
RB=(40+30*5*sind(45)+20*2*7)/6  //kN
HA=30*cosd(45)  //kN
VA=30*sind(45)-RB+40  //kN (downwards)
printf("Required values are:-\nRB=%.2f kN\nHA=%.2f
   kN\nVA=%.2f kN downwards",RB,HA,-VA)
```

Scilab code Exa 3.25 determine support reactions

```
//determine support reactions
//Refer fig. 3.52
//Taking moment about B
RA=((10/2)*(1/3+5)+(2*10/2)*(5-2/3)
   +10*3*1.5+3*10*1/2)/5  //kN
RB=-26+5+10+30+15  //kN
```

```
printf("Required values are:-\nRA=%.2f kN\nRB=%.2f
   kN",RA,RB)
```

Scilab code Exa 3.26 determine distance of support C from end A

```
//determine distance of support C from end A
//Refer fig. 3.53
//assume that the support at C is at a distance of x
    metres from end A
//applying equilibrium conditions
RC=(30+120+50)/2  //kN
RD=RC  //kN  (Given)
//Taking moment about A
x=(1000+1200-100*12)/200  //m
printf("The required distance is x=%.0f m",x)
```

Scilab code Exa 3.28 determine reactions at A along with B and D

```
//determine reactions at A,B and D
//Refer fig. 3.55
//Taking moment about C
RD=(3*5*2.5+(5*9*2*5)/(2*3))/7  //kN
RC=15+22.5-RD  //kN
//Taking moment about A
RB=(2*RC/5)  //kN
RA=RC-RB  //kN
printf("Required reactions are:-\nRA=%.2f kN\nRB=%.2
   f kN,\nRD=%.2f kN",RA,RB,RD)
```

Scilab code Exa 3.29 determine reactions at A along with C and D

```
//determine reactions at A,C and D
//Refer fig. 3.56
//Taking moment about A
RE=(20*3+40*4*sind(45))/3  //kN
HA=40*cosd(45)  //kN
VA=20+40*sind(45)-RE  //kN  (Downwards)
//Taking moment about C
RD=((20*sind(60))-(10)+(57.71*2))/3  //kN
HC=20*cosd(60)  //kN
VC=20*sind(60)+RE-RD  //kN
printf("Required Values are:-\nHA=%.2f kN\nVA=%.2f
   kN downward\nRD=%.2f kN\nHC=%.2f kN\nVC=%.2f kN",
   HA,-VA,RD,HC,VC)
```

Chapter 4

Analysis of pin jointed plane frames

Scilab code Exa 4.1 `finding forces in members of truss`

```
//finding forces in members of truss
//Refer fig. 4.8
//Step 1
theta=atand(3/3)  //Degree
//Step 2
//Assume Notations as in fig. 4.8
//Step 3
//applying equilibrium conditions
FCB=40/sind(45)  //kN
FCD=FCB*cosd(45)  //kN
//Step 4
//Mark and analyse at joint C
//Step 5
//Analyse joint D
FDB=40  //kN
FDE=40  //kN
//step 6
```

```
//Analysis of joint B
FBE=(40+56.57*sind(45))/sind(45)  //kN
FBA=FBE*cosd(45)+56.57*cosd(45)  //kN
//step 7
//Tabulating answers
printf("The required forces in members are:-\nAB=%.2
   f kN  (Tension)\nBC=%.2f kN  (Tension)\nCD=%.2f
   kN  (Compression)\nDE=%.2f kN  (Compression)\nBE=
   %.2f kN  (Compression)\nBD=%.2f kN  (Tension)",
   FBA,FCB,FCD,FDE,FBE,FDB)
```

Scilab code Exa 4.2 finding forces in members of truss

```
//finding forces in members of truss
//Refer fig. 4.9
//Find support reactions
//applying equilibrium conditions
//Taking moment about A
RD=(40*1+60*2+50*3)/4  //kN
RA=150-RD  //kN
//Consider equilibrium of joint A
FAB=RA/sind(60)  //kN  (Compression)
FAE=FAB*cosd(60)  //kN  (Tension)
//Joint D
FDC=RD/sind(60)  //kN  (Compression)
FDE=FDC*cosd(60)  //kN  (Tension)
//Joint B (Refer Fig. 4.9 (d)
FBE=((FAB*sind(60))-40)/sind(60)  //kN  (Tension)
FBC=FAB*cosd(60)+FBE*cosd(60)  //kN  (Compression)
//Joint C (Refer fig. 4.9 (e))
FCE=(FDC*sind(60)-50)/sind(60)  //kN  (Tension)
//Refer fig. 4.9 (e),(f)
printf("RD=%.2f kN\nRA=%.2f kN\nFAB=%.2f kN  (
```

```
Compression)\nFAE=%.2f kN (Tension)\nFDC=%.2f kN
  (Compression)\nFDE=%.2f kN (Tension)\nFBE=%.2f
 kN (Tension)\nFBC=%.2f kN (Compression)\nFCE=%
.2f kN (Tension)",RD,RA,FAB,FAE,FDC,FDE,FBE,FBC,
FCE)
```

Scilab code Exa 4.3 `Analysing the truss`

```
//Analysing the truss
//Refer fig. 4.10(a)
//inclined members make an angle theta with the
   horizontal
theta=atand(4/3) //Degree
//Joint E
//Refer fig. 4.10 (c)
//applying equilibrium conditions
FED=20/sind(theta) //kN (Tension)
FEF=25*cosd(theta) //kN (Compression)
//Refer fig 4.10 (b)
//Taking moment about A
RC=20*6/8 //kN
VA=20 //kN
HA=RC //kN
//Joint A
//Refer fig.4.10 (d)
//applying equilibrium conditions
FAB=VA //kN (Compression)
FAF=HA //kN (Compression)
//Joint C
//Refer fig. 4.10 (E)
FCB=RC/cosd(theta) //kN (Compression)
FCD=FCB*sind(theta) //kN (Tension)
//Joint B
```

```
//Refer fig. 4.10 (f)
FBF=(FBC*sind(theta)-FAB)/sind(theta)  //kN
FBD=0+25*cosd(theta)  //kN  (Tension)
//Joint F
//Refer Fig. 4.10(g)
FFD=0
FBF=0
printf("FED=%.2f kN (Tension)\nFEF=%.2f kN (Compression)\nRC=%.2f kN\nVA=%.2f kN\nHA=%.2f kN\nFAB=%.2f kN (Compression)\nFAF=%.2f kN (Compression)\nFCB=%.2f kN (Compression)\nFCD=%.2f kN (Tension)\nFBF=%.2f kN\nFBD=%.2f kN (Tension)\nFFD=%.2f kN\nFBF=%.2f kN",FED,FEF,RC,VA,HA,FAB,FAF,FCB,FCD,FBF,FBD,FFD,FBF)
```

Scilab code Exa 4.4 Finding forces in all members

```
//Finding forces in all members
//Refer fig. 4.11(a)
theta1=atand(4/6)  //Degree
theta2=atand(8/6)  //Degree
theta3=atand(4/3)  //Degree
//Joint H
FHG=20/sind(53.13)  //kN  (Compression)
FHF=25*cosd(53.13)  //kN  (Tension)
//Taking moment about A
RG=(20*9+12*6)/6  //kN
VA=32-42  //kN  (Downwards)
HA=0
//Joint A
//applying equilibrium conditions
FAC=10/sind(33.69)  //kN  (Compression)
FAB=18.03*cosd(33.69)  //kN  (Tension)
```

```
//Joint B
FBC=0
FCE=FAC  //kN  (Compression)
//Joint D
FDE=0
FDF=FBD  //kN  (Tension)
//Joint E
FEF=0
FEG=FCE  //kN  (Compression)
//Joint F
FAG=12  //kN  (Compression)
printf("Required values are:-\nFHG=%.2f kN  (Compression)\nFHF=%.2f kN  (Tension)\nRG=%.2f kN\nVA=%.2f kN  (Downwards)\nHA=%.2f kN\nFAC=%.2f kN  (Compression)\nFAB=%.2f kN  (Tension)\nFBC=%.2f kN\nFCE=%.2f kN  (Compression)\nFDE=%.2f kN\nFDF=%.2f kN  (Tension)\nFEF=%.2f kN\nFEG=%.2f kN  (Compression)\nFAG=%.2f kN  (Compression)",FHG,FHF,RG,-VA,HA,FAC,FAB,FBC,FCE,FDE,FDF,FEF,FEG,FAG)
```

Scilab code Exa 4.5 Analyse truss

```
//Analyse truss
//Refer fig. 4.12 (a)
//All triangles are equilateral
//applying equilibrium conditions At
//Joint G
FGF=20/sind(60)  //kN  (Tension)
FGE=FGF*cosd(60)  //kN  (Compression)
//Joint F
FFE=FGF  //kN  (Compression)
FFD=FGF*cosd(60)+FFE*cosd(60)-10  //kN  (Tension)
//Consider eqiulibrium of entire truss
```

```
RE=(-10*3*sind(60)+40*3*cosd(60)+30*(3+3*cosd(60))+20*9)/6   //kN
VA=(40+30+20)-58.17   //kN
HA=10   //kN
//Joint A
FAB=31.83/sind(60)   //kN  (Compression)
FAC=36.75*cosd(60)-10   //kN  (Tension)
//Joint B
FBC=(40-FAB*sind(60))/sind(60)   //kN  (Compression)
FBD=36.75*cosd(60)-9.44*cosd(60)   //kN  (Compression)
//Joint C
FCD=FBC   //kN  (Tension)
FCE=9.44*cosd(60)+9.44*cosd(60)-8.38   //kN  (Compression)
//Joint D
FDE=(30+FCD*sind(60))/sind(60)   //kN  (Compression)
printf("\nRequired Forces are:-\nFGF=%.2f kN  (Tension)\nFGE=%.2f kN  (Compression)\nFFE=%.2f kN  (Compression)\nFFD=%.2f kN  (Tension)\nRE=%.2f kN\nVA=%.2f kN\nHA=%.2f kN\nFAB=%.2f kN  (Compression)\nFAC=%.2f  (Tension)\nFBC=%.2f kN  (Compression)\nFBD=%.2f kN  (Compression)\nFCD=%.2f kN  (Tension)\nFCE=%.2f kN  (Compression)\nFDE=%.2f kN  (Compression)",FGF,FGE,FFE,FFD,RE,VA,HA,FAB,FAC,FBC,FBD,FCD,FCE,FDE)
```

Scilab code Exa 4.6 Determine The Forces in the member

```
//Determine The Forces in the member
//Using symmetry
RA=70/2   //kN
RB=RA   //kN
```

```
//Consider fig.4.13 (b)
//Taking moment about G
FFH=(35*12-10*10-10*6-10*2)/(4*sind(60))  //kN (
    Compression)
FGH=(35-10-10-10)/sind(60)  //kN (Compression)
FGI=69.28+5.77*cosd(60)  //kN (Tension)
printf("The required values are:-\nFFH=%.2d kN (
    Compressive)\nFGH=%.2d kN (Compressive)\nFGI=%.2
    d kN (Tension)",FFH,FGH,FGI)
```

Scilab code Exa 4.7 `finding magnitude and nature of forces`

```
//finding magnitude and nature of forces
//refer fig. 4.14(a)
//considering equilibrium if entire truss
//taking moment about L0
R2=(200*6+200*12+150*18+100*24+100*30)/36  //kN
R1=200+200+150+100+100-R2  //kN
//consider equilibrium of right hand side of section
     (1)-(1)
theta1=atand(1/6)  //degree
theta2=atand(6/8)  //degree
//taking moment about U4
FL3L4=(-100*6+325*12)/8  //kN (tension)
//taking moment about L3
FU3U4=456.2  //kN (compression)
FU4L3=(456.2*cosd(9.46)-412.5)/sind(36.87)  //kN (
    tension)
printf("The required forces are:-\nMember     Force\
    nU3U4=    %.2d kN (Compression)\nL3L4=    %.2d kN
     (Tension)\nU4L3=    %.2d kN (Tension)",FU3U4,
    FL3L4,FU4L3)
```

Scilab code Exa 4.8 Finding forces in members

```
//Finding forces in members
//Refer fig. 4.15 (a)
RA=7*20/2  //kN
RB=RA  //kN  (symmetry)
//CE is perpendicular on AB
CE=5.196  //m
DE=3  //m
theta=atand(5.196/3)  //degree
//The fact that 20 kN loads are equidistant can be
   used to find out horizontal distances of loads
   from A
//Consider equilibrium of left hand side portion of
   section (1)-(1)
//taking moment about A
F2=(20*2.25+20*4.5+20*6.75)/(6*sind(60))  //kN  (
   Tension)
F1=(70-20-20-20+51.96*sind(60))/sind(30)  //kN  (
   compression)
F3=-51.96*cosd(60)+110*cosd(30)  //kN  (Tension)
printf("The required forces are:-\nF1=%.2d kN (
   Compression)\nF2=%.2d kN (Tension)\nF3=%.2d kN (
   Tension)",F1,F2,F3)
```

Scilab code Exa 4.9 Finding required forces

```
//Finding required forces
```

```
//Refer fig. 4.16 (a)
//Symmetry gives
RE=(15+30+30+30+15)/2  //kN
RA=RE  //kN
FAE=30  //kN
//After construction as shown in ref. fig
//Taking moment about C
FAE=(60*5-15*5-30*2.5)/5  //kN  (Tension)
//assumptions are made as shown in fig. 4.16 (b)
//Apply equilibrium conditions and solving equations
FFC=15/0.366  //kN  (Tension)
FBC=(0.866*40.98+15)/0.707  //kN  (Compression)
//Lets analyse Joint B
//Applying equilibrium conditions
FBF=30*cosd(45)  //kN  (Compression)
FAB=71.41+21.21  //kN  (Compression)
//Lets analyse Joint A
//Applying equilibrium conditions
FAF=(92.62*sind(45)-45)/sind(30)  //kN  (Tension)
printf("\nThe required forces are:-\nFAB=%.2d kN (
    Compression)\nFBC=%.2d kN (Compression)\nFBF=%.2d
     kN (Compression)\nFAF=%.2d kN (Tension)\nFFC=%.2
    d kN (Tension)\nFAE=%.2d kN (Tension)",FAB,FBC,
    FBF,FAF,FFC,FAE)
```

Scilab code Exa 4.10 Finding unknown forces

```
//Finding unknown forces
//Refer fig. 4.8
//Let us assume joint E as origin,EC as x-axis,EA as
     y-direction
//accordingly the co-ordinates are
//A(0,3),B(3,3),C(6,0),D(3,0),E(0,0)
```

```
YD=-40  //kN
YC=-40  //kN
//Using co−ordinates lengths are found out to be
LAB=3  //m
LBC=3*sqrt(2)  //m
LCD=3  //m
LDE=3  //m
LBD=3  //m
LBE=3*sqrt(2)  //m
//Consider joint C
//applying equilibrium conditions
tCB=40/3
tCD=-40/3
FCB=tCB*LBC  //kN
FCD=-13.333*LCD  //kN
//Consider joint D
//applying equilibrium conditions
tDE=tCD
FDE=tCD*LCD  //kN
tDB=40/3
FDB=tDB*LBD
//Consider joint B
//applying equilibrium conditions
tBE=-(13.333+13.333)
FBE=tBE*LBE  //kN
tBA=40
FBA=tBA*LAB
printf("The required forces are:−\nForce in member
    AB=%.2d kN\nForce in member BC=%.2d kN\nForce in
    member CD=%.2d kN\nForce in member DE=%.2d kN\
    nForce in member EB=%.2d kN\nForce in member BD=%
    .2d kN",FBA,FCB,FCD,FDE,FBE,FDB)
```

Scilab code Exa 4.11 Analyse the truss by method of tension coefficient to determi

```
//Analyse the truss by method of tension coefficient
    to determine the forces
//Refer fig.4.19
//Consider entire structure
//Taking moment about A
YD=(40*2+50*6+30*4*sind(60)+60*4)/8
XA=-30  //kN
YA=40+50+60-90.49  //kN
//Take A as origin and determine co-ordinates of all
    other point
//Consider equilibrium of individual joints
//Joint A
tAB=-(59.51/3.464)
FAB=tAB*4  //kN
tAE=64.36/4
FAE=tAE*4  //kN
//Joint B
tBE=-11.547+17.18
FBE=tBE*4  //kN
tBC=0.5*(-17.18-5.637)
FBC=tBC*4  //kN
//Joint C
tCD=-(14.434+37.818)/2
FCD=4*tCD  //kN
tCD=4  //kN
tCE=-14.434-tCD
FCE=11.692*4  //kN
//Joint D
tDE=-0.5*(-26.126)
FDE=tDE*4  //kN
printf("The forces in different members are:-\nAB=%
    .2d kN\nBC=%.2d kN\nCD=%.2d kN\nDE=%.2d kN\nEA=%
    .2d kN\nEB=%.2d kN\nEC=%.2d kN",FAB,FBC,FCD,FDE,
    FAE,FBE,FCE)
```

Scilab code Exa 4.12 Analyse the truss

```
//Analyse the truss
//Refer fig. 4.20
//Consider equilibrium of entire truss
//taking moment about A
YE=(60*8-40*4)/4   //kN
XA=40  //kN
YA=60-80  //kN
//Take A as origin and determine co-ordinates of
   various point
//Lengths in m are
AB=4
CE=4
AE=4
ED=4
BE=4*sqrt(2)
CD=4*sqrt(2)
//Consider equilibrium of joints individually
//Joint A
tAB=5
FAB=tAB*AB  //kN
tAE=-10
FAE=tAE*AE  //kN
//Joint B
tBE=-5
FBE=tBE*BE  //kN
tBC=10-tBE
BC=4
FBC=tBC*BC  //kN
//Joint C
tCD=15
```

```
CD=4*sqrt(2)
FCD=15*4*sqrt(2)  //kN  The answer provided in the
    textbook is wrong
tCE=-15
FCE=tCE*CE  //kN
//Joint D
tDE=-15
DE=4
FDE=tDE*DE  //kN
printf("The forces in different members are:-\nAB=%
    .2d kN\nBC=%.2d kN\nCD=%.2d kN\nDE=%.2d kN\nEA=%
    .2d kN\nEC=%.2d kN\nEB=%.2d kN",FAB,FBC,FCD,FDE,
    FAE,FCE,FBE)
```

Chapter 5

Friction

Scilab code Exa 5.1 Value of P

```
   //Value of P
//Refer fig. 5.5 (a),(b)&(c)
//(a) when P is Horizontal Phor
//Consider equilibrium
//block A
N1=1000  //N
F1=0.25*N1  //N
T=F1  //N
//Block B
N2=N1+2000  //N
F2=3000/3  //mu*N2  N
Phor=F1+F2  //N
//(b) when P is inclined  (Pinc)
//Considering equilibrium of block B
//Using law of friction
//Pinc*cosd(30)-F1-F2=0
Pinc=1250/(cosd(30)+(0.5/3))  //N
printf("\nPhor=%0.2d N\nPinc=%0.2d N",Phor,Pinc)
```

Scilab code Exa 5.2 `Value of theta`

```
//Value of theta
//refer fig.5.6
//Consider equilibrium
//300N block
//N1=300*cosd(theta)
//Law of friction
//F1=100*cosd(theta)
//consider equilibrium of 900 N block
//N2=1200*cosd(theta)
//Law of friction
//F2=400*cosd(theta)
theta=atand(5/9)  //degree
printf("Required value is\ntheta=%0.2d degree",theta)
```

Scilab code Exa 5.3 `finding the inclination of the plane and coefficient of fricti`

