Scilab Textbook Companion for Engineering Mechancis-Schaum Series by McLean[1]

Created by
Akshay
Engineering
Civil Engineering
university Of Pune
College Teacher
None
Cross-Checked by
Spandana

June 2, 2016

[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 Mechancis-Schaum Series

Author: McLean

Publisher: McGraw-Hill, New Delhi

Edition: 5

Year: 1997

ISBN: 0-07-046193-7

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 1

Vectors

Scilab code Exa 1.1 Vectors

```
//Initialisation of Variables
f1=120 //lb
f2=100 //lb
theta=((60*%pi)/180) //radians
//Calculations
R=sqrt(120^2+100^2-(2*120*100*cos(theta))) //
    Applying Thr rule of Cosines
alpha1=(((asin(120*sin(theta)/111))*180)/%pi) //
    Applying the Law of Sines
alpha=alpha1+270 //As the vector lies in the fourth
    Quadrant by obsrevaton
//Results
clc
printf('The Resultant of The force system is equal
    to:%f N\n ',R ) //lb
printf('The Resultant is at:%f degrees',alpha) //
    degrees
```

Scilab code Exa 1.2 Vectors

```
//Initilization of variables
P=100 //lb
Q=120 //lb
theta=((30*%pi)/180) //radians
//Calculations
R_x=Q*cos(theta) //lb
R_y=Q*sin(theta)-P //lb
R=sqrt(R_x^2+R_y^2) //lb Triangle law
Theta_1=((atan(R_y/R_x))*180)/%pi //degrees
Theta_R=360+Theta_1 //degrees
//Result
clc
printf('The resultant of the force system is:%f N\n'
    ,R) //lb
printf('The resultant is at:%f degrees',Theta_R) //
    Degrees
```

Scilab code Exa 1.3 Vectors

```
//Initialization of variables
R=400 // N
F2=200 //N
Theta1=((120*%pi)/180) //radians
Theta2=((20*%pi)/180) //radians
Theta=Theta1-Theta2 //radians
//Calculation
F=sqrt(R^2+F2^2-(2*R*F2*cos(Theta))) //N Applying
    the Rule of Cosine
Theta_r=asin((400*sin(Theta))/F) //radians Applying
    the rule of sines
Theta_R=(Theta_r*180)/%pi
//Result
clc
printf('The resultant of the force system is:%f N \n
    ',F) //N
```

```
printf('The Angle between F and 200N force is :%f degrees \n',Theta_R) //degrees
```

Scilab code Exa 1.4 Vectors

```
//Initilization of variables
F1=280 //N
F2=130 //N
Theta1=((320*%pi)/180) //Radians
Theta2=((60*%pi)/180) //Radians
//Calculations
R_x=-F1*cos(Theta1)+F2*cos(Theta2) //N
R_y=F1*sin(Theta1)-F2*sin(Theta2) //N
R=sqrt(R_x^2+R_y^2) //N Applying Triangle Law
ThetaR=atan(R_y/R_x) //radians
Theta_R=360-(ThetaR*180/%pi) //degrees
//Result
clc
printf('The resultant of the force system is:%f N\n',R) //N
printf('The resultant is at:%f degrees',Theta_R) //degrees
//The decimal point accuracy causes discrepancy in answers
```

Scilab code Exa 1.5 Vectors

```
//Initialization of variables
F1=26 //lb
F2=39 //lb
F3=63 //lb
F4=57 //lb
T1=((10*%pi)/180)//Radians
```

```
T2=((114*%pi)/180)//Radians
T3=((183*%pi)/180) //radians
T4=((261*%pi)/180) //radians
//Calculations
R_x=F1*cos(T1)+F2*cos(T2)+F3*cos(T3)+F4*cos(T4) //lb
    Resolving vectors
R_y=F1*sin(T1)+F2*sin(T2)+F3*sin(T3)+F4*sin(T4) //lb
    resolving vectors
R=sqrt(R_x^2+R_y^2) //lb Applying Triangle Law
theta=atan(R_y/R_x)//radians
Theta=theta*180/%pi //degrees
Theta_R=180+Theta
//Results
clc
printf('The Resultant of the force system is:%f lb\n
   ',R) //lb
printf('The resultant is at:%f degrees',Theta_R) //
   degrees
```

Scilab code Exa 1.6 Vectors

```
//Initilization of variables
F=10 //lb
theta1=((60*%pi)/180) //radians
theta2=((45*%pi)/180) //radians
theta=theta1-theta2 //radians
//Calculation
F_OH=F/cos(theta) //lb resolving vectors
//Result
clc
printf('The component of F in the direction of OH is
   :%f lb',F_OH) //lb
```

Scilab code Exa 1.7 Vectors

```
//Initilization of variables
weight=80 //kg
theta=((20*%pi)/180) //radians
theta_p=((70*%pi)/180)
//Calcuations
//Part (a)
F=weight*9.81 //N
R=F*cos(theta) //N
//part (b)
R_p=F*cos(theta_p) //N
//Result
clc
printf('The normal component is:%f N \n',R) //N
printf('The parallel component is:%f N',R_p) //N
```

Scilab code Exa 1.8 Vectors

```
//Initilization of variables
P=235 //N
theta=((60*%pi)/180) //radians
bet=((22*%pi)/180) //radians
gam=((38*%pi)/180) //radians
//Calculations
//Part (a)
P_h=P*cos(theta) //N
P_v=P*sin(theta) //N
//Part (b)
P_l=P*cos(theta-bet) //N
P_p=P*sin(gam) //N
//Result
clc
printf('The horizontal component is:%f N\n',P_h) //N
printf('The vertical component is:%