```
//finding the inclination of the plane and coefficient of friction
//refer fig 5.7
//consider equilibrium of system
//Case (a)
//N=500*cosd(theta)
//Using law of friction
//F1=mu*N
//500*sind(theta)-500*mu*cosd(theta)=200
```

```
//Case (b)
//N=500*cosd(theta)
//usin law of friction
//F2=mu*N
//500*mu*cosd(theta)+500*sind(theta)=300
//add final equations from both cases
theta=asind(0.5)  //degree
//substitute this value in final equation from case
    (b)
mu=(50)/(500*cosd(30))
printf("\ntheta=%.2d degree\nmu=%0.3f",theta,mu)
```

Scilab code Exa 5.4 `Value of P`

```
   //Value of P
//Refer fig.5.8
//consider equilibrium
mu=0.2
//750N block
N1=750*cosd(60)  //N
F1=mu*N1  //N
T=F1+750*sind(60)  //N
//500N block
//N2=500-0.5P
//Law of friction
//F2=0.2*N2
P=(724.52+100)/(cosd(30)+0.1)  //N
printf("\nP=%0.2f N",P)
```

Scilab code Exa 5.5 Smallest weight W

```
    //Smallest weight W
//Refer fig. 5.9
mu=0.4
//consider equilibrium of block B
//using law of friction
N1=5/((0.5)+(tand(20))*(sind(20)))  //kN
F1=N1*tand(20)
C=N1*cosd(30)-F1*cosd(60)  //kN
//Consider the equilibrium of block A
F2=C  //kN
//Law of friction
N2=4.196/0.4  //kN
W=N2  //kN
printf("\nW=%0.2f kN",W)
```

Scilab code Exa 5.6 force required to prevent slipping

```
    //force required to prevent slipping
//refer fig. 5.10
mu=0.25
//assumptions are made and shown in fig.5.10
//F1=mu*N1
//consider equilibrium of block A
C1=(2000)/((0.25*cosd(30))+(0.5))  //N
N1=C1*cosd(30)
//Lami's theorem at joint O gives
P=(C1*sind(90))/sind(120)  //N
C=(C1*sind(150))/sind(120)  //N
//Consider equilibrium of block B for verification
//F2=C2*cosd(60)  N
//N2=2000+C2*sind(60)  N
```

```
//LF=mu*N2  N  (limiting friction)
//actual frictional force F2 developed is less than
   the limiting frictional force hence block B is
   stationary
//P is the correct answer
printf("Requiref force is\nP=%0.2f N",P)
```

Scilab code Exa 5.7 Least and the greatest value of W for equilibrium

```

  //Least and the greatest value of W for
   equilibrium
//refer fig.
//case (a) for least value Wmin
//refer fig. (b)
//considering equilibrium of 1000 N block
N1=866.03  //N
//law of friction gives
F1=0.28*N1  //N
T=500-242.49  //N
//consider equilibrium of W
//F2=0.1W
Wmin=266.57  //N
//case (b) for greatest value Wmax
//refer fig. 5.11 (c)
//consider equilibrium of 1000N block
T=742.49  //N
//consider equilibrium of W
//F2=0.2*0.5*W
Wmax=969.28  //N
printf("\nThe greatest and least values of W are:-\
   nWmax=%0.2f N\nWmin=%0.2f N",Wmax,Wmin)
```

Scilab code Exa 5.8 `Force P for impending motion`

```
    //Force P for impending motion
//Refer fig. 5.12
//consider equilibrium of block A
//NA*cosd(30)+FA*sind(30)-1500-500=0
//Law of friction gives
NA=2000  //N
FA=NA*tand(15)  //N
C=NA*sind(30)-FA*cosd(30)  //N
//consider equilibrium of block B
NB=2000*cosd(60)+535.90*cosd(30)  //N
FB=NB*tand(15)  //N
P=(392.30)+(2000*sind(60))-(535.90*sind(30))  //N
printf("The required force is P=%0.2f N",P)
```

Scilab code Exa 5.9 `Minimum force required`

```
    //Minimum force required
//refer fig. 5.13
//Applying Lami's theorem to system of forces on
    block
R1=20*sind(145)/sind(140)  //kN
R2=20*sind(75)/sind(140)  //kN
//Applying Lami's theorem to system of forces on
    wedge
P=R2*sind(130)/sind(105)  //kN
printf("required force is P=%0.2f kN",P)
```

Scilab code Exa 5.10 `Value of force P`

```
   //Value of force P
//refer fig. 5.14
mu=0.25
//Let fi be the angle of limiting friction
fi=atand(0.25)  //degree
 //Consider equilibrium of block C
//apply Lami's theorem
R1=160*sind(180-16-fi)/sind(2*(fi+16))  //kN
//Consider equilibrium of Wedge A
//apply Lami's theorem
P=R1*sind(180-fi-fi-16)/sind(90+fi)  //kN
printf("The required value is P=%0.3f kN",P)
```

Scilab code Exa 5.11 `Minimum horizontal force required to avoid slipping`

```
   //Minimum horizontal force required to avoid
   slipping
//refer fig.5.15
//consider equilibrium
//taking moment about A
//0.866*NB+0.5*FB=275
//Law of friction
//FB=0.2*NB
//Thus
 NB=275/(0.866+0.5*0.2)  //N
```

```
NA=200+600-56.934  //N
FA=0.3*NA  //N
P=NB-FA  //N
printf("The required force is P=%0.2d N",P)
```

Scilab code Exa 5.12 Least value of alpha and reactions developed

```
   //Least value of alpha and reactions developed
//refer fig. 5.16
//Using law of friction and equilibrium
//FA=0.25*NA
//FB=0.4*NB
//NA+0.4*NB=1100
//0.25*NA=NB
//Solving this we get
NA=1000  //N
FA=0.25*NA  //N
NB=0.25*NA  //N
FB=0.4*250  //N
//Taking moment about A
alpha=atand(3)  //degree
printf("\nNA=%0.2f N\nFA=%0.2f N\nNB=%0.2f N\nFB=%0
   .2f N\nalpha=%0.2f degree",NA,FA,NB,FB,alpha)
```

Scilab code Exa 5.13 A horizontal bar

```
//A horizontal bar
//refer fig. 5.17
//from law of friction
```

```
//FA=0.2*NA
//FB=0.2*NB
//Applying equilibrium condition
//0.6732*NA+0.5657*NB=1100
//NA=1.1077*NB
//Solving we get
NB=838.79  //N
NA=1.1077*NB  //N
//Taking moment about B
x=(-750+0.2*929.13*3*sind(60)+929.13*3*sind(30))
   /(600)  //m from B
//For impending motion
//Applying equilibrium condition
//0.3268*NA+0.8485*NB=1100
//solving
//NA=0.5856*NB
//Thus impending values are
INB=1057.82  //N
INA=0.5856*INB
//Taking moment about B
Ix=(-750+619.67*3*sind(30)-0.2*619.67*3*sind(30))
   /(600)  //m
printf("Thus 600N load can be placed anywhere
   between B and a point 1.85 m from B")
```

Scilab code Exa 5.14 rope and drum

```
//rope and drum
//refer fig. 5.19
//Angle of contact
theta=1.25*2*%pi  //radian
//Case (a)- Impending motion of weight be downward
aT1=600  //N
```

```
//aT2=W
//from law of rope friction
aW=600*((%e)^(0.75*%pi))  //N
//Case (b)-Impending motion of weight be upwards
//bT1=W
bT2=600  //N
//thus
bW=(600)/((%e)^(0.75*%pi))  //N
printf("Thus a 600 N force can support a range of
    loads between %.2f N to %.2f N",bW,aW)
```

Scilab code Exa 5.15 minimum weight W to prevent downward motion of 1000N block

```
    //minimum weight W to prevent downward motion of
    1000N block
mu1=0.2
mu2=0.3
//Refer fig. 5.20
alpha=atand(3/4)  //degree
//considering equilibrium of block W
//N1=W*cosd(alpha)
//F1=mu2*N1
//T1=0.84*W
theta=180  //degree
//Friction equation of rope gives
//T2=T1*%e^(mu2*theta)
//solving
//T2=2.156*W
//Consider equilibrium of 1000 N block
//N2-N1=800
//N2=0.8*W+800
//F2=0.3*N2
//F1+F2+T2-1000*sind(alpha)=0
```

```
//solving we get
W=(1000*sind(alpha)-240)/(0.24+0.24+2.156)   //N
printf("\nRequired force is W=%0.2f N",W)
```

Scilab code Exa 5.16 Determine force P

```
   //Determine force P
//refer fig.5.21
//From FBD
theta=250*%pi/180   //radians
r=250   //mm
mu=0.3
//from rope friction equation
//T2=T1*%e^(mu*theta)
//also (T2-T2)*r=M
//solving we get
T1=(300*1000)/(250*(3.7025-1))   //N
T2=3.7025*T1   //N
//Consider the equilibrium of lever arm,
P=(T2*50)/300   //N
printf("\n The required force is P=%0.1f N",P)
```

Chapter 6

Lifting machines

Scilab code Exa 6.1 Determine velocity ratio mechanical advantage efficiency ideal

```
//Determine velocity ratio,mechanical advantage, efficiency,ideal effort,effort lost in friction, ideal load and frictional resistance.
 W=10000  //N
P=500  //N
//distance moved by effort
D=20  //m
d=0.8  //m
//mechanical advantage
MA=W/P
printf("\nMechanical advantage=%0.1d",MA)
//Velocity ratio
VR=D/d
printf("\nVelocity ratio=%0.1d",VR)
//efficiency
e=MA*100/VR  //percent
printf("\nEfficiency=%0.1d percent",e)
//ideal effort
Pi=W/VR  //N
printf("\nIdeal effort=%0.1d N",Pi)
//Effort lost in friction
```

```
E=P-Pi  //N
printf("\nEffort lost in friction=%0.1d N",E)
//Ideal load
Wi=P*VR  //N
printf("\nIdeal load=%0.1d N",Wi)
//Frictional resistance
F=Wi-W  //N
printf("\nFrictional resistance=%0.1d N",F)
```

Scilab code Exa 6.2 `Simple machine`

```
//Simple machine
//assume law of machine P=m*W+C
//From first and second conditions we obtain 2 equations
//150=2400*m+C
//180=3000*m+C
//upon solving them
//P=0.05*W+30
//When force of 200 N is applied
W=(200-30)/(0.05)  //N
//Ideal effort
Pi=3400/30  //N
//Effort wasted in overcoming the friction
Ew=200-113.33  //N
//Mechanical advantage
MA=3400/200
//Efficiency
Eff=(17*100)/(30)  //percent
printf("\n Law of machining is P=0.05*W+30\nW=%.2f N \nEffort wasted in overcoming the friction=%.2f N \nMechanical advantage=%.2f \nEfficiency=%.2f percent",W,Ew,MA,Eff)
```

Scilab code Exa 6.3 maximum mechanical advantage and maximum efficiency

```
  //maximum mechanical advantage and maximum
     efficiency
//Effort
P=150  //N
W=7700  //N
MA=W/P  //mechanical advantage
//If efficiency=60%
eff=0.6
VR=(MA)/(eff)
//When an effort of 250 N raised a load of 13200 N
P1=250  //N
W1=13200  //N
MA1=(W1)/(P1)
eff1=MA1*100/VR  //percent
//assume law of machine as P=m*W+C
//from first case 150=7700*m+C
//from second case 250=13200*m+C
//Solving we get
m=100/5500
//maximum mechanical advantage
MAmax=1/(m)
//maximum efficiency
Effmax=100/(m*VR)  //percent
printf("\nMechanical advantage=%0.2f\nVelocity ratio
   =%0.2f\nEfficiency=%0.2f percent\nMaximum
   mechanical advantage=%0.2f\nMaxumum efficiency=%0
   .2f percent",MA,VR,eff1,MAmax,Effmax)
```

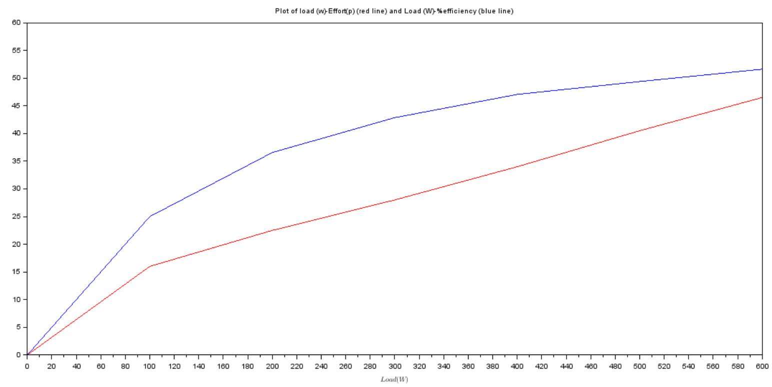

Figure 6.1: determine law of machine and maximum efficiency

Scilab code Exa 6.4 `determine law of machine and maximum efficiency`

```
   //determine law of machine and maximum efficiency
//fig.6.4 shows graph of effort vs load
//from graph
C=10  //N
//slope
m=30/500
//Law of machine is P=m*W+C
//P=0.06*W+10
//eta=MA/VR=W/25P
//table 6.5 shows calculation of efficiency for
   various loads
//Refer fig. 6.5
//from graph it can be seen that maximum efficiency
   =50% but the graph is plotted for infinitely
```

```
large load
//thus
//Maximum efficiency
VR=25
Emax=100/(m*VR)  //percent
printf("\nLaw of machines is P=0.06*W+10\nMaximum efficiency=%.2f percent",Emax)
//graph 1(g1)
x=[0,100,200,300,400,500,600]
y=[0,16,22.5,28.0,34,40.5,46.5]
plot(x,y,"r")
xtitle("Plot of load (w)-Effort(p) (red line) and Load (W)-%efficiency (blue line)")
xlabel('$Load (W)$')
//graph 2 (g2)
x=[0,100,200,300,400,500,600]
y=[0,25,36.56,42.86,47.06,49.38,51.61]
plot(x,y,"b")
//Red graph is load (w)- Effort(p)
//blue graph is Load (W)-%efficiency
```

Scilab code Exa 6.5 Lifting machine

```
//Lifting machine
VR=30
W=5000  //N
P=360  //N
MA=W/P
//efficiency
eta=MA*100/VR  //percent
printf("Since the efficiency eta=%0.2d percent is less than 50 percent,it is a self locking machine",eta)
```

```
//ideal load
Wi=P*VR   //N
//Frictional resistance
FR=Wi-W   //N
printf("\nIdeal load=%0.2d N\nFrictional resistance=%0.2d N",Wi,FR)
```

Scilab code Exa 6.6 Finding the additional pulleys required

```
//Finding the additional pulleys required
//if n is number of movable pulleys then
n=3
VR=2^n
eta=0.8
MA=eta*VR
P=6000/6.4   //N
//In second case
Effort=520   //N
//efficiency eta=0.80-n1*0.05
//n1=number of additional pulleys required=(n-3)
//thus
//W=P[0.8-(n-3)*0.05]*2^n
//by using trial and erroe method
nfinal=4
printf("Number of movable pulleys required=%.0f",nfinal)
```

Scilab code Exa 6.7 calculate force required

```
//calculate force required
//refer fig.6.15
//VR=2*number of movable pulleys
VR=2*3
eta=0.85
MA=eta*VR
W=12000
P=W/5.1
printf("The required force is P=%.2f N",P)
```

Scilab code Exa 6.8 `Pull required`

```
//Pull required
//Refer fig.6.16
//Let weight W be lifted by a distance x
//Consider first order system
//VR=2^2
//Consider second order system
VR=8
eta=0.78
MA=eta*VR
W=12000
P=W/6.24  //N
printf("Required pull P=%0.2f N",P)
```

Scilab code Exa 6.9 `machine efficiency and effort lost in friction`

```
//machine efficiency and effort lost in friction
//For third order system of pulleys
```

```
W=1000
VR=(2^3)-1
eta=(1000*100)/(180*7)   //percent
//ideal effort
Pi=(W)/(VR)   //N
P=180   //N
//effort lost in friction
Pl=P-Pi   //N
printf("Efficiency=%.2f percent\nEffort lost in
    friction=%.2f N",eta,Pl)
```

Scilab code Exa 6.10 Force P required to raise the load

```
//Force P required to raise the load
eta=0.70
W=2500   //N
//refer fig. 6.17
//For third order  pulley
//VR=2^2-1
//For whole system
VR=3+3
P=W/(eta*VR)   //N
printf("Required force p=%.2f N",P)
```

Scilab code Exa 6.11 find necessary effort

```
//find necessary effort
//effective wheel diameter
D=(6/2)+500+(6/2)   //mm
```

```
//effective axle diameter
d=(20/2)+200+(20/2)   //mm
VR=D/d
eta=0.7
MA=eta*VR
W=1200
P=1200/1.63   //N
printf("The effort necessary=%.2f N",P)
```

Scilab code Exa 6.12 `required effort`

```
//required effort
//differential axle diameters
d1=300   //mm
d2=250   //mm
//wheel diameter
D=800   //mm
//load
W=20000   //N
eta=0.55
VR=(2*D)/(d2-d1)
MA=eta*VR
P=W/MA   //N
printf("Required effort =%0.1f N",-P)
```

Scilab code Exa 6.13 `Effort required`

```
//Effort required
D=500   //mm
```

```
d=200   //mm
W=5000   //N
eta=0.6
VR=2*D/(D-d)
MA=eta*VR
P=W/MA   //N
printf("Required effort=%0.0f N",P)
```

Scilab code Exa 6.14 Force required at the end of lever

```
//Force required at the end of lever
d=40   //mm
p=20/3   //mm
W=40000   //N
R=400   //mm
mu=0.12
theta=atand(p/(%pi*d))   //degree
P=(d*W*(mu+tand(theta)))/(2*R*(1-mu*tand(theta)))
   //N
printf("Force required at the end of lever P=%0.2f N
   ",P)
```

Scilab code Exa 6.15 Screw jack parameters

```
//Screw jack parameters
p=10   //mm
d=50   //mm
W=6000   //N
theta=atand(p/(%pi*d))   //degree
```

```
fi=atand(0.05)  //degree
R=300  //mm
P=(d*W*tand(theta+fi))/(2*R)  //N
VR=(2*%pi*R)/(p)
MA=W/P
eta=MA*100/VR  //percent
//torque required to keep the load from descending
T=(50*600*tand(3.6426-2.8624))/2  //N-mm
printf("Efficiency eta=%0.2f percent > 50 percent\nThus the screw jack is not self locking\nTorque required to keep the load from descending T=%0.2f N-mm",eta,T)
```

Scilab code Exa 6.16 Differential screw jack

```
//Differential screw jack
//pitch
pA=10  //mm
pB=5  //mm
//lever arm length
R=500  //mm
VR=(2*%pi*R)/(pA-pB)
P=185  //N
W=15000  //N
MA=W/P
eta=MA*100/VR  //percent
//let the law of machine be P=m*W+C
//from first case  185=m*15000+C
//from second case  585=m*50000+C
//solving we get
m=4/350
C=185-(m*15000)  //N
printf("Law of machine is P=%.4f*W+%0.2d ",m,C)
```

Scilab code Exa 6.17 single purchase crab

```
//single purchase crab
//load drum radius
r=200/2  //mm
//Length of lever arm
R=1200  //mm
T2=100
T1=10
VR=(R*T2)/(r*T1)
//let the law of machine be P=m*W+C
//in first case  100=m*3000+C
//in second case  160=m*9000+C
//solving we get
m=1/100
C=70
//P=0.01*W+70
//case 1
MA1=3000/100
eta1=MA1*100/VR  //percent
//case 2
MA2=9000/160
eta2=MA2*100/VR  //percent
printf("\nVR=%.0f\nP=0.01*W+70\nIn first case\neta=
    %0.2f percent\nIn second case\neta=%0.2f percent"
    ,VR,eta1,eta2)
```

Scilab code Exa 6.18 Double purchase crab

```
//Double purchase crab
T1=15
T2=45
T3=20
T4=40
R=400   //mm
r=150/2   //mm
VR=(R*T2*T4)/(r*T1*T3)
eta=0.40
MA=eta*VR
W=12.8*250   //N
printf("Applied load lifts a load of W=%.2f N",W)
```

Chapter 7

Transmission of power

Scilab code Exa 7.1 Diameter of wheels

```
//Diameter of wheels
//refer fig.7.2
d1=240  //mm
N1=250  //rpm
N2=100  //rpm
d2=N1*d1/N2  //mm
ans1=d1+d2  //mm
d3=840/3  //mm
d4=840-d3  //mm
d5=(840)/(1+(5/3))  //mm
d6=840-d5  //mm
printf("\nd2=%.2f mm\nd3=%.2f mm\nd4=%.2f mm\nd5=%.2
   f mm\nd6=%.2f mm",d2,d3,d4,d5,d6)
```

Scilab code Exa 7.2 speed of follower

```
//speed of follower
d1=600  //mm
d2=400  //mm
N1=160  //rpm
N2=2  //rpm
//if there is no slip
aN2=(600*160)/(400)  //rpm
//when there is 2.5% slip
p=2.5/100
bN2=(N1*d1*(100-p))/(d2*100)  //rpm
printf("\nWhen there is no slip \nN2=%0.2f rpm\nWhen
    there is 2.5 percent slip \nN2=%0.2f rpm",aN2,
    bN2)
```

Scilab code Exa 7.3 Length of belt

```
//Length of belt
N2=80
N1=200
d1=240
d2=d1*N1/N2  //mm
r1=120  //mm
r2=300  //mm
l=2500  //mm
//length of crossbelt
L=%pi*(r1+r2)+2*l+((r1+r2)^2)/l  //mm
printf("The length of crossbelt L=%.2f mm",L)
```

Scilab code Exa 7.4 range of weight

```
//range of weight
//refer fig. 7.5
//angle of contact
theta=1.25*2*%pi
//case 1-Let impending motion of the weight be downward
//T1=600  //N
//T2=W
//from law of rope friction
//T2=T1*%e^mu*theta
aW=600*(%e^(0.3*2.5*%pi))  //N
//case-2 impending motion of W be upward
//T1=W
//T2=600 N
//from law of rope friction
bW=600/(%e^(0.75*%pi))  //N
printf("The range of weight that can be supported is from %.2f N to %.2f N",aW,bW)
```

Scilab code Exa 7.5 Maximum power that can be transmitted

```
//Maximum power that can be transmitted
alpha=asin((500-300)/2*2500)  //radians
//angle of contact
theta=%pi-2*alpha  //radians
f1=4
b=100
t=3
//T2 may be allowed upto
T2=f1*b*t  //N
mu=0.3
T1=1200/2.505  //N
r=500/2
```

```
omega=(2*%pi*100)/60
P=(T2-T1)*r*omega/1000000  //kW
printf("The maximum power that can be transmitted=%
    .3f kW",P)
```

Scilab code Exa 7.6 Force P required

```
//Force P required
//refer fig. 7.6
mu=0.3
theta=(250*%pi)/180  //radians
r=250  //mm
T1=(300*10^3)/(250*(%e^mu*theta-1))  //N
T2=(%e^mu*theta)*T1  //N
//Considering equilibrium of lever arm
P=(1644.06*50)/(300)  //N
printf("The required value is P=%.2f N",P)
```

Scilab code Exa 7.7 maximum power that can be transmitted

```
//maximum power that can be transmitted
mu=0.3
alpha=asin(((480/2)-(320/2))/2500)  //radians
theta=%pi-2*alpha
//neglecting centrifugal tension
f=3
b=150
t=8
//maximum force permitted is
```

```
T2=f*b*t  //N
T1=T2/%e^mu*theta
r=480/2
omega=(2*%pi*800)/60
aP=((3600-1429.82)*480*2*800*%pi)/(2*60*(10^6))  //
   kW
//If centrifugal tension is considered
v=r*omega/1000  //m/sec
m=1.32
Tc=m*v^2  //N
//maximum force that can be permitted on the belt
//T=T2+Tc=f*b*t
bT2=3600-533.62  //N
bT1=3066.38/2.5178  //N
//maximum torque that can be transferred
bP=(bT2-bT1)*v/1000  //kW
printf("\nneglecting centrifugal tension P=%0.3f kW\
   nConsidering centrifugal tension P=%0.3f kW",aP,
   bP)
```

Scilab code Exa 7.8 Maximum power transmitted

```
//Maximum power transmitted
m=0.9
v=20
Tc=m*v^2  //N
f=1.5
d=36
T=(f*%pi*d^2)/(4)  //N
T2=T-Tc  //N
mu=0.3
theta=220*%pi/180  //radians
T1=T2/(%e^mu*theta*(1/sind(30)))  //N
```

```
P=(T2-T1)*v/1000 //kW
printf("maximum power P=%.3f kW ",P)
```

Scilab code Exa 7.9 parallel shafts connected with spur gearing

```
//parallel shafts connected with spur gearing
l=540 //mm
//d2=1080/4 //mm
//d1=3*d2
//for a module of 8
//T1=810/8 but number of teeth is whole no.
T1=102
//T2=270/8 but number of teeth is whole no.
T2=34
//pitch circle diameter are
d1=102*8 //mm
d2=34*8 //mm
//The exact distance between the shafts
l=(d1+d2)/2 //mm
printf("\nT1=%.0d \nT2=%.0d \nl=%.0d mm",T1,T2,l)
```

Scilab code Exa 7.10 speed of pinion

```
//speed of pinion
P=480 //N
Pw=1800 //W
v=Pw/P //m/sec
//module
m=8
```

```
d=25*m  //mm
r=d/2000  //m
omega=v/r  //rad/sec
N=(60*omega)/(2*%pi)  //rpm
//rest is theory
printf("Speed of the pinion N=%.1f rpm",N)
```

Scilab code Exa 7.11 speed of driven shaft

```
//speed of driven shaft
//a-If the intermediate gears are on different shaft
TA=25
TD=100
NA=160  //rpm
aND=TA*NA/TD  //rpm
//b-If the intermediate gears are on the same shaft
bND=(75*25*160)/(50*100)  //rpm
printf("\ncase a\nND=%.2f rpm\ncase b\nND=%.2f rpm",
   aND,bND)
```

Chapter 8

Virtual work

Scilab code Exa 8.1 Resultant of system

```
//Resultant of system
//Refer fig. 8.5
//Let E be the equilibriant
//using virtual work principle
//-E*cosd(theta)=50*cosd(45)+80*cosd(25)+70*cosd(50)
   =152.86 N
//-E*sind(theta)=-50*sind(45)+80*sind(25)+70*sind
   (50)=52.07*sind(25)
//Thus
a=152.86  //N  (R*cosd(theta))
b=52.07  //N  (R*sind(theta))
R=sqrt(a^2+b^2)  //N
theta=atand(b/a)  //degree
printf("Resultant R=%.2f N inclined at theta=%.2f
   degree w.r.t positive x-axis",R,theta)
```

Scilab code Exa 8.2 Reactions developed in beam

```
//Reactions developed in beam
//refer fig 8.6
//Let RA and RB be the reactions at supports A and B
//applying virtual work principle
RB=(20/3)+(80/3)  //kN
RA=(40/3)+(40/3)  //kN
printf("Reactions are-\nRA=%.2f kN\nRB=%.2f kN",RA,
   RB)
```

Scilab code Exa 8.3 Reactions of overhanging beam

```
//Reactions of overhanging beam
//refer fig. 8.7
//Applying virtual work principle for beam in
   equilibrium
RB=(8/6)*(30+60*4/8-20/8)  //kN
RA=(7/6)*(20+60*2/7-30*2/7)  //kN
printf("The reactions are-\nRA=%0.2f kN\nRB=%.2f kN"
   ,RA,RB)
```

Scilab code Exa 8.4 Reactions at A

```
//Reactions at A
//refer fig. 8.8
//applying virtual work principle
//(RA-60)delta(y)=0
//thus
RA=60  //kN
printf("\n Reaction is RA=%.2f kN",RA)
```

Scilab code Exa 8.5 `Reaction at A in overhanging beam`

```
//Reaction at A in overhanging beam
//refer fig. 8.9
//Applying virtual work principle
//(-2*RA+180-20)*delta(y)=0
//Thus
RA=(180-20)/2   //kN
printf("\n Reaction is RA=%.2f kN",RA)
```

Scilab code Exa 8.6 `Reaction at B`

```
//Reaction at B
//refer fig. 8.10
//Applying virtual work principle
//(-4-6+RB)*delta(y)=0
//Thus
RB=6+4   //kN
printf("Reaction at B is RB=%.2f kN",RB)
```

Scilab code Exa 8.7 `Determine reaction RA`

```
//Determine reaction RA
```

```
//refer fig. 8.11
//give virtual displacement at A
//Applying virtual work principle
//(RA-32+8+0+4)*delta(y)=0
RA=32-8-4  //kN
printf("Reaction at A is RA=%.2f kN",RA)
```

Scilab code Exa 8.8 Determine reaction RA

```
//Determine reaction RA
//refer fig.8.12
//Applying virtual work principle
//give virtual displacement at A
//Virtual work equation is
//(RA-20+8)*delta(y)=0
RA=20-8  //kN
printf("The reaction RA=%.2d kN",RA)
```

Scilab code Exa 8.10 Horizontal force P

```
//Horizontal force P
//refer fig.8.14
//Applying virtual work principle
//-1500*(delta(s)*sind(30))+P*delta(s)*cosd(30)+0=0
P=1500*tand(30)  //N
printf("Magnitude of P=%.2f N",P)
```

Scilab code Exa 8.11 Range of force P

```
//Range of force P
//refer fig. 8.15
//a-when the motion is impending up the plane
mu=0.3
N=1000*cosd(70)
F=mu*N  //N
//Applying virtual work principle
aP=1000*sind(70)+300*cosd(70)  //N
//b-when the motion is impending down the plane
//Applying virtual work principle
bP=1000*sind(70)-300*cosd(70)  //N
printf("Block is in equilibrium for P=%.2f N to %.2f
    N",bP,aP)
```

Scilab code Exa 8.12 Position of the balls

```
//Position of the balls
//refer fig. 8.16
//Let a virtual displacement be given to the system
    of balls as shown
//Applying virtual work principle
//200*sind(30)*delta(DB)+150*sind(60)*delta(EB)=0
theta=atand((150*sind(60))/(200*sind(30)))  //degree
printf("Thus theta=%.2f degree",theta)
```

Scilab code Exa 8.13 Force P to make motion impending to the left

```
//Force P to make motion impending to the left
//Refer fig.8.17
N1=250
N2=1000*cosd(45)
N3=500
F1=0.25*N1  //N
F2=0.25*N2  //N
F3=0.25*N3  //N
//let us give virtual displacement towards left
//Applying virtual work principle
//(P-F1-1000*sind(45)-F2-F3)*delta(s)=0
P=F1+1000*sind(45)+F2+F3  //N
printf("The required force is P=%.2f N",P)
```

Scilab code Exa 8.14 Determine WB

```
//Determine WB
//refer fig.8.18
theta=atand(3/4)
mu=0.3
WA=200
F=mu*WA*cosd(theta)  //N
//Let us give virtual displacement of delta(s) up the plane to block A
//Applying virtual work principle
//(-200*sind(theta)-F+WB/2)*delta(s)=0
```

```
WB=2*(200*sind(theta)+F)  //N
printf("Required value of WB=%.2f N",WB)
```

Scilab code Exa 8.15 Determine force P

```
//Determine force P
//refer fig.8.19
//(a) P at floor level
//Applying virtual work principle
//P*delta(x)-200*delta(y)=0
P=100*tand(30)  //N
//(b) If the rope is used instead of force P
//refer fig.8.20
//Taking C as origin
//Applying virtual work principle
//(-3*T*cosd(theta)+400*sind(theta))*delta(theta)=0
T=(400*tand(30))/3  //N
printf("\nP=%.2f N\nT=%.2f N",P,T)
```

Scilab code Exa 8.16 Least value of theta to avoid slipping of ladder

```
//Least value of theta to avoid slipping of ladder
//refer fig.8.21
//Applying virtual work principle
//-0.4*NA*6*cosd(theta)*delta(theta)-200*(-3*sind(
    theta)*delta(theta))-900*(-5*sind(theta)*delta(
    theta))+0.25*NB(-6*sind(theta)*delta(theta))=0
NA=1100/(1+0.25*0.4)  //N
NB=0.4*1000  //N
```

```
theta=atand(2400/4500)   //degree
printf("Thus theta=%.2f degree",theta)
```

Scilab code Exa 8.17 Force in the member FH

```
//Force in the member FH
//refer fig.8.22
//use symmetry
//Apply virtual work principle
//RA*12*delta(theta)-10*10*delta(theta)-10*6*delta(
    theta)-10*2*delta(theta)+FFH*2*tand(60)=0
FFH=-240/(2*tand(60))   //kN
printf("FFH is a compressive force of %.2f kN",-FFH)
```

Scilab code Exa 8.18 Force developed in member DF

```
//Force developed in member DF
//refer fig. 8.23
//Applying virtual work principle
//-100*10*delta(theta)-100*5*delta(theta)+FDF*5*
    delta(theta)*cosd(45)=0
FDF=1500/(5*cosd(45))   //kN (tensile)
printf("Thus force developed in member DF is FDF=%.2
    f kN (Tensile)",FDF)
```

Chapter 9

Centroid and moment of inertia

Scilab code Exa 9.1 Locating the centroid of T section

```
//Locating the centroid of T section
//Refer fig. 9.10
//due to symmetry the centroid lies on y axis
//distance of centroid from top is
ybar=(100*20*10+20*100*70)/(100*20+20*100)
printf("Centroid of T-section is on the symmetric
   axis at a distance of %.2f mm from the top",ybar)
```

Scilab code Exa 9.2 Centroid of angle

```
//Centroid of angle
//refer fig. 9.11
//the given figure can be divided into two
   rectangles
A1=150*12  //mm^2
A2=(200-12)*12  //mm^2
//total area
```

```
A=A1+A2  //mm^2
xbar=(1800*75+2256*6)/(4056)  //mm
ybar=(1800*6+2256*106)/(4056)  //mm
printf("The centroid is at \nxbar=%.2f mm\nybar=%.2f
    mm ",xbar,ybar)
```

Scilab code Exa 9.3 locating centroid

```
//locating centroid
//refer fig. 9.12
//due to symmetry centroid must lie on y-axis
xbar=0
A1=100*20  //mm^2
//for A1
y1=30+100+20/2  //mm
//for A2
A2=100*20  //mm^2
y2=30+100/2  //mm
A3=150*30  //mm^2
y3=30/2  //mm
A=2000+2000+4500
ybar=(A1*y1+A2*y2+A3*y3)/A  //mm
printf("The centroid is on the symmetric axis at a
   distance of %.2f mm from the bottom as shown in
   figure 9.12",ybar)
```

Scilab code Exa 9.4 Centroid of dam

```
//Centroid of dam
```

```
//refer fig. 9.13 and select axis accordingly
A1=2*6/2  //m^2
A2=2*7.5  //m^2
A3=3*5/2  //m^2
A4=1*4  //m^2
A=A1+A2+A3+A4  //m^2
//centroid of simple geometries are
x1=2*2/3  //m
y1=6/3  //m
x2=2+1  //m
y2=7.5/2  //m
x3=2+2+3/3  //m
y3=1+5/3  //m
x4=4+4/2  //m
y4=0.5  //m
xbar=(A1*x1+A2*x2+A3*x3+A4*x4)/(A)  //m
ybar=(A1*y1+A2*y2+A3*y3+A4*y4)/(A)  //m
printf("centroid is at\nxbar=%.3f mm\nybar=%.3f mm",
   xbar,ybar)
```

Scilab code Exa 9.5 Determine centroid

```
//Determine centroid
//refer fig. 9.14
//This figure is divided into three simple figures
A1=3*4/2  //m^2
A2=6*4  //m^2
A3=%pi*(1/2)*2^2  //m^2
A=A1+A2+A3  //m^2
//Co-ordinates of centroid
x1=6+3/3  //m
ya=4/3  //m
x2=3  //m
```

```
y2=2   //m
R=2
x3=(-4*R)/(3*%pi)   //m
y3=2   //m
xbar=(A1*x1+A2*x2+A3*x3)/(A)   //m
ybar=(A1*ya+A2*y2+A3*y3)/(A)   //m
printf("\nxbar=%.3f m\nybar=%.3f m",xbar,ybar)
```

Scilab code Exa 9.6 Centroid of the gusset plate

```
//Centroid of the gusset plate
//refer fig. 9.15
//The composite area is divided into algebraic sum
   and differences of simple geometries
//for rectangle
A1=160*280   //mm^2
x1=140   //mm
y1=80   //mm
//for triangle
A2=280*40/2   //mm^2
x2=2*280/3   //mm
y2=160+40/3   //mm
//1st hole
A3=(-%pi*21.5^2)/(4)   //mm^2
x3=70   //mm
y3=50   //mm
//second hole
A4=-363.05   //mm^2
x4=140   //mm
y4=50   //mm
//third hole
A5=-363.05   //mm^2
x5=210   //mm
```

```
y5=50  //mm
//fourth hole
A6=-363.05  //mm^2
x6=70  //mm
y6=120  //mm
//fifth hole
A7=-363.05  //mm^2
x7=140  //mm
y7=130  //mm
//sixth hole
A8=-363.05  //mm^2
x8=210  //mm
y8=140  //mm
A=A1+A2+A3+A4+A5+A6+A7+A8  //mm^2
sumAixi=A1*x1+A2*x2+A3*x3+A4*x4+A5*x5+A6*x6+A7*x7+A8
    *x8  //mm^3
xbar=sumAixi/A  //mm
sumAiyi=A1*y1+A2*y2+A3*y3+A4*y4+A5*y5+A6*y6+A7*y7+A8
    *y8
ybar=sumAiyi/A  //mm
printf("\xbar=%.3f mm \nybar=%.3f mm",xbar,ybar)
```

Scilab code Exa 9.7 Determine co ordinates

```
//Determine co-ordinates
//total area
A=200*150-(100*75/2)-((%pi*100^2)/(4))  //mm^2
xc=2375000/26250  //mm
yc=1781250/26250  //mm
printf("\xc=%.3f mm\nyc=%.3f mm",xc,yc)
```

Scilab code Exa 9.8 Locate centroid

```
//Locate centroid
//refer fig.9.17
x=40  //mm
A1=168*x^2
A2=12*x^2
A3=-16*x^2
A4=-8*%pi*x^2
A5=-4*%pi*x^2
x1=7*x
x2=16*x
x3=2*x
x4=6*x
x5=12.3023*x
y1=6*x
y2=4*x/3
y3=10*x
y4=(16*x/(3*%pi))
y5=10.3023*x
A=126.3009*x^2
sumAixi=1030.6083*x^3
sumAiyi=691.8708*x^3
xbar=1030.6083*x/126.3009  //mm
ybar=691.8708*x/126.3009  //mm
printf("centroid is at (%.2f, %.2f)",xbar,ybar)
```

Scilab code Exa 9.9 Determine moment of inertia

```
//Determine moment of inertia
//refer fig.9.34
//composite section can be divided into simple ones
A1=150*10  //mm^2
A2=140*10  //mm^2
A=A1+A2  //mm^2
//due to symmetry centroid lies on y-axis
ybar=(1500*5+1400*(10+70))/(2900)  //mm
Ixx=((150/12)*10^3)+(1500*36.21^2)+((10/12)*140^3)
    +(1400*38.79^2)  //mm^4
Iyy=((10*150^3)/(12))+((140*10^3)/(12))  //mm^4
kxx=sqrt(Ixx/A)  //mm
kyy=sqrt(Iyy/A)  //mm
printf("\nIxx=%.2f mm^4\nIyy=%.2f mm^4\nkxx=%.2f mm\
    nkyy=%.2f mm",Ixx,Iyy,kxx,kyy)
```

Scilab code Exa 9.10 Moment of inertia of L section

```
//Moment of inertia of L-section
//Divide the section into two rectangles A1 and A2
A1=125*10  //mm^2
A2=75*10  //mm^2
A=A1+A2  //mm^2
xbar=((1250*5)+750*(10+75/2))/A  //mm
ybar=((1250*125/2)+(750*5))/A  //mm
Ixx=((10*125^3)/12)+(1250*21.56^2)+((75/12)*10^3)
    +(750*39.94^2)  //mm^4
Iyy=((125*10^3)/12)+(1250*15.94^2)+((10*75^3)/12)
    +(750*26.56^2)  //mm^4
Izz=Ixx+Iyy  //mm^4
printf("\nIxx=%.2f mm^4\nIyy=%.2f mm^4\nIzz=%.2f mm
    ^4",Ixx,Iyy,Izz)
```

Scilab code Exa 9.11 Moment of inertia of I section

```
//Moment of inertia of I section
//Refer fig. 9.36
A1=200*9   //mm^2
A2=(250-9*2)*6.7   //mm^2
A3=200*9   //mm^2
A=A1+A2+A3   //mm^2
Ixx=((200*9^3)/12)+(1800*120.5^2)+((6.7*232^3)/12)
    +(0)+((200*9^3)/12)+(1800*120.5^2)   //mm^4
Iyy=((9*200^3)/12)+((232*6.7^3)/12)+((9*200^3)/12)
    //mm^4
Izz=Ixx+Iyy   //mm^4
printf("\nIxx=%.2f mm^4\nIyy=%.2f mm^4\nIzz=%.2f mm
    ^4",Ixx,Iyy,Izz)
//The answers vary due to round off error
```

Scilab code Exa 9.12 Second moment of Inertia

```
//Second moment of Inertia
//refer fig. 9.37
A1=100*13.5   //mm^2
A2=(400-27)*8.1   //mm^2
A3=100*13.5   //mm^2
A=A1+A2+A3   //mm^2
Ixx=((100*13.5^3)/12)+(1350*193.25^2)+((8.1*373^3)
    /12)+((100*13.5^3)/12)+(1350*193.25^2)   //mm^4
```

```
Iyy=((13.5*100^3)/12)+(1350*24.27^2)+((373*8.1^3)
    /12)+(3021.3*21.68^2)+((13.5*100^3)/12)
    +(1350*24.27^2)
printf("\nIxx=%.2d mm^4\nIyy=%.2d mm^4",Ixx,Iyy)
```

Scilab code Exa 9.13 Polar moment of inertia

```
//Polar moment of inertia
//Refer fig.9.38
//section can be divided into three triangles
A1=80*12  //mm^2
A2=(150-22)*12  //mm^2
A3=120*10  //mm^2
A=A1+A2+A3  //mm^2
Ixx=((80*12^3)/12)+(960*74.22^2)+((12*128^3)/12)
    +(1536*4.22^2)+((120*10^3)/12)+(1200*64.78^2)  //
    mm^4
Iyy=((12*80^3)/12)+((128*12^3)/12)+((10*120^3)/12)
    //mm^4
//Polar moment of Inertia (Izz)
Izz=Ixx+Iyy  //mm^4
kxx=sqrt(Ixx/A)  //mm
kyy=sqrt(Iyy/A)  //mm  (The answer provided in the
    textbook is wrong)
printf("\nIzz=%.2d mm^4\nkxx=%.2f mm\nkyy=%.2f mm",
    Izz,kxx,kyy)
```

Scilab code Exa 9.14 Determine moment of inertia

```
//Determine moment of inertia
//refer fig.9.39
A1=100*30   //mm^2
A2=100*25   //mm^2
A3=200*20   //mm^2
A4=87.5*20/2   //mm^2
A5=87.5*20/2   //mm^2
A=A1+A2+A3+A4+A5   //mm^2
ybar=(3000*135+2500*70+4000*10+875*(20/3+20)*2)/A
    //mm
Ixx=((100*30*30*30)/(12))+(3000*(75.74^2))
    +((25*(100^3))/(12))+(2500*(10.74^2))
    +((200*(20^3))/(12))+(4000*(49.26^2))
    +((87.5*(20^3))/(36))+(875*(32.59^2))
    +((87.5*(20^3))/(36))+(875*(32.59^2))   //mm^4
Iyy=((30*(100^3))/(12))+((100*(25^3))/(12))
    +((20*(200^3))/(12))+((20*(87.5^3))/(36))
    +(875*(41.66^2))+((20*(87.5^3))/(36))
    +(875*(41.66^2))   //mm^4
printf("\nIxx=%.2d mm^4\nIyy=%.2d mm^4",Ixx,Iyy)
//The answers vary due to round off error
```

Scilab code Exa 9.15 Moment of inertia of section

```
//Moment of inertia of section
//refer fig. 9.40
IAB=((400*20^3)/12)+(400*20*10^2)+(((100*10^3)/12)
    +100*10*(20+5)^2)*2+(((10*380^3)/12)
    +10*380*(30+190)^2)*2+(((100*10^3)/12)
    +100*10*(20+10+380+5)^2)*2   //mm^4
printf("IAB=%.2d mm^4",IAB)
```

Scilab code Exa 9.16 Moment of inertia

```
//Moment of inertia
//refer fig.9.41
sumAiyi=250*10*5+2*40*10*(10+20)+40*10*(10+5)
   +40*10*255+250*10*(10+125)   //mm^3
A=2*250*10+40*10*4   //mm^2
ybar=sumAiyi/A
Ixx=((250*10^3)/12)+(250*10*(73.03-5)^2)+(((10*40^3)
   /12)+40*10*(73.03-30)^2)*2+((40*10^3)/12)
   +40*10*(73.03-15)^2+((10*250^3)/12)
   +250*10*(73.03-135)^2+((40*10^3)/12)
   +40*10*(73.03-255)^2   //mm^4
printf("Ixx=%.2d mm^4",Ixx)
```

Scilab code Exa 9.17 moment of inertia

```
//moment of inertia
//refer fig.9.42
sumAiyi=(600*15)*((600/2)+20)+140*10*2*(70+30)
   +150*10*2*(5+20)+400*20*10   //mm^3
A=600*15+140*10*2+150*10*2+400*20   //mm^2
ybar=sumAiyi/A   //mm
Ixx=((15*(600^3))/(12))+(600*15*((145.39-320)^2))
   +((10*2*(140^3))/(12))+(1400*2*((145.39-100)^2))
   +((150*2*(10^3))/(12))+(1500*2*((145.39-15)^2))
   +((400*(20^3))/(12))+(400*20*((145.39-10)^2))
printf("Ixx=%.2f mm^4",Ixx)
```

```
//The answer provided in the textbook is wrong
```

Scilab code Exa 9.18 Compute moment of inertia

```
//Compute moment of inertia
//refer fig. 9.43
Ixx=((125*60^3)/36)+(125*(60/2)*(60+60/3)^2)
   +((125*60^3)/36)+(125*(60/2)*(2*60/3)^2)
   +((125*60^3)/36)+(125*(60/2)*(60/3)^2)
   +((125*60^3)/36)+(125*(60/2)*(60/3)^2)   //mm^4
printf("Ixx=%.2d mm^4",Ixx)
```

Scilab code Exa 9.19 moment of inertia of shaded region

```
//moment of inertia of shaded region
//refer fig. 9.44
//The figure is divided into simple geometry
IAB=((80*80^3)/12)+((%pi*80^4)/128)-((%pi*40^4)/64)
    //mm^4
printf("IAB=%.2d mm^4",IAB)
```

Scilab code Exa 9.20 Find second moment of inertia

```
//Find second moment of inertia
```

```
ybar=28.47   //mm
xbar=39.21   //mm
Ixx=((80*20^3)/36)+(80*(20/2)*(60-(2*20/3)-28.47)^2)
    +((80*40^3)/12)+(80*40*(28.47-20)^2)
    -((0.0068598*20^4)+(20^2)*(%pi/2)*(28.47-((4*20)
    /(3*%pi)))^2)   //mm^4
Iyy=((20*30^3)/36)+((20/2)*30*(39.21-(2*30/3))^2)
    +((20*50^3)/36)+(20/2)*50*(39.21-(30+50/3))
    ^2+((40*80^3)/12)+(40*80*(39.21-40)^2)-(%pi*40^4)
    /(2*64)-((%pi)*(40^2)*(40-39.21)^2)/(4*2)   //mm^4
printf("\nIxx=%.2d mm^4\nIyy=%.2d mm^4",Ixx,Iyy)
```

Chapter 10

Centre of gravity and mass moment of inertia

Scilab code Exa 10.4 Locate centre of gravity

```
//Locate centre of gravity
//refer fig. 10.5
W1=0.6*0.75*0.5*25000  //N
W2=(%pi*(0.2^2)*0.3*25000)/(4)  //N
sumWi=7889.38
sumWixi=3241.57
sumWiyi=2593.25
sumWizi=1745.91
xbar=(sumWixi)/(sumWi)
ybar=(sumWiyi)/(sumWi)
zbar=(sumWizi)/(sumWi)
printf("\nxbar=%.3f m\nybar=%.3f m\nzbar=%.3f m",
   xbar,ybar,zbar)
```

Scilab code Exa 10.5 Locate centroid

```
//Locate centroid
//Refer fig.10.6
//lets assign random value to w
w=1
sumWi=1053.98*w
sumWixi=95055.54*w
sumWiyi=125214.83*w
sumWizi=59201.4*w
xbar=(sumWixi)/(sumWi)
ybar=(sumWiyi)/(sumWi)
zbar=(sumWizi)/(sumWi)
printf("\nxbar=%.2f mm\nybar=%.2f mm\nzbar=%.2f mm",
   xbar,ybar,zbar)
```

Scilab code Exa 10.13 Determine radius of gyration

```
//Determine radius of gyration
//refer fig.10.23
//composite body may be divided into
//1.a solid block of size (80*120*100 mm) and 2.two
   semicircular grooves each of 40 mm radius and 80
   mm length
//Let's assign random value to rho
rho=1
//Ig=1.029*10^8*rho
//Ix2=Ig+M2*d'^2
Ixx=10.0816*(10^8)*rho
M=557876.14*rho  //units
k=sqrt(Ixx/M)  //mm
printf("\nk=%.2f mm",k)
```

Scilab code Exa 10.14 Moment of Inertia of flywheel

```
//Moment of Inertia of flywheel
//refer fig. 10.24
//Moment of inertia of rim
aRo=1.5/2
aRi=1.4/2
at=0.30
rho=7200  //kg/m^3
I1=((%pi)*0.3*7200*(0.75^4-0.7^4))/(2)  //units
//Moment of inertia of hub
bRo=0.25/2  //m
bRi=0.1/2  //m
bt=0.2  //m
I2=(%pi)*(0.2*7200)*(0.125^4-0.05^4)/(2)  //units
//Moment of inertia of Arms
A=8000*(10^-9)  //m^2
l=0.575  //m
d=(0.575/2)+0.125  //m
M=l*A*rho  //kg
//there are six such arms
I3=6*0.03312*((0.575)^2/(12))*(0.4125^2)  //units
//moment of inertia of flywheel
I=I1+I2+I3  //units
printf("\nmoment of inertia of flywheel=%.2f units",
    I)
```

Chapter 12

Linear motion

Scilab code Exa 12.2 Steel ball shot vertically up

```
//Steel ball shot vertically up
//refer fig.12.6
//For upward motion
au=18  //m/sec
av=0
aa=-9.81  //m/sec^2
//s=h
//let t1 be the time required to reach maximum
   height
t1=1.83  //sec
h=(18^2)/(2*9.81)  //m
//total height from the ground
ah=25+h  //m
//Downward motion
bu=0
bs=41.51  //m
ba=9.81  //m/sec^2
v2=sqrt(2*9.81*41.51)  //m/sec
t2=28.54/9.81  //m/sec
//total time during which the body is in motion
t=t1+t2  //sec
```

```
printf("\nt1=%.2f sec\nh=%.2f m\nv2=%.2f m/sec\nt2=%
    .2f sec\nt=%.2f sec",t1,h,v2,t2,t)
```

Scilab code Exa 12.3 Height from which stone fell

```
//Height from which stone fell
//refer fig.12.7
//Let the stone be dropped from A at a height h
    above window
//h=(g*t^2)/2   ...(1)
//h+2.45=((g)*(t+0.5)^2)/2   ...(2)
//from (1) and (2)
t=0.2495  //sec
g=9.81  //m/sec^2
h=(g*t^2)/2  //m
printf("\nh=%.3f m",h)
```

Scilab code Exa 12.4 Crossing of balls

```
//Crossing of balls
//refer fig. 12.8
//1.for motion of first ball
au=0
//1s=30-h
aa=9.81  //m/sec^2
//2.for motion of second ball
bu=15  //m/sec
//s=h
ba=-9.81  //m/sec^2
```

```
//30-h=0*t+(9.81*t^2)/2   ...(1)
//h=15*t-(9.81*t^2)/2   ...(2)
//solving (1) and (2)
t=30/15
h=15*2-(9.81*2^2)/2   //m
//at t=2
//downward velocity of first ball
v1=0+9.81*2   //m/sec
//Upward velocity of second ball
v2=15-9.81*2   //m/sec
//relative velocity vr
vr=v1-(-v2)   //m/sec
printf("\nt=%.2f sec\nh=%.2f m\nvr=%.2f m/sec",t,h,vr)
```

Scilab code Exa 12.5 `Stone dropped into well`

```
//Stone dropped into well
//let
//h=depth of well
//t1=time tataken by stone to strike water
//t2=time taken by sound to travel h
//t1+t2=4
//h=(g*t1^2)/2
//h=335*t2
//solving
//t1^2+68.30*t1-273.19=0
t1=3.79   //sec
h=(9.81*t1^2)/2   //m
printf("h=%.2f m",h)
```

Scilab code Exa 12.6 Distance covered

```
//Distance covered
//refer fig.12.9
//Let the particle start from A and come to halt at
   E
//Let initial velocity be u m/sec
//consider motion between A and B
//u+a=10
//consider motion between A and C
//70=7*u+7*a
//solving
a=-10/17.5  //m/sec^2
u=10-(a)  //m/sec
//Let distance AD be s1
s1=10.571*10+(-0.571*10^2)/2  //m
//Distance covered in the interval 7 sec to 10 sec
CD=77.16-60  //m
//Let AE=s
s=(10.571^2)/(2*0.571)  //m
printf("\nCD=%.2f m\ns=%.2f m",CD,s)
```

Scilab code Exa 12.7 motorist and traffic light

```
//motorist and traffic light
//initial velocity
u=(80*1000)/(60*60)  //m/sec
t=10 //sec
```

```
s=200  //m
//a be acceleration
//using equation of motion
a=(200-22.22*10)*2/10^2  //m/sec^2
//final velocity
v=(22.22-0.444*10)*(3600/1000)  //kmph
printf("\na=%.2f m/sec^2\nv=%.2f kmph",a,v)
```

Scilab code Exa 12.9 time required to cover the distance between two stations

```
//time required to cover the distance between two
   stations
//refer fig.12.11
v=(48*1000)/(60*60)  //m/sec
t1=30  //sec
//after application of brakes the vehicle retards
   from 13.33 m/sec to 0 in t3 sec
t3=13.33  //sec
//Let t2 be the time during which the automobile
   travels with uniform velocity
//s=s1+s2+s3
s=5200  //m
t2=((5200)-(13.33*30/2)-(13.33*13.33/2))/13.33  //
   sec
//total time
t=t1+t2+t3  //sec
printf("Total time taken=%.2f sec",t)
```

Scilab code Exa 12.10 cage and mine shaft

```
//cage and mine shaft
//t is time during which stone is in motion
//s=(9.81*t^2)/2
//consider motion of cage
//t1 be the time taken to travel first 30 m
a=0.6  //m/sec^2
t1=10  //sec
//When the stone strikes
//s=(0.6*(t+10)^2)/2
//solving
t=3.286  //sec
s=(9.81*3.286^2)/2  //m
printf("\nt=%.2f sec\ns=%.2f m",t,s)
```

Scilab code Exa 12.11 Train B overtakes train A

```
//Train B overtakes train A
//refer fig.12.12
//speed of A
v1=7.5  //m/sec
//speed of B
v2=15  //m/sec
//motion of train A
//using equation of motion
t1=7.5/0.15  //sec
//distance travelled in time t
//s=7.5*t-187.5
//Motion of train B
//using equation of motion
t2=15/0.3  //sec
//distance travelled t seconds after train A started
//s=15*t-975
//solving
```

```
t=(975-187.5)/(15-7.5)   //sec
s=15*t-975   //m
printf("\nTrain B overtakes train A %.2d sec\ns=%.2d m",t,s)
```

Scilab code Exa 12.12 Two cars

```
//Two cars
//refer fig.12.13
//Let A and B be the positions of cars when the drivers see each other and apply brakes
//Let they meet at C
//1.car A
au=12   //m/sec
av=0
//s=x
//a1 be acceleration
//using equation of motion
//a1=(-12)/t
//x=6*t
//2.car B
bu=9   //m/sec
bv=0
//a=a2
//time=t
//s=100-x
//using equation of motion
//a2=-9/t
//100-x=4.5*t
//solving
t=100/10.5   //sec
a1=-12/t   //m/sec^2
a2=-9/t   //m/sec^2
```

```
x=57.14  //m
//distance traveled by second car
bx=100-x  //m
printf("\nt=%.2f sec\na1=%.2f m/sec^2\na2=%.2f m/sec^2\nDistance travelled by first car=%.2f m\nDistance travelled by second car=%.2f m",t,a1,a2,x,bx)
```

Scilab code Exa 12.13 `Car and truck`

```
//Car and truck
//refer fig.12.14 and 12.15
u=12.5  //m/sec
//sT=10+12.5*t+(aT*t^2)/2
aT=-2  //m/sec^2
//t is the time at any instant after the brakes are applied
//sT=10+12.5*t-t^2
//distance moved by car
//sC=u*2+u*(t-2)+(aC*(t-2)^2)/2
//sT=sC
//Apply equations of motion
//we get quadratic equation whose solution gives
aC=-10/3  //m/sec^2
printf("the deceleration of the car is=%.2f m/sec^2",aC)
```

Scilab code Exa 12.14 `motion of particle`

```
//motion of particle
//s=t^3-3*t^2+2*t+5
//v=ds/dt=3*t^2-6*t+2
//a=6*t-6
//after 4 seconds
v=3*4*4-6*4+2   //m/sec
a=6*4-6   //m/sec^2
//minimum velocity Vmin by using maxima and minima principle
Vmin=-1   //m/sec
//let at time t the velocity be zero,then
t1=1.577   //sec
t2=0.423   //sec
printf("\nv=%.2f m/sec\na=%.2f m/sec^2\nMinimum velocity=%.2f m/sec\nVelocity is zero at t=%.2f sec and %.2f sec",v,a,Vmin,t1,t2)
```

Scilab code Exa 12.15 `particle in motion`

```
//particle in motion
//v=t^3-t^2-2*t+2
//a=3*t^2-2*t-2
//acceleration after 4 seconds
a=3*4^2-2*4-2   //m/sec^2
//s=(t^4)/4-(t^3)/3-(t^2)+2*t+C
//c is constant of acceleration
//applying given condition
C=4/3
s=(4^4)/4-(4^3)/3-(4^2)+2*4+(4/3)   //m
//using maxima and minima principle
//minimum value of acceleration (amin)
amin=3*((1/3)^2)-2*(1/3)-2   //m/sec^2
printf("\nMinimum value of acceleration=%.2f m/sec^2
```

```
",amin)
```

Scilab code Exa 12.16 body moving along straight line

```
//body moving along straight line
//refer fig. 12.16
//a=2-3*t
//v=2*t-(3/2)*(t^2)+C1
//C1 is constant of integration
//v=20 //m/sec
//t=5 //sec
//thus
C1=47.5
//s=47.5*t+t^2-0.5*t^3+C2
//s=85 m when t=10 sec thus
C2=10
a=2-3*0 //m/sec^2
v=47.5 //m/sec
as=10 //m
//let the time when velocity becomes zero be t, thus
t=6.33 //sec
//Corresponding distance from origin
s=10+47.5*6.33+6.33^2-0.5*6.33^3
printf("\na=%.2f m/sec^2\nv=%.2f m/sec\ns=%.2f m\nt=%.2f sec\nDistance from origin\ns=%.3f",a,v,as,t,s)
```

Scilab code Exa 12.18 Car moving

```
//Car moving
//let the expression for retardation be
//a=-k*s   ...k=constant
//v^2/2=(-k*(s^2)/2)+C1
//When brakes are applied
//s=0 and v=72 kmph
v=20  //m/sec
C1=200
//when vehicle stops
//v=0  s=15 m
k=400/225
//expression for retardation is
//a=-1.778*s    ...theory approach
printf("The expression for retardation is a=-1.778*s
  ")
```

Chapter 13

Projectiles

Scilab code Exa 13.1 Pilot and his bomber

```
//Pilot and his bomber
//refer fig.13.3
h=2000  //m
u=(600*1000)/(60*60)  //m/sec
//initial velocity in vertical direction
//gravitational acceleration=9.81 m/sec^2
//if t is the time of flight
t=sqrt((2000*2)/(9.81))  //sec
//during this period horizontal distance travelled
   by the bomb must be  (d)
d=u*t  //m
printf("Bomb should be released at %.2f m from the
   target",d)
```

Scilab code Exa 13.2 Person jumping over ditch

```
//Person jumping over ditch
//refer fig. 13.4
h=2  //m
Range=3  //m
//let t be the time of flight and u the minimum
   horizontal velocity required
//consider vertical motion
t=sqrt((2*2)/(9.81))  //sec
//consider horizontal motion of uniform velocity
u=3/0.6386  //m/sec
printf("Person should jump with u=%.2f m/sec",u)
```

Scilab code Exa 13.3 Pressure tank

```
//Pressure tank
//refer fig. 13.5
//Required velocity to enter at B
h=1  //m
//If t1 is the time of flight , considering vertical
    motion
t1=sqrt(2/9.81)  //sec
//Considering horizontal motion
u1=3/t1  //m/sec
//Required velocity to enter at C
//let t2  be the time required for flight from A to
   C
bh=2.5  //m
Range=3  //m
//Considering Vertical motion
t2=sqrt((2*2.5)/9.81)  //sec
//Considering horizontal motion
u2=3/t2  //m/sec
printf("The range of velocity for which the jet can
```

```
enter the opening BC is %.2f m/sec to %.2f m/sec"
,u2,u1)
```

Scilab code Exa 13.4 Rocket released from fighter jet

```
//Rocket released from fighter jet
//refer fig. 13.6
h=3000  //m
//If t is time of flight then
//using equations of motion
t=sqrt((2*3000)/(9.81))  //sec
u=(1200*1000)/(60*60)  //m/sec
//Horizontal distance covered during the time of
   flight=range
a=6  //m/sec^2
Range=u*t+(1/2)*a*(t^2)  //m
//Angle theta below the horizontal at which the
   pilot must see the target while releasing the
   rocket is
theta=atand(3000/10078.5)  //degree
printf("Angle theta below the horizontal at which
   the pilot must see the target while releasing the
    rocket is=%.3f degree",theta)
```

Scilab code Exa 13.5 Body is projected

```
//Body is projected
//u be the velocity of projection and alpha the
   angle of projection
```

```
//then maximum height reached=((u^2)*(sind(alpha))
   ^2)/(2*g)
//range=((u^2)*sind(2*alpha))/(g)
//in this case
//Range=3*maximum height reached
//thus
alpha=atand(4/3)  //degree
printf("\n alpha=%.2f degree",alpha)
```

Scilab code Exa 13.6 Projectile aimed at target

```
//Projectile aimed at target
//refer fig. 13.9
//let s be the distance of the target from the point
   of projection
//u be the velocity of projection
//range
//R=((u^2)*sind(2*alpha))/(g)
//applying it to first case
//s-12=(u^2)/(2*g)
//from second case
//s+24=(u^2)/(g)
//solving we get
s=24+24  //m
//let the correct angle of projection be alpha, then
//sind(2*alpha)=48/72
alpha=41.81/2  //degree
printf("Angle of projection=%.3f degree",alpha)
```

Scilab code Exa 13.7 Projectile

```
//Projectile
//let u be the initial velocity and alpha its angle of projection
//Vertical component of velocity=u*sind(alpha)
//Horizontal component of velocity=u*cosd(alpha)
//thus according to given condition
alpha=atand(1/2)  //degree
//when x=18 m y=3 m
//using equation of trajectory
u=sqrt((9.81*(18^2))/(6*2*(cosd(26.565))^2))  //m/sec
g=9.81  //m/sec
//range on the horizontal plane (range)
range=((u^2)*sind(2*alpha))/(g)  //m
printf("Range on the horizontal plane=%.2f m",range)
```

Scilab code Exa 13.8 Find the least initial velocity

```
//Find the least initial velocity
//refer fig. 13.10
//Let u the initial velocity required and alpha the angle of projection
//here
range=9  //m
//at P x=5m and y=4m
//u^2=(9*g)/(sind(2*alpha))
//from the equation of trajectory
alpha=atand(1.8)  //degree
//thus
u=sqrt((9*9.81)/(sind(2*60.95)))  //m/sec
printf("u=%.2f m/sec",u)
```

Scilab code Exa 13.9 Bullet fired

```
//Bullet fired
//refer fig. 13.13
//velocity of projection
u=(360*1000)/(60*60)  //m/sec
//(a) total time of flight
//method 1
y0=-120  //m
//considering vertical motion and solving quadratic
   equation
t=12.20  //sec
//method 2
//t1=(100*sind(30))/(9.81)  //sec
//maximum height reached in this time
//h=((100^2)*(sind(30))^2)/(2*9.81)  //m
//during downward motion
//t2=7.1  //sec
//t=t1+t2  //sec
//method 3
//time required to travel from A to D
//t1=10.19  //sec
//g=9.81  //m/sec^2
//distance travelled=120 m
t=12.20 //sec
//(b) Maximum height reached by the bullet
//h=((100^2)*(sind(30))^2)/(2*9.81) m above point A
h=127.42+120  //m above the ground
//(c)Horizontal range
Hrange=100*12.2*cosd(30)  //m
//(d)Velocity of the bullet just before striking the
    ground
```

```
//vertical component of velocity=69.682 m/sec
//horizontal component of velocity=86.603 m/sec
//velocity at strike
v=sqrt((69.682^2)+(86.603^2))  //m/sec
theta=atand(69.682/86.603)  //degree
printf("\nt=%.2f sec\nh=%.2f m above the ground\nHorizontal range=%.2f m\nv=%.2f m/sec\ntheta=%.2f degree",t,h,Hrange,v,theta)
```

Scilab code Exa 13.10 cricket ball

```
//cricket ball
//refer fig. 13.15
u=20  //m/sec
alpha=30  //degree
Y0=-1.5  //m
t=2.179  //sec
//Distance of the fielder from the wickets
Range=u*t*cosd(alpha)  //m
printf("The distance of the fielder from the wickets=%.3f m",Range)
```

Scilab code Exa 13.11 Gravel is thrown in bin

```
//Gravel is thrown in bin
//refer fig. 13.16 and 13.17
//taking O as origin
u=5  //m/sec
alpha=50  //degree
```

```
//for point B
y=-10  //m
t=1.871  //sec
//Horizontal distance travelled in this time=6.012 m
//Vertical component of velocity of gravel at the time of striking the bin is=14.524 m/sec (downwards)
//Horizontal component of velocity=5*cosd(50)  m/sec
//Velocity of strike
v=sqrt((14.524^2)+(3.214^2))  //m/sec
theta=atand(14.524/3.214)  //degree to the horizontal
printf("\nt=%.2f sec\nHorizontal distance travelled =6.012 m\nv=%.2f m/sec\ntheta=%.2f degree to horizontal",t,v,theta)
```

Scilab code Exa 13.12 soldier fires a bullet

```
//soldier fires a bullet
//refer fig 13.18
//equation of trajectory of bullet is known thus
//For the point on ground where bullet strikes
y=-50  //m
x=100 //m
u=31.32  //m/sec
//alpha=0 or
alpha=atand(2)  //degree
//when alpha =0
//Horizontal component of velocity
vx=31.32  //m/sec
//Vertical component of velocity
vy=sqrt(2*9.81*50)  //m/sec
//Velocity of strike
```

```
v=sqrt((31.32^2)+(31.32^2))  //m/sec
theta=atand(1)  //degree
//when alpha=63.435 degree   vx=14.007 m/sec
//vy=42.02 m/sec
bv=sqrt((14.007^2)+(42.02^2))  //m/sec
btheta= atand(42.02/14.007)  //degree to horizontal
printf("\nalpha=%.2f degree\nv=%.2f m/sec\ntheta=%.2
   f degree\nv=%.2f m/sec\ntheta=%.2f degree to
   horizontal",alpha,v,theta,bv,btheta)
```

Scilab code Exa 13.13 a rebounding ball

```
//a rebounding ball
//refer fig.13.19
//at A the vertical component of velocity =u*asind(
   alpha)
//when h=19 m Vertical component of velocity =0
//y-coordinate of B=-24.033m
//considering the motion in vertically upward
   direction
t=4.93  //sec
//x-coordinate of B=72.1 m
//considering the horizontal motion of the ball
alpha=atand(19.308/14.625)  //degree
u=14.625/cosd(52.86)  //m/sec
printf("\nalpha=%.2f degree\nu=%.2f m/sec",alpha,u)
```

Scilab code Exa 13.14 Flying bomber

```
//Flying bomber
//refer fig. 13.20
h=2400  //m
//Let the time required for bomb to reach ground be t seconds
//then
t=sqrt((2400*2)/(9.81))  //sec
u=(1000*1000)/(60*60)  //m/sec
//Horizontal distance moved by bomb d
d=u*t  //m
//muzzle velocity=600 m/sec
//velocity of projection u=600 m/sec
//alpha=60 degree
//shell has to hit the bomber at height h=2400 m
//let time required for the shell to rise to this height be t1
//then
//t1=110.370 sec or 4.433 sec
//when t1=110.370 sec
//horizontal distance moved by the shell=600*cosd(60)*110.370 m
//distance moved by plane during this period =30658.58 m
//the gun must fire the shell when the bomber is at a distance=33111+30658.58  m
//when t1=4.839 sec
//horizontal distance moved by the shell=1331.70 m
//distance moved by plane during this period=1233.07 m
//the gun must fire the shell when the bomber is at a distance=2564.77 m
printf("\nThe bomb should be released when the bomber is %.2f m away from the target",d)
printf("\nWhen the shell is fired at a distance of 63769.58 m, it will hit the plane in its downward motion.")
printf("\nIf the shell is fired when the bomber is at a distance of 2795.87 m, then it will hit the
```

```
bomber during its upward motion")
```

Scilab code Exa 13.15 A plane

```
//A plane
//initial velocity
u=200  //m/sec
//angle of projection
alpha=30  //degree
//Inclination of the plane=atand(5/12) degree
//(a) When the shot is fired up the plane
abeta=22.62  //degree
aRange=((200^2)/(9.81*(cosd(22.62))^2))*((sind(2*30-22.62))-(sind(22.62)))  //m
//(b) When the shot is fired down the plane
bbeta=-22.62  //degree
bRange=((200*200)/(9.81*(cosd(22.62))^2))*(sind(82.62)+sind(22.62))  //m
printf("\nWhen the shot is fired up the plane\nRange=%.2f m\nWhen the shot is fired down the plane\nRange=%.2f m",aRange,bRange)
```

Scilab code Exa 13.16 person throws a ball

```
//person throws a ball
//refer fig. 13.23
//(a) Up the plane
atheta=35  //degree
aalpha=atheta+20  //degree
```

```
//maximum range
aRangemax=((30*30)/(9.81*(cosd(20))^2))*(sind
    (2*55-20)-sind(20))  //m
//(b) Down the plane
//refer fig. 13.24
btheta=(90+20)/2  //degree
balpha=55-20  //degree
//maximum range
bRangemax=((30*30)/(9.81*(cosd(-20))^2))*(sind
    (2*35+20)-sind(-20))  //m
printf("\nUp the plane\nMax Range=%.3f m",aRangemax)
printf("\nDown the plane\nMax Range=%.3f m",
    bRangemax)
```

Chapter 14

Relative velocity

Scilab code Exa 14.1 Passenger train

```
//Passenger train
//refer fig. 14.7
//let vb= velocity of goods train
vA=(72*1000)/(60*60)  //m/sec
//relative velocity of B w.r.t A
//vB/A=20-vB
//relative distance moved to overtake the goods train=250+200 m
//45 seconds are required to cover this relative distance
vB=(10*60*60)/(1000)  //kmph
printf("\nvB=%.2f kmph",vB)
```

Scilab code Exa 14.2 passenger train

```
//passenger train
//velocity
```

```
vA=20 //m/sec
//let velocity of goods train be vB m/sec
//relative velocity=20-vB m/sec
//when t=25 relative distance moved is x metres
//(20-vB)*25=x
//In the next t=30 seconds
//relative distance moved=length of passenger train =240 m
vB=(20-(240/30))*((60*60)/(1000)) //km/h
x=(20-12)*25 //m
printf("\nLength is %.2f m\nSpeed is %.2f km/h",x,vB)
```

Scilab code Exa 14.3 Two trains

```
//Two trains
//Taking the direction of motion of train A as positive
//let velocity of A be v m/sec
//Relative velocity=1.5*v
vA=20*((60*60)/(1000)) //kmph
//velocity of B
vB=-10*((60*60)/(1000)) //kmph
printf("\nVelocity of A=%.2f kmph\nVelocity of B=%.2f kmph",vA,-vB)
```

Scilab code Exa 14.4 Two ships

```
//Two ships
```

```
//refer fig.14.8,14.9 and 14.10
//Taking west direction as x-axis and north
   direction as y-axis
//velocities in kmph are
vAx=30*sind(30)
vAy=30*cosd(30)
vBx=40*sind(45)
vBy=-40*sind(45)
vrx=15-28.284
vry=25.98-(-28.284)
vr=sqrt((13.284^2)+(54.264^2))
theta=atand((13.284)/(54.264))  //degree
printf("\nFrom B, ship A appears to move with a
   velocity of %.2f kmph in N %.2f degree E
   direction",vr,theta)
//relative distance after half an hour (drel)
drel=55.866*(1/2)  //km
printf("\nRelative distance after half an hour=%.2f
  km",drel)
```

Scilab code Exa 14.5 Enemy ship location

```
//Enemy ship location
//refer fig. 14.10
//taking north as y direction and west as x
   direction
//vAy=36*cosd(theta)
//vAx=36*sind(theta)
//Components of velocity of enemy ship
vBy=18*cosd(30)  //kmph
vBx=-18*sind(30)  //kmph
//then
//vrx=36*sind(theta)+9
```

```
//vry=36*cosd(theta)-15.588
//solving
x=0.2777
theta=16.12  //degree
printf("\nWar ship must move in N %.2f W direction",theta)
vrx=36*sind(theta)+9
vry=36*cosd(theta)-15.588
vr=sqrt((19^2)+(19^2))  //kmph
//relative distance moved
dr=25-5  //km
//time interval
t=20/26.870  //hour
printf("\n%.2f hour after sighting the enemy ship the shell is to be fired",t)
```

Scilab code Exa 14.6 Motor boat crossing river

```
//Motor boat crossing river
//refer fig. 14.11 and 14.12
//let the motor boat start from A and reaches C
vrx=15  //kmph
//distance to be moved in x direction=1 km
//time required t is
t=4  //min
//boat will move down the stream
vy=5  //kmph
//Distance moved in downstream direction (d)
d=333.33  //m
//Let the direction of boat be set at theta to x-direction
theta=asind(1/3)  //degree
printf("The boat should be set in the direction
```

```
theta=%.2f degree and time required is t=%.2f min
\nDistance moved in downstream direction=%.2f m",
theta,t,d)
```

Scilab code Exa 14.7 ship approaching port

```
//ship approaching port
//refer fig. 14.13 and 14.14
//let west be x and north be y axes
//speed in kmph is
vBx=25*sind(45)
vBy=25*cosd(45)
vAx=-15
vAy=0
//Let vr be the relative velocity of B w.r.t. A
vrx=17.678-(-15)  //kmph
vry=17.678    //kmph
vr=sqrt((32.678^2)+(17.678^2))  //kmph
alpha=atand(17.678/32.678)  //degree
t=(50*cosd(alpha))/(vr)  //hours
//during this time A has moved in east by (da)
da=15*1.1837  //km
//and B has moved in N 45 degree W a distance (db)
db=25*1.1837  //km
printf("\nt=%.2f hours\nA has moved in east by da=%
    .2f km\nB has moved in N 45 degree W a distance
    db=%.2f km",t,da,db)
```

Scilab code Exa 14.8 ship B approaching port

```
//ship B approaching port
//refer fig. 14.15 and 14.16
//Considering west as x-axis and south as y-axis
vAx=24*cosd(30)
vAy=24*sind(30)
vBx=-18
vBy=0
//Let relative velocity of A w.r.t. B be vr at an
   angle alpha to western direction
vrx=vAx-vBx  //kmph
vry=vAy-vBy
v=sqrt((vrx)^2+(vry)^2)  //kmph
alpha=atand(vry/vrx)  //degree
//Holding B stationary and allowing A to move with
   relative velovity ,BC is given by
BC=60*sind(alpha)  //km
//from triangle BCD
DC=sqrt((25^2)+(17.735)^2)  //km
CE=DC  //km
AC=60*cosd(alpha)  //km
AD=AC-DC  //km
AE=AC+CE  //km
//Time taken to reach D
t1=39699*60/40899  //min
//time taken to reach E
t2=74939*60/40599  //min
printf("\nThe two ships can start exchanging signals
    %.2f min after ship A leaves the port and
   continue to do so for %.2f min",t1,t2-t1)
```

Scilab code Exa 14.9 passenger observing rain drops

```
//passenger observing rain drops
```

```
//refer fig. 14.17
//Let the true velocity of rain be v kmph at a true
   angle theta with vertical
//Taking the direction of train as x and that of
   vertical downward as y
//Velocity components of train are
//v1x=v*sind(theta)
//v1y=v*cosd(theta)
//when the velocity of train was 36 kmph
v2x=36
v2y=0
//alpha is the direction of relative velocity and is
    given as 30 degree and when the velocity of
   train is 54 kmph alpha=45 degree thus
//v*cosd(theta)=v*sind(theta)-54
//v*sind(theta)=-11.402
//solving we get
v=sqrt(4407.43)  //kmph
theta=asind(-(11.402)/(66.388))
printf("\nv=%.3f kmph\ntheta=%.2f degree",v,theta)
```

Scilab code Exa 14.10 Jet of water

```
//Jet of water
//refer fig. 14.18 and 14.19
//time taken to move a horizontal distance of 5m
t=5/20  //sec
//During this period vertical downward velocity
   gained by water (Vw)
Vw=0+(9.81/4)
//Horizontal component of velocity of plate (HCp)
HCp=0
//Vertical component of velocity of plate (VCp)
```

```
VCp=1   //m/sec
//relative velocity of water w.r.t. plate
vry=Vw-VCp   //m/sec
vrx=20   //m/sec
vr=sqrt((20)^2+(1.453)^2)   //m/sec
alpha=atand(1.453/20)   //degree
printf("\nvr=%.2f m/sec\nalpha=%.2f degree",vr,alpha
    )
```

Scilab code Exa 14.11 railway carriage

```
//railway carriage
//refer fig. 14.20
//Velocity of components of train are
v1x=96   //kmph
v1y=0
//Velocity components of bullet are
//v2x=0.9848*v
//v2y=0.1736*v
//Component of relative velocity of B w.r.t. A are
//vrx=0.9848*v-96
//vry=0.1736*v
//Direction of relative velocity alpha with x-axis
//tand(alpha)=((vry)/(vrx))
//thus
v=183.96   //kmph
//consider motion in y direction
vry=0.1736*51.1   //m/sec
//Time period (t)
t=(1.8)/(0.1736*51.1)   //sec
printf("\nv=%.2f kmph\nt=%.3f sec",v,t)
```

Chapter 15

DAlemberts principle

Scilab code Exa 15.2 Elevator cage

```
//Elevator cage
//refer fig. 15.4
u=0
v=25  //m/sec
s=187.5  //m
//using equations of motion
a=(25^2)/(2*187.5)  //m/sec^2
//summing up the forces in vertical direction
T=8600-((8600*1.667)/(9.81)) //N
//Equilibrium condition gives
R=600-((600*1.667)/(9.81)) //N
printf("\nT=%.2f N\nR=%.2f N",T,R)
```

Scilab code Exa 15.3 Motorist travelling

```
//Motorist travelling
//refer fig. 15.5
```

```
u=(70*1000)/(60*60)  //m/sec
v=0
s=50  //m
//Using equation of linear motion
a=-(19.44^2)/(2*50)  //m/sec^2
//again
t=19.44/3.78  //sec
//Applying equilibrium equationswe get
mu=(3.78)/(9.81)
printf("\nt=%.2f sec\nmu=%.3f ",t,mu)
```

Scilab code Exa 15.4 `block on horizontal plane`

```

//block on horizontal plane
//refer fig. 15.6 (a) and (b)
//Inertia force of block m*a=3/9.81 kN
//applying equilibrium conditions
//N=1+P/2
//P*cosd(30)-F-3/9.81
//From law of friction
//F=mu*N
//Solving above equations
P=0.561  //kN
printf("\nP=%.3f kN",P)
```

Scilab code Exa 15.5 `Crate resting on cart`

```
//Crate resting on cart
```

```
//refer fig. 15.7 (a),(b) and (c)
//Applying equilibrium condition
//N=W=750 N
//Frictional force
mu=0.3
N=750
F=mu*N
a=(225*9.81)/(750)  //m/sec^2
//Consider dynamic equilibrium of the system
P=250+((1250*2.943)/(9.81))  //N
printf("\nMaximum allowable P=%.2f N and a=%.3f m/
   sec^2",P,a)
```

Scilab code Exa 15.6 Body on an inclined plane

```
//Body on an inclined plane
//Refer fig. 15.8 (a),(b) and (c)
//Consider 1200 N block
//applying equilibrium condition
N=1200*cosd(12)  //N
mu=0.2
//From Law of friction
F=mu*N
//applying equilibrium condition
a=(800-484.25)/(122.32+((800)/(9.81)))  //m/sec^2
//solving for T
T=800-((800*1.549)/(9.81))  //N
//initial velocity=0
t=3  //sec
//distance moved in 3 sec
s=0*3+((1.549*3^2)/(2))  //m
printf("\na=%.3f m/sec^2\nT=%.2f N\ns=%.3f m",a,T,s)
```

Scilab code Exa 15.7 `Two weights connected by weight`

```
//Two weights connected by weight
//refer fig.15.9 (a) and (b)
//Consider dynamic equilibrium of 200 N Weight
N1=200  //N
mu=0.3
//From law of friction
F1=mu*N1  //N
//applying equilibrium condition
//T1-(200*a)/9.81=60
//Consider 800N body
N2=800  //N
//From Law of friction
F2=mu*N2  //N
//applying equilibrium condition
//T+(800*a)/9.81=160 N
//Solving
a=((160-60)*9.81)/(200+800)  //m/sec^2
T=160-((800*a)/(9.81))  //N
printf("\na=%.3f m/sec^2\nT=%.2f N",a,T)
```

Scilab code Exa 15.8 `Two incloned planes`

```
//Two incloned planes
//refer fig.15.10 (a),(b) and (c)
//Let the assembly move down the 60 degree plane by
   an acceleration a m/sec^2
```

```
//Consider the block weighing 100 N
//Applying equilibrium conditions
N1=50  //N
mu=1/3
//From law of friction
F1=mu*N1  //N
//T+((100*a)/(9.81))=69.93
//Now consider 50 N block
N2=50*cosd(30)  //N
//From the law of friction
F2=mu*N2
//((50*a)/(9.81))-T=-39.43
//Solving we get
a=(69.93-39.43)*9.81/(100+50)  //m/sec^2
T=69.93-(100*1.9947/9.81) //N
printf("\na=%.4f m/sec^2\nT=%.2f N",a,T)
```

Scilab code Exa 15.9 Two blocks on an inclined plane

```
//Two blocks on an inclined plane
//refer fig. 15.11 (a,b,c) and (d)
//Let block A move with an acceleration a1 and block
    B with an acceleration a2
//Consider block A
//Using equilibrium conditions
//NA=WA*cosd(30)
mu1=0.2
WA=100  //N
//From the law of friction
FA=mu1*WA*cosd(30)  //
a1=3.2058  //m/sec^2
//Consider block B
//NB=WB*cosd(30)
```

```
mu2=0.4
//From law of friction
//FB=mu2*WB*cosd(30)
a2=1.5067  //m/sec^2
//Let t be the time elapsed until the blocks touch each other
//displacement of block A in this period be s1
//displacement of block B in this period be s2
//when the two blocks touch each other
//s1=s2+18
//thus
t=4.60  //sec
//After the blocks touch each other
a=2.45  //m/sec^2
P=100*sind(30)-(0.2*100*cosd(30))-((100*2.45)/(9.81))  //N
printf("\nt=%.2f sec\nP=%.1f N",t,P)
```

Scilab code Exa 15.10 Two bodies hung to the rope ends

```
//Two bodies hung to the rope ends
//refer fig. 15.12 (a),(b) and (c)
//Let a be the acceleration with which the system moves and T be the tension in the string
//Considering 300 N body
//T-(300*a)/(9.81)=300
//Considering 450 N body
//T+(450*a)/(9.81)=450
//solving we get
a=(450-300)*9.81/(450+300)  //m/sec^2
T=300+((300*1.962)/(9.81)) //N
printf("\na=%.4f m/sec^2\nT=%.0f N",a,T)
```

Scilab code Exa 15.11 Tension in the string and accelerations of blocks

```
//Tension in the string and accelerations of blocks
//refer fig. 15.13 (a),(b) and (c)
//Considering 1500 N block
//2*T+(1500*a)/(9.81)=1500
//Considering 500N block
//T-(2*500*a)/(9.81)=500
//Solving this we get
a=(500*9.81)/(1500+2000)  //m/sec^2
T=(1500-((1500*1.401)/(9.81)))/2  //N
printf("\na=%.3f m/sec^2\nT=%.2f N",a,T)
```

Scilab code Exa 15.12 Train along an inclined plane

```
//Train along an inclined plane
//refer fig. 15.14 and 15.15
u=0
v=(36*1000)/(60*60)  //m/sec^2
s=1000  //m
//From kinematic equation
a=100/2000 //m/sec^2
//Tractive resistance (Tr)
Tr=5*1500  //N
//Component of weight of train (Wt1)
Wt=1500/100  //kN
//Inertia force (I1)
I=(1500*0.05)/(9.81)  //kN  (Down the plane)
```

```
//Dynamic equilibrium equation gives
T=7.5+15+7.645 //kN
//Consider dynamic equilibrium of train
//Total tractive resistance (Rt)
Rt=5*2000 //N
//Inertia force (I2)
I2=(2000*0.05)/(9.81) //kN (Down the plane)
//Component of weight down the plane (Wt2)
Wt2=(2000)/(100) //kN
//Dynamic equilibrium equation gives
P=10+10.194+20 //kN
printf("\nT=%.3f kN\nP=%.3f kN",T,P)
```

Chapter 16

Work energy method

Scilab code Exa 16.1 Pump

```
//Pump
//Work done in lifting 40 m^3 of water to a height of 50 m (W1)
W1=40*9810*50  //N-m
//Kinetic energy at delivery KE1
KE1=(40*9810*25)/(2*9.81)  //N-m
//Total energy spent (TE)
TE=19620000+500000  //N-m
//This energy is spent by the pump in half an hour
//Pump output power (PO)
PO=(20120000)/(1800*1000)  //kW
//Input power (Ip)
Ip=PO/0.7  //kW
printf("\Energy spent=%.2f N-m\nInput power=%.4f kW",TE,Ip)
```

Scilab code Exa 16.2 Man and his wish

```
//Man and his wish
//refer fig. 16.4 (a),(b)
//Work done in sliding
N=1 //kN
W=N //kN
mu=0.3
F=mu*N //kN
//Applied force
P=F //kN
//Work to be done in sliding to a distance of 5 m (
    W1)
W1=0.3*5 //kJ
//Work to be done in tipping
//Height (h)
h=(1/sqrt(2))-0.5 //m
//Work done in one tipping (W2)
W2=W*h //kJ
//To move a distance of 5m, Five tippings are
    required
//Hence
W3=5*W2 //kJ
printf("\nThe man needs to spend only %.2f kJ while
    tipping and it is less than %.2f kJ spent in
    sliding\nHe should move the box by tipping",W3,W1
    )
```

Scilab code Exa 16.3 body pushed up the plane

```
//body pushed up the plane
//refer fig. 16.6
//applying equilibrium condition
N=300*cosd(30) //N
mu=0.2
```

```
//Frictional force
F=mu*N  //N
//initial velocity
u=1.5  //m/sec
//displacement
s=6  //m
//let final velocity be v m/sec
//Equating work done by forces along the plane to
   change in K.E
v=sqrt(77.71+2.25)  //m/sec
printf("After moving 6 m the body will have velocity
    v=%.4f m/sec",v)
```

Scilab code Exa 16.4 Power of a locomotive

```
//Power of a locomotive
//refer fig.16.7 (a) and (b)
v=(56*1000)/(60*60)  //m/sec
F=5*420/1000  //kN
W=420  //kN
P=F+W*(1/120)  //kN
//Power of Locomotive Pw
Pw=P*v  //kW (mistake in book)
u=15.556  //m/sec
//Resultant force parallel to the plane R
Res=F+W*(1/120)  //kN (Down the plane)
s=((420*(15.556^2))/(2*9.81*5.6))  //m
printf("Power of locomotive=%.4f kW\ns=%.4f m",Pw,s)
//The answers vary due to round off error
```

Scilab code Exa 16.5 A tram car

```
//A tram car
//refer fig.16.8 (a),(b) and (c)
//frictional resistance
W=120  //kN
F=5*120/1000  //kN
v=(20*1000)/(60*60)  //m/sec
// (1) on level track
P1=F  //kN
//output power Pw1
Pw1=P1*v  //kW
eta1=0.8
//input power Ip1
Ip1=Pw1/0.8  //kW
// (2) Up the plane
P2=F+W*(1/300)  //kN
//output power required Pw2
Pw2=P2*v  //kW
//Input power of engine Ip2
Ip2=Pw2/0.8  //kW
// (3) Down the incline plane
Pd=F-W*(1/300)
Pwd=0.2*5.5556  //kW
//Input power
Ipd=1.1111/0.8  //kW
printf("\On level track Input Power=%.3f kW\nUp the
    plane Input Power=%.3f kW\nDown the incline plane
    Input Power=%.3f kW",Ip1,Ip2,Ipd)
```

Scilab code Exa 16.6 Police investigation

```
//Police investigation
//refer fig. 16.9
//Let the probable speed of the car just before
   brakes are applied be u m/sec
//F=0.5*W
//Final velocity=0
s=60  //m
//applying work energy equation
u=((sqrt(0.5*60*2*9.81))*60*60)/1000  //m/sec
printf("\nThe probable speed of the car just before
   brakes are applied is %.2f kmph",u)
```

Scilab code Exa 16.7 Block being pulled

```
//Block being pulled
//refer fig. 16.10 (a) and (b)
//when pull P is acting
W=2500  //N
P=1000  //N
N=W-P*sind(30)
mu=0.2
F=mu*N  //N
//Initial velocity=0
//Let final velocity be v
s=30  //m
```

```
//Applying work energy equation for the horizontal motion
v=sqrt((0.866*1000-400)*30*2*9.81/2500)
printf("\nv=%.3f m/sec",v)
//Now if the 1000 N force is removed,let the distance moved before rest be s
//Initial velocity=10.4745  //m/sec
//Final velocity=0
s=(2500*(10.4745^2))/(400*2*9.81)  //m
printf("\ns=%.3f m",s)
//The answer provided in the textbook is wrong
```

Scilab code Exa 16.8 Small block sliding down the plane

```
//Small block sliding down the plane
//refer fig. 16.11 (a),(b) and (c)
//Length AB
AB=sqrt((3^2)+(4^2))
//Consider FBD of the block on inclined plane A
//It moves down the plane, hence
//N1=W*0.8
mu=0.3
//F1=0.3*W
//Applying work energy equation for the motion from A to B
vB=sqrt((0.6-0.24)*5*2*9.81)  //m/sec
//For the motion on horizontal plane
//final velocity=0
//Writing work energy equation for the motion along BC
s=(5.943^2)/(2*9.81*0.3)  //m
printf("\ns=%.2f m",s)
```

Scilab code Exa 16.9 Force P required

```
//Force P required
//refer fig. 16.13 (a),(b)
//The system of forces acting on connecting bodies
   is shown in figure
N1=250  //N
mu=0.3
F1=mu*N1  //N
N2=(1000*3)/(5)  //N
F2=0.3*N2  //N
N3=500  //N
F3=mu*N3  //N
//Let the constant force be P
//writing work energy equation
P=((250+1000+500)*3*3/(2*9.81*4.5))
   +75+180+1000*0.8+150  //N
printf("\nThus P=%.3f N",P)
```

Scilab code Exa 16.10 Body A

```
//Body A
//refer fig. 16.14
mu=0.2
//let theta1 and theta2 be the slopes of the
   inclined planes
//sind(thets1)=4/5 cosd(theta1)=0.6
//sind(theta2)=3/5 cosd(theta2)=0.8
```

```
//1500*sind(theta1)=1200 N down the plane
F1=mu*1500*0.6   //N up the plane
F2=0.2*2000*0.8   //N down the plane
//Equating work done to change in kinetic energy
//v=3 m/sec
s=((1500*3*3+2000*1.5*1.5)/(2*9.81*260))   //m
printf("\nThus s=%.3f m",s)
```

Scilab code Exa 16.11 Two bodies hung to rope

```
//Two bodies hung to rope
//refer fig. 16.15 (a) and (b)
s=(450+300)*(4*4-2*2)/(2*9.81*150)   //m
//Let T be the tension in the string
//apply work energy principle
T=((450*3.058)-((450*12)/(2*9.81)))/3.058   //N
printf("\nT=%.0f N\ns=%.3f m",T,s)
```

Scilab code Exa 16.12 Block slides down a plane

```
//Block slides down a plane
//refer fig. 16.17 and 16.18
N=3000*cosd(50)   //N
mu=0.2
F=mu*N   //N
//let the maximum deformation of the spring be s mm
//then
s=721.43   //mm
```

```
//Velocity will be maximum when the acceleration is zero
//Let x be the deformation when net force on the body in the direction of motion is zero
x=(3000*sind(50)-385.67)/(20)  //mm
//applying work energy principle
v=5.061  //m/sec
printf("\nv=%.3f m/sec\ns=%.3f mm",v,s)
```

Scilab code Exa 16.13 Wagon strikes bumper post

```
//Wagon strikes bumper post
//refer fig. 16.19
W=500
//Component of weight down the plane Wd
Wd=W/100  //kN
//Track resistance Rt
Rt=2.5  //kN
//u=0
s=30  //m
//Let the velocity of wagon while striking be v m/sec
//Applying work energy equation
v=1.716  //m/sec
//Let spring compresion be x
k=15000  //kN/m
//Applying work energy equation and solving quadratic equation
x=100.2  //mm
printf("\nThe spring will be compressed by %.1f mm",x)
```

Chapter 17

Impulse momentum

Scilab code Exa 17.1 glass marble

```
//glass marble
//refer fig. 17.1
g=9.81  //m/sec^2
//applying kinematic equations
//velocity with which marble strikes the floor Vm
Vm=sqrt(2*g*10)  //m/sec (downward)
//applying kinematic equations
//Velocity of rebound Vr
Vr=sqrt(2*g*8)  //m/sec  (upward)
//Taking upward direction as positive and applying
   impulse momentum equation
//Impulse I
I=(0.2*(12.52+14.007))/9.81  //N-sec
//average force F
F=0.541*10 //N
printf("\nImpulse=%.3f N-sec\nF=%.2f N",I,F)
```

Scilab code Exa 17.2 batsman

```
//batsman
//refer fig 17.2 (a) and (b)
//Let Fx be the horizontal component and Fy be the
   vertical component
//Applying impulse momentum equation in horizontal
   direction
Fx=(48*cosd(30)+20)/(9.81*0.02)  //N
//Applying impulse momentum equation in vertical
   direction
Fy=(48*sind(30))/(9.81*0.02)   //N
//Resultant force
F=sqrt(((Fx)^2)+((Fy)^2))  //N
theta=atand(Fy/Fx)  //degree
printf("\nF=%.3f N\ntheta=%.3f degree",F,theta)
```

Scilab code Exa 17.3 Block in contact with level plane

```
//Block in contact with level plane
//refer fig. 17.3
//Normal reaction
N=1500  //N
mu=0.1
F=mu*N  //N
//Applying impulse momentum equation in the
   horizontal direction
t=(1500*(16-0))/(9.81*(300-150))  //sec
//If force is then removed, the only horizontal
   force is F=150 N
//Applying impulse momentum equation
t1=-(1500*(0-16))/(9.81*(300-150))  //sec
printf("\nThe block takes %.3f sec before comming to
```

```
    rest",t)
```

Scilab code Exa 17.4 Automobile moving

```
//Automobile moving
//refer fig. 17.4
//initial velocity
u=19.44   //m/sec
//final velocity
v=0
//applying impulse momentum equation
//t=1.982/mu
//on concrete road
t1=1.982/0.75   //sec
//on ice
t2=1.982/0.08   //sec
printf("\nOn concrete road t=%.3f sec\nOn ice t=%.3f
    sec",t1,t2)
```

Scilab code Exa 17.5 Block on inclined plane

```
//Block on inclined plane
//refer fig. 17.5
theta=atand(5/12)   //degree
N=130*cosd(theta)   //N
mu=0.3
F=mu*N   //N
//Force down the plane
R=130*sind(theta)-36   //N
```

```
u=2.4   //m/sec
//v is final velocity
t=5   //sec
//applying impulse momentum equation
v=((14*5*9.81)/(130))+2.4   //m/sec
printf("\nv=%.3f m/sec",v)
```

Scilab code Exa 17.6 moving weight

```
//moving weight
//refer fig.17.6 (a),(b)and(c)
//first method
//For 2000 N block
W1=2000   //N
mu=0.2
N1=W1*cosd(30)   //N
F1=mu*N1   //N
//For 1800N block
W2=1800   //N
N2=W2*cosd(60)   //N
F2=mu*N2   //N
//Let T be the tension in the chord
u=0
v=9.81   //m/sec
//applying impulse momentum equation for the 2000 N
   block in upward direction parallel to the plane
//(T-1346.41)*t=2000
//Applying impulse momentum equation for 1800 N
   block
T=1363.48   //N
//Thus
t=117.11   //sec
printf("\nBy first method-\nT=%.2d N\nt=%.2d sec",T,
```

```
   t)
//second method
//Writing impulse momentum equation in the direction
    of motion
t1=117.11  //sec
//To find tension in the chord, consider impulse
   momentum equation of any block
T1=1363.48  //N
printf("\nBy second method-\nt=%.3f sec\nT=%.3f N",
   t1,T1)
```

Scilab code Exa 17.7 Tensions in the strings

```
//Tensions in the strings
//refer fig. 17.7
//Case (a)- Initial velocity u=0 t=5 sec
//Writing impulse momentum equation for 500 N block
   and 1500 N block and solving obtained equations
v=7.007  //m/sec
T=642.86  //N
//Case (b)-Initial velocity u=3 m/sec
//Writing impulse momentum equation for 500 N block
   and 1500 N block and solving obtained equations
v1=9.15 //m/sec
T1=655.96  //N
printf("\nFor case (a)\nv=%.3f m/sec\nT=%.3f N",v,T)
printf("\nFor case (b)\nv=%.3f m/sec\nT=%.3f N",v1,
   T1)
```

Scilab code Exa 17.8 Frictionless pulleys

```
//Frictionless pulleys
//refer fig. 17.8 (a) and (b)
//consider combined FBD of the system
N1=500 //N
F1=0.2*500 //N
N2=1000*cosd(30) //N
F2=0.2*N2 //N
//writing impulse momentum equation
v=20.19 //m/sec
printf("\nv=%.3f m/sec",v)
```

Scilab code Exa 17.9 Value of P

```
//Value of P
//refer fig. 17.9 (a),(b)and(c)
//Let t1 be the time required to bring the system to
    rest
N=1000 //N
F=0.2*1000 //N
//Applying impulse momentum equation upto stationary
    condition and leftward motion and solving those
   equations by trial and error method we get
P=645.74 //N
printf("Value of P is %.3f N",P)
```

Scilab code Exa 17.10 Nozzle issuing jet of water

```
//Nozzle issuing jet of water
//refer fig. 17.10 and 17.11 (a) and (b)
//Weight of water whose momentum is changed in t
   second is (W)
//W=(%pi*(0.05^2)*30*9810*t)/4 N
Px=236.75  //N
Py=883.58  //N
P=sqrt((Px^2)+(Py^2))  //N
//Inclination with horizontal
theta=atand(Py/Px)  //degree
printf("The force exerted by jet is P=%.3f N\
   nInclination to horizontal=%.3f degree",P,theta)
```

Scilab code Exa 17.11 `Vane is moving`

```
//Vane is moving
//refer fig. 17.12 (a) and (b)
//Velocity of approach Va
Va=20  //m/sec
//Weight of water impinging in t seconds=385.24*t
//Velocity of departure Vd
Vd=30-10  //m/sec
//Writing impulse momentum equation in x direction
Px=105.22  //N
Py=392.70  //N
P=sqrt((Px^2)+(Py^2))  //N
//inclination
theta=atand(Py/Px)  //degree
printf("\nPressure exerted P=%.3f N",P)
```

Scilab code Exa 17.12 `Man moving`

```
//Man moving
//Weight of man
W1=800   //N
v=3   //m/sec
//Weight of system after man jumps into boat
W2=800+3200   //N
//(a) Initial velocity of boat
//using principle of conservation of momentum
v=0.6   //m/sec
//(b) Initial velocity of boat =0.9 m/sec towards
    the pier
//Applying principle of conservation of momentum
v1=-0.12   //m/sec
printf("\nVelocity of boat and man will be %.3f m/
    sec towards the pier",-v1)
```

Scilab code Exa 17.13 `Car running`

```
//Car running
//(1) When three men jump off in succession
u=0
v1=10+((700*5)/(11000+3*700))
v2=v1+((700*5)/(11000+2*700))
v3=v2+((700*5)/(11000+700))   //m/sec
//(2) When three men jump together
v=10+((3*5*700)/(11000+3*700))   //m/sec
```

```
printf("\nWhen three men jump off in succession v=%
    .3f m/sec\nWhen three men jump together v=%.3f m/
    sec",v3,v)
```

Scilab code Exa 17.14 car and lorry

```
//car and lorry
//refer fig. 17.13 (a) and (b)
//Let the velocity of vehicle after collision be vx
    in x direction and vy in y direction
vx=18  //kmph
//applying impulse momentum equation in y-direction
vy=12  //kmph
//Resultant velocity
v=sqrt((vx^2)+(vy^2))  //kmph
//its inclination to main road
theta=atand(vy/vx)  //degree
printf("\nv=%.3f kmph\ntheta=%.3f degree",v,theta)
```

Scilab code Exa 17.15 A gun

```
//A gun
//applying principle of conservation of momentum
v=-5  //m/sec
printf("\nGun will have a velocity of %.2d m/sec in
    the direction opposite to that of bullet",-v)
//Let the gun recoil for a distance s
//Using work energy equation
s=(300*25)/(2*9.81*600)  //m
```

```
//Applying impulse momentum equation to gun
t=(300*5)/(600*9.81)  //sec
printf("\ns=%.3f m\nt=%.3f sec",s,t)
```

Scilab code Exa 17.16 Bullet fired horizontally

```
//Bullet fired horizontally
//refer fig. 17.14
//Let the velocity of the block be u immediately after bullet strikes it
//Applying work energy principle
u=1.025  //m/sec
//Let v be the velocity of the bullet before striking the block
//Principle of conservation of momentum gives
v=342.69  //m/sec
//Initial energy of bullet Ei
Ei=(0.3*342.69^2)/(2*9.81)  //J
//Energy of the block and the bullet system E
E=((100+0.3)*1.025^2)/(9.81*2)  //J
//Loss of energy El
El=1795.68-5.37  //J
printf("\nv=%.3f m/sec\nLoss of energy=%.3f J",v,El)
```

Scilab code Exa 17.17 Bullet

```
//Bullet
//refer fig. 17.15
//Initial momentum of the system=Final momentum
```

```
v=21.31   //m/sec
//Kinetic Energy lost= Initial K.E- Final K.E
loss=((0.5*400^2)/(9.81*2)+(30*15^2)/(2*9.81))
    -((30.5*21.31^2)/(2*9.81))   //J
printf("\nLoss of energy=%.3f J",loss)
```

Scilab code Exa 17.19 A pile hammer

```
//A pile hammer
u=0
h=0.75   //m
g=9.81   //m/sec^2
//at the time of strike
v=sqrt(2*g*h)   //m/sec
//Applying principle of conservation of momentum of
    pile and hammer
V=(20*3.836)/(30)   //m/sec
//Applying work energy equation
R=130000/1000   //kN
printf("\nResistance to penetration of the ground=%
    .3f kN",R)
```

Scilab code Exa 17.20 A pile hammer

```
//A pile hammer
h=0.6   //m
v=sqrt(2*9.81*0.6)   //m/sec
V=(15*3.431)/(22.5)   //m/sec
//Applying work energy equation
```

```
s=(22.5*2.287^2)/(2*9.81*117.5)   //m
printf("\ns=%.3f m",s)
```

Scilab code Exa 17.21 Hammer

```
//Hammer
//refer fig.17.17
//Applying impulse momentum equation
V=4.808   //m/sec
//Applying work energy equation to the system
R=306.3   //N
printf("Resistance of the block=%.3f N",R)
```

Chapter 18

Impact of elastic bodies

Scilab code Exa 18.1 `Direct central impact`

```
//Direct central impact
//Refer fig. 18.3
u1=6  //m/sec
u2=-10  //m/sec
//Principle of conservation of momentum
//2*v1+v2=2
//From the defination of coefficient of restitution
//v2-v1=12.8
//solving
v1=-3.6  //m/sec
v2=12.8-(-v1)  //m/sec
printf("\nv1=%.2f m/sec\nv2=%.2f m/sec",v1,v2)
```

Scilab code Exa 18.2 `Body moving to the right`

```
//Body moving to the right
v2=3  //m/sec
```

```
u2=-10 //m/sec
v2=4 //m/sec
//Applying principles of conservation of momentum
v1=((80*3)-100-40)/(80) //m/sec
//Defination of coeff. of restitution gives
e=(4-1.25)/(3+10)
printf("\nv1=%.3f m/sec\ne=%.3f ",v1,e)
```

Scilab code Exa 18.3 A golf ball

```
//A golf ball
h0=10 //m
//u1=sqrt(2*g*h0)
u2=0
v2=0
//defination of coefficient of restitution gives
//v1=%e*sqrt(2*g*h0) in upward direction
//From kinematic equation
h1=10*0.894^2 //m
//After second bounce
h2=6.388 //m
//After third bounce
h3=5.105 //m
printf("\nh1=%.3f m\nAfter second bounce h2=%.3f m\
   nAfter third bounce h3=%.3f m",h1,h2,h3)
```

Scilab code Exa 18.4 Ball is dropped from height

```
//Ball is dropped from height
```

```
u1=sqrt(2*9.81*1)  //m/sec
v1=-sqrt(2*9.81*0.810)  //m/sec
//There is no movement of the floor before and after striking
//u2=0
//v2=0
//From the defination of coefficient of restitution
e=(3.987/4.429)
//Let the velocity of the ball after second bounce be v2
v2=e*3.987  //m/sec  upward
//Expected height h2
h2=(3.576^2)/(2*9.81)  //m
printf("\nCoefficient of restitution=%.3f \nExpected height of second bounce h2=%.4f m",e,h2)
```

Scilab code Exa 18.5 `Ball in frictionless tube`

```
//Ball in frictionless tube
//refer fif. 18.4
u1=sqrt(2*9.81*2)  //m/sec
u2=0
//By principle of conservation of momentum
//v1+2*v2=6.264
//From defination of coefficient of restitution
//case(1)-e=1
//v2-v1=6.264
//solving
v2=4.176  //m/sec
v1=6.264-(2*4.176)  //m/sec
//Let h be the height to which hanging ball will rise
//Change in K.E=Work Done
```

```
h=(v2^2)/(2*9.81)   //m
//case(2)- e=0.7
//v2-v1=4.385
//solving
bv2=(6.264+4.385)/(3)   //m/sec
//Height to which ball will rise
h2=(bv2^2)/(2*9.81)   //m
printf("\nCase(1)-\nh=%.4f m\nCase(2)-\nh2=%.4f m",
    h,h2)
```

Scilab code Exa 18.6 `Two identical balls`

```
//Two identical balls
//refer fig. 18.5 and 18.6 (a) and (b)
//Before impact
uAY=4.5   //m/sec
uAX=7.794   //m/sec
uBY=10.392   //m/sec
uBX=-6   //m/sec
vAY=uAY   //m/sec
vBY=uBY   //m/sec
//Applying principle of conservation of momentum
//vAX+vBX=1.794
//From defination of coefficient of restitution
//vBX-vAX=12.415
//Solving
vBX=(12.415+1.794)/(2)   //m/sec
vAX=1.794-7.104   //m/sec
vA=sqrt((5.31^2)+(4.5^2))   //m/sec
thetaA=atand(4.5/5.31)   //degree
vB=sqrt((7.104^2)+(10.392^2))   //m/sec
thetaB=atand(10.392/7.104)   //degree
printf("\nvA=%.3f m/sec\nthetaA=%.3f degree\nvB=%.3f
```

```
m/sec\nthetaB=%.3f degree",vA,thetaA,vB,thetaB)
```

Scilab code Exa 18.7 `Ball is dropped`

```
//Ball is dropped
//refer fig. 18.7
//Normal to line of impact
u1x=1.986  //m/sec
//In the line of impact
u1y=-7.411  //m/sec
//Let the velocity after impact be v1
v1x=u1x
//Initial and final velocities of floor=0
//From the defination of coefficient of restitution
v1y=5.929  //m/sec
v1=sqrt((1.986^2)+(5.929^2))
theata=atand(v1x/v1y)  //degree to the line of impact also theta is spelled theata
//Inclination to the plane
I=90-18.52  //degree  The answer provided in the textbook is wrong
printf("\nv1=%.3f m/sec\nInclination to the plane=%.3f degree",v1,I)
```

Scilab code Exa 18.8 `Ball falls vertically`

```
//Ball falls vertically
//refer fig. 18.8
//Velocity of the ball which striking plane=3*g
```

```
//Component of velocity down the plane=3*g*sind(20)
//Component of velocity in the line of impact before striking
//vy=-3*g*cosd(20)
//velocity after the impact after striking plane
//vy=2.4*g*cosd(20)
//Acceleration in the line of impact=-g*cosd(20)
//Using kinematic equation
t=4.8  //sec
//vx=3*g*sind(20)
//Acceleration in this direction=g*sind(20)
//Distance travelled in 4.8 sec
s=(3*9.81*t*sind(20))-((9.81*t*t*sind(20))/(2))  //m The answer provided in the textbook is wrong
printf("\nt=%.4f sec\ns=%.4f m",t,s)
```

Scilab code Exa 18.11 Ball dropped from height

```
//Ball dropped from height
//refer fig. 18.11
g=9.81  //m/sec^2
h0=1.2  //m
uy=sqrt(2*g*h0)  //downward
h1=1  //m
v1y=sqrt(2*9.81*h1)  //upwards
e=sqrt(1/1.2)
//Time of flight
t1=(2*sqrt(2*9.81*1))/(9.81)  //sec
ux=0.4/0.903  //m/sec
//Vertical component of velocity after second bounce
v2y=0.903*sqrt(2*9.81*1)  //m/sec
h2=((v2y)^2)/(2*9.81)  //m
t2=(2*4)/(9.81)  //sec
```

```
//Horizontal range
D2=0.443*0.815  //m
printf("\ne=%.3f \nh2=%.3f m \nD2=%.3f m ",e,h2,D2)
```

Scilab code Exa 18.12 Sphere

```
//Sphere
//refere fig. 18.12
u1=3  //m/sec
u2=0.6  //m/sec
//From principle of conservation of momentum
//v1+5*v2=6
//From the defination of coefficient of restitution
//v2-v1=1.8
//solving
v1=6-1.3*5  //m/sec
//The velocity of first ball is reversed after
   impact
//Loss of K.E.=Initial K.E.-final K.E.
loss=1.07  //joules
printf("\nLoss=%.3f joules",loss)
```

Scilab code Exa 18.13 Loss in KE

```
//Loss in K.E.
uAX=7.79  //m/sec
uBX=-6  //m/sec
vAX=-5.31  //m/sec
vBX=7.104  //m/sec
```

```
//mass of the ball m
m=10/9.81
//Loss of K.E.
loss=(10*((7.79^2)+(6*6)-(5.31*5.31)-(7.104*7.104))
   /(2*9.81)  //J
printf("\nLoss of K.E.=%.3f J",loss)
//The answer provided in the textbook is wrong
```

Chapter 19

Circular motion of rigid bodies

Scilab code Exa 19.1 `Automobile moving on road`

```
//Automobile moving on road
//refer fig. 19.7
v=13.889  //m/sec
//case (1)-When vehicle is at A
CFF1=(25*13.889^2)/(9.81*80)  //kN
//Vertical reaction
R1=25-6.145  //kN
//case (2)-When automobile is at B
CFF2=(25*13.889^2)/(9.81*120)  //kN
R2=25+4.097  //kN
//case (3)-On level track at C
R3=25  //kN
printf("\nWhen vehicle is at A, vertical reaction=%
   .3f kN\nWhen automobile is at B, vertical
   reaction=%.3f kN\nOn level track at C, vertical
   reaction=%.3f kN ",R1,R2,R3)
```

Scilab code Exa 19.2 `Car on road`

```
//Car on road
//refer fig. 19.8
//Consider dynamic equilibrium of car
v=sqrt(0.4*9.81*50)*((60*60)/(1000))  //kmph
//Limiting speed from the consideration of
   preventing overturning
//Taking moment about point of contact of outer
   wheel with road and noting that R1=0 when the
   vehicle is about to overturn
//Limiting speed v=50.42 kmph
//If the vehicle moves with a velocity of 40 kmph
v=11.111  //m/sec
//Taking moment about outer wheel
R1=5.612  //kN
R2=15-R1  //kN
printf("\nLimiting speed v=50.42\nR1=%.3f kN\nR2=%.3
   f kN",R1,R2)
```

Scilab code Exa 19.3 `Angle of banking`

```
//Angle of banking
v=33.33  //m/sec
//If alpha is the angle of banking then
alpha=atand((v^2)/(9.81*200))  //degree
printf("\nalpha=%.3f degree",alpha)
```

Scilab code Exa 19.4 `Vehicle moving round a curve`

```
//Vehicle moving round a curve
r=40  //m
mu=0.4
//(1) On level road, limiting speed from the
    consideration of avoiding skidding
v=sqrt(0.4*9.81*40)  //m/sec
//(2) On a road banked to an inclination of 1 in 10
v1=sqrt((9.81*40*(0.4+0.1))/(1-0.4*0.1))  //m/sec
//If lateral forces are not to be experienced
v3=sqrt(0.1*9.81*40)  //m/sec
printf("\nOn level road v=%.3f m/sec\nOn a road
    banked v=%.3f m/sec\nIf lateral forces are not to
     be experienced v=%.3f m/sec",v,v1,v3)
```

Scilab code Exa 19.5 Car going around a curve

```
//Car going around a curve
//refer fig. 19.9
v=26.667  //m/sec
F=20*(((26.667^2)/(9.81*60))-sind(30))  //kN
//Taking moment about point of contact of outer
    wheel with road surface, we get
R1=20*(((0.8*sind(30))/(1.6))+((cosd(30))/(2))
    +((26.667^2)/(9.81*60))*(((sind(30))/(2))-((0.8*
    cosd(30))/(1.6))))  //kN
//Taking summation of forces normal to road surface
R2=(20*((cosd(30))+(((sind(30))*26.667^2)/(9.81*60))
    ))-9.238  //kN
printf("\nR1=%.3f kN\nR2=%.3f kN",R1,R2)
```

Scilab code Exa 19.6 Super elevation

```
//Super elevation
G=1.68  //m
r=800  //m
//(1)
v=16.667  //m/sec
alpha=atand((v^2)/(9.81*800))  //degree
//Super elevation
e=1000*G*tand(alpha)  //mm
//(2)
v2=22.222  //m/sec
F2=1000*(((0.99937*22.222^2)/(9.81*800))-0.03537)
   //kN
printf("\ne=%.3f mm\nF=%.3f kN",e,F2)
```

Scilab code Exa 19.7 Aeroplane

```
//Aeroplane
//refer fig. 19.10
r=1300  //m
W=8  //kN
v=(400*1000)/(60*60)  //m/sec
//Angle of bank
alpha=atand((111.111^2)/(9.81*1300))  //degree
//Lift under flight condition is
N=80*((cosd(alpha))+(((sind(alpha))*111.111^2)
   /(9.81*1300)))  //kN
```

```
printf("\nalpha=%.2f degree\nN=%.2f kN",alpha,N)
```

Chapter 20

Rotation of rigid bodies

Scilab code Exa 20.1 `Fly wheel`

```
//Fly wheel
//omega=3*t^3-2*t+2
//theta=t^3-t^2+2*t+C
//When t=1  theta=4
C=2
//theta=t^3-t^2+2*t+2
//When t=3
theta=3*3*3-3*3+2*3+2  //radian
omega=3*3*3-2*3+2  //rad/sec
//angular acceleration alpha
//alpha=6*t-2
//when t=3
alpha=6*3-2  //rad/sec^2
printf("\ntheta=%.3f radian\nomega=%.3f rad/sec\nalpha=%.3f rad/sec^2",theta,omega,alpha)
```

Scilab code Exa 20.2 `Flywheel`

```
//Flywheel
//alpha=12-t
//omega=12*t-(t^2)/2+C
//When t=4 sec  omega=60 rad/sec
C1=20
//When t=6 sec
omega=12*6-((6*6)/(2))+20  //rad/sec
//theta=6*t^2-(t^3)/6+20*t+C2
//When t=0  theta0=C2
//When t=6 sec theta6=180+C2
//Angular displacement during 6 seconds=180 rad
//Number of revolution
N=180/(2*%pi)
printf("\nomega=%.3f rad/sec\nNumber of revolution=%.3f ",omega,N)
```

Scilab code Exa 20.3 `Wheel rotating about fixed axis`

```
//Wheel rotating about fixed axis
//Initial velocity
omega0=2.0944  //rad/sec
t=70  //sec
//Angular displacement
theta=100*%pi  //radian
//Using kinematic equation
alpha=0.06839  //rad/sec^2
//Angular velocity at the end of 70 seconds interval
omega=2.0944+0.06839*70  //rad/sec
//Let the time required for the velocity to reach 100 rpm be t
t=(((200*%pi)/(60))-(2.0944))*((1)/(0.06839))  //sec
printf("\nomega=%.3f rad/sec\nt=%.3f sec",omega,t)
```

Scilab code Exa 20.4 Flywheel

```
//Fly-wheel
//theta=200*%pi  //radian
omega0=(120*2*%pi)/(60)  //rad/sec
omega=(160*2*%pi)/(60)  //rad/sec
//Using kinematic relation
alpha=0.0977  //rad/sec^2
//Also
t=(16.755-4*%pi)/0.0977  //sec
//theta' be the total angular displacement in reaching the velocity of 160 rpm
theta=(1436.1)/(2*%pi)  //revolution
printf("\nt=%.3f sec\ntheta=%.3f revolution",t,theta)
```

Scilab code Exa 20.5 Power driven wheel

```
//Power driven wheel
omega0=30*%pi
omega=0
theta=720*%pi //rad
//thus using kinematic equations
alpha=-1.9635  //rad/sec^2
//Also
t=(30*%pi)/(1.9635)  //sec
printf("\nRetardation is %.3f rad/sec^2\nt=%.3f sec",-alpha,t)
```

Scilab code Exa 20.6 The step pulley

```
//The step pulley
//refer fig. 20.3
theta=20  //radian
alpha=2  //rad/sec^2
omega0=0
//Using kinematic relation
t=sqrt(20)  //sec
//Velocity of A
vA=8.944  //m/sec
vB=0.6*8.944  //m/sec
printf("\nt=%.3f sec\nvA=%.3f m/sec\nvB=%.3f m/sec",
   t,vA,vB)
```

Scilab code Exa 20.7 A flywheel

```
//A flywheel
omega0=41.888  //rad/sec
omega=29.322  //rad/sec
t=120  //sec
//Kinematic equation gives
alpha=(29.3224-41.888)/(120)  //rad/sec^2
//Weight of flywheel
Wf=50000  //N
//Radius of gyration
k=1  //m
```

```
I=(50000/9.81)  //kg-m^2
//(1) Retarding torque acting on the flywheel Tr
Tr=5096.84*0.1047  //N-m
//(2) Change in K.E.
C.K.E=(5096.84*((41.888^2)-(27.322^2)))/(2)  //N-m The answer provided in the textbook is wrong
//(3) Change in its angular momentum
C.A.M=5096.84*(41.888-29.322)  //N-sec
printf("\nRetarding torque acting on the flywheel Tr=%.3f N-m\nChange in K.E.=%.3f N-m\nChange in its angular momentum=%.3f N-sec",Tr,C.K.E,C.A.M)
```

Scilab code Exa 20.8 Pulley

```
//Pulley
//refer fig. 20.6
//Let a be the resulting acceleration and T be the tension in the rope
//Angular acceleration of pulley
//alpha=1.667*a rad/sec^2
//Dynamic equilibrium condition for the block gives
//T=(600-(600*a)/(9.81))
//From kinetic equation for pulley
T=(200*7.358)/(9.81)  //N
a=(600*9.81)/(800)  //m/sec^2
printf("\nT=%.3f N\na=%.3f m/sec^2",T,a)
```

Scilab code Exa 20.9 Composite pulley

```
//Composite pulley
//refer fig. 20.7 (a) and (b)
//Let aA be acceleration of 4000 N block and aB that
    of 2000 N block,and alpha be the angular
   velocity of pulley, then
//aA=0.5*alpha
//aB=0.75*alpha
//Writing dynamic equilibrium equation for the two
   blocks and from kinetic equation of pulley
alpha=500/245.97  //rad/sec^2
TA=4000*(1-(0.5*2.033)/(9.81))  //N
TB=2000*(1+(0.75*2.033)/(9.81))  //N
printf("\nalpha=%.3f rad/sec^2\nTA=%.3f N\nTB=%.3f N
   ",alpha,TA,TB)
```

Scilab code Exa 20.10 `welded cylinder`

```
//welded cylinder
//refer fig. 20.8 (a) and (b)
//Mass moment of inertia of the bar about A
IBA=((200)/(2*9.81))+((200*0.5^2)/(9.81))
//Moment of inertia of the cylinder about A
ICA=((500*0.2*0.2)/(2*9.81))+((500*1.2*1.2)/(9.81))
//mass moment of inertia of the system about A
I=6.7958+74.41
//Rotational moment about A
Mt=200*0.5+500*1.2  //N-m
//Equating it to I*alpha
alpha=((700)/(81.2097))  //rad/sec
//Instantaneous acceleration of rod AB is vertical
   with magnitude
Iaccnrod=0.5*8.6197  //m/sec
//Instantaneous acceleration of cylinder is vertical
```

```
    with magnitude
Iaccncylinder=1.2*8.6197  //m/sec
//Applying D'Alembert's dynamic equilibrium equation to the system of forces
RA=200+500-((200*4.3100)/(9.81))-((500*10.344)/(9.81))  //N
printf("\nalpha=%.3f rad/sec\nRA=%.3f N",alpha,RA)
```

Scilab code Exa 20.11 Rods welded

```
//Rods welded
//refer fig. 20.9 (a) and (b)
//Mass moment of inertia of AB about axis of rotation
AB=((200*1.2*1.2)/(12*9.81))+((200*0.6*0.6)/(9.81))
//Mass moment of inertia of rod CD about A
CD=((100*0.6*0.6)/(12*9.81))+((100*1.2*1.2)/(9.81))
//Total mass moment of the system about A
I=9.786+147.0
//Let alpha be the instantaneous angular acceleration
//Kinetic equation for motion gives
alpha=(300*0.75)/(156.786)  //rad/sec
//Writing the dynamic equilibrium condition
VA=200+100  //N
HA=300-((200*0.6*alpha)/(9.81))-((100*1.2*alpha)/(9.81))  //N
printf("\nVA=%.3f N\nHA=%.3f N",VA,HA)
```

Chapter 21

Mechanical vibration

Scilab code Exa 21.1 SHM

```
//S.H.M
//a=-25*s
omega=5
//Period
T=(2*%pi)/(5)  //sec
f=(1/T)  //osc. per second
printf("\nT=%.3f sec\nf=%.3f osc. per second",T,f)
```

Scilab code Exa 21.2 SHM

```
//S.H.M
r=0.75  //m
T=1.2  //sec
omega=((2*%pi)/(1.2))  //rad/sec
vxmax=0.75*5.236  //m/sec
axmax=0.75*5.236^2  //m/sec^2
```

```
printf("\nvx(max)=%.3f m/sec\nax(max)=%.3f m/sec^2",
   vxmax,axmax)
```

Scilab code Exa 21.3 Displacement

```
//Displacement
//After 0.5 sec
theta=5.236*0.5*(180/%pi)
//displacement
x=0.75*sind(150)   //m
//Velocity
vx=0.75*5.236*cosd(150)
ax=0.375*5.236^2   //m/sec^2
printf("\nx=%.3f m\nvx=%.3f m/sec\nax=%.2f ",x,vx,ax
   )
```

Scilab code Exa 21.4 SHM

```
//SHM
//r*sind(theta1)=0.2
//r*omega*cosd(theta1)=0.5
//r*sind(theta2)=0.3
//r*omega*cosd(theta2)=0.35
//thus
theta1=asind(0.44)
r=(0.2)/(sind(26.1))   //m
omega=1.225   //rad/sec^2
vmax=0.454*1.225   //m/sec
amax=-0.454*1.225^2   //m/sec^2
```

```
f=(1.225)/(2*%pi)  //osc. per sec
printf("\namax=%.3f m/sec^2\nf=%.3f osc. per. sec\nvmax=%.3f m/sec",amax,f,vmax)
```

Scilab code Exa 21.6 Period of simple pendulum

```
//Period of simple pendulum
T=2*%pi*sqrt(1.5/9.81)  //sec
printf("\nT=%.2f sec",T)
```

Scilab code Exa 21.7 Length of pendulum

```
//Length of pendulum
L=(1*9.81)/(4*%pi*%pi)  //m
printf("\nL=%.3f m",L)
```

Scilab code Exa 21.9 Compound pendulum

```
//Compound pendulum
//refer fig. 21.8
//Length of uniform rod
l=0.6  //m
//Radius of uniform disc
r=0.3  //m
```

```
//Mass moment of inertia about centre of suspention
Iz=((25*0.6^2)/(9.81*12))+((25*0.6^2)/(9.81*2*2))+((40*0.15^2)/(2*9.81))+((40*(0.6+0.15)^2)/(9.81))  //units
M=((25)/(9.81))+((40)/(9.81))
//Kzz^2=0.3992
//Distance of centre of gravity of compound pendulum from centre of suspension
r=(25*0.3+40*0.75)/(25+40)  //m
T=2*%pi*sqrt((0.3992)/(0.5769))  //sec
//equivalent length
Le=(0.3992/0.5769)  //m
printf("\T=%.3f sec\nLe=%.3f m",T,Le)
```

Scilab code Exa 21.10 `Circular ring`

```
//Circular ring
//refer fig. 21.9 (a) and (b)
//Radius of gyration about centre of the ring
//Kz^2=(((R1^2)+(R2^2))/(2))+R2^2
//thus
Kz=sqrt(((1+0.75*0.75)/(2))+(0.75^2))
T=2*%pi*sqrt(1.34375/(9.81*0.75))  //sec
printf("\nT=%.3f sec",T)
```

Chapter 22

General plane motion of rigid bodies

Scilab code Exa 22.2 1 m radius wheel

```
//1 m radius wheel
//refer fig. 22.4(a),(b),(c),(d),(e) and (f)
vA=1*5  //m/sec
aA=1*4  //m/sec^2
vBA=1*5  //m/sec
vB=vA+vBA  //m/sec
aBA=1*4  //m/sec^2
an=5^2  //m/sec^2
aB=sqrt((8^2)+(25^2))  //m/sec^2
theta=atand(25/8)  //degree
//Consider rotation of point D
vDx=5+3*sind(60)  //m/sec
vDy=3*cosd(60)  //m/sec
vD=7.745  //m/sec
//inclination to horizontal
theta2=atand(1.5/7.598)  //degree
vDA=0.6*5  //m/sec^2
aD=sqrt((14.190^2)+(1.422^2))  //m/sec^2
theta3=atand(14.190/1.422)  //degree
```

```
printf("\nAt B\naB=%.3f m/sec^2\ntheta=%.2f degree\nvB=%.3f m/sec\nAt D\nvD=%.3f m/sec^2\ntheta2=%.2f degree\naD=%.3f m/sec^2\ntheta3=%.2f degree",aB,theta,vB,vD,theta2,aD,theta3)
```

Scilab code Exa 22.3 slender beam

```
//slender beam
//refer fig. 22.5 (a),(b) and (c)
//from vector diagram
vB=2*cotd(60)  //m/sec
vBA=(2/sind(60))  //m/sec
printf("\nvB=%.3f m/sec^2\nvB/A=%.3f m/sec",vB,vBA)
//acceleration of point B
aB=(1.778*sind(60))+(3*0.958*sind(30))
printf("\naB=%.3f m/sec^2",aB)
```

Scilab code Exa 22.4 Length of crank

```
//Length of crank
//refer fig. 22.6 (a)
//angular velocity
omega=(1500*2*%pi)/(60)  //rad/sec
r=0.100
//Tangential velocity of end B
vB=r*omega  //m/sec
//Consider motion of connecting rod BC
theta=asind((100*sind(30))/(250))  //degree
//Refer fig. 22.6
```

```
//Let omega' be the angular velocity of BC
omega1=13.6035/0.244  //rad/sec
//Considering horizontal component of velocities
vC=15.7080*cosd(60)+0.25*55.547*sind(11.5378)  //m/
    sec
printf("\nomega1=%.3f rad/sec\nvC=%.2f m/sec",omega1
    ,vC)
```

Scilab code Exa 22.5 Velocities of point B and D

```
//Velocities of point B and D
//refer fig. 22.8
vA=5*1  //m/sec
//Instantaneous centre in vertically downward
    direction
Ic=5/5  //m
vB=2*5  //m/sec
CP=1+0.6*sind(60)  //m
PD=0.6*cosd(60)  //m
CD=sqrt((1.520^2)+(0.3^2))  //m
vD=1.549*5  //m/sec
//Inclination to horizontal
theta=atand((0.3)/(1.520))  //degree
printf("\nvD=%.2f m/sec\ntheta=%.2f degree",vD,theta
    )
```

Scilab code Exa 22.6 Velocity of B

```
//Velocity of B
```

```
//refer fig.2.9
omega=(2)/(3*sind(60))  //rad/sec
vB=3*0.770*cosd(60)  //m/sec
printf("\nvB=%.3f m/sec",vB)
```

Scilab code Exa 22.7 Solid cylinder acted upon by force P

```
//Solid cylinder acted upon by force P
//refer fig. 22.11 (a) and (b)
//(1)Maximum Value of P for Rolling without slipping
//aA=0.8*alpha
I=(1200*0.8^2)/(2*9.81)
W=1200  //N
N=W  //N
//From law of friction
F=0.2*1200  //N
//Consider moment equilibrium equation about C
//on solving
alpha=(240)/(97.859-73.394)  //rad/sec
P=73.394*9.81  //N
//(2) When P=1000 N
F2=0.15*1200  //N
//Taking moment about A
alpha2=(1000*0.8-180*0.8)/(39.144)  //rad/sec^2
aA=((1000+180)*9.81)/(1200)  //rad/sec^2
printf("\nMaximum Value of P for Rolling without slipping \nP=%.2f N\nWhen P=1000 N\nalpha=%.2f rad/sec^2\naA=%.3f rad/sec^2",P,alpha2,aA)
```

Scilab code Exa 22.9 Uniform bar

```
//Uniform bar
//refer fig. 22.15(a),(b),(c),(d),(e) and (f)
beta=atand(0.6928/1.7856)  //degree
//aG=alpha*sqrt((1.7856^2)+(0.6928^2))
I=(300*1.6^2)/(12*9.81)
//Equating
alpha=((300*0.6928)/(82.3160))  //rad/sec^2
//Taking horizontal components of the forces
NB=(300*1.7856*2.5249)/(9.81*cosd(45))  //N
//Taking vertical components of the forces
NA=(300)-(194.98*sind(45))+((300*0.6928*2.5249)/(9.81))  //N (Printing mistake in text book)
printf("\nNA=%.2f N\nNB=%.2f N\nalpha=%.2f rad/sec^2",NA,NB,alpha)
```

Scilab code Exa 22.10 rotating crank

```
//rotating crank
//refer fig. 22.16 (a),(b),(c),(d),(e),(f),(g),(h) and (i)
//from sine rule
theta=asind((80*sind(60))/(200))  //degree
//angular velocity of crank
omega=(2*%pi*1800)/(60)  //rad/sec
vB=0.08*omega  //m/sec
//it is at right angles to BC
aB=(15.0796^2/0.8)  //m/sec^2
vA=15.8436  //m/sec
aB=2842.4292  //making 60 degree with horizontal
alpha=13120.457  //rad/sec^2
aAB=0.2*alpha
```

```
aA=512.2027   //m/sec^2
aX=1118.2109   //m/sec^2
aY=-1174.862   //m/sec^2  downward
//Consider dynamic equilibrium of piston A
HA=(4000)-((50*512.2027)/(9.81))   //kN The answer provided in the textbook is wrong
//Taking moment about B
vA=813.95   //N
vB=2001.57   //N
HB=2598.51   //N
printf("\nHA=%.2f kN\nHB=%.2f N\nvA=%.2f N\nvB=%.2f N",HA,HB,vA,vB)
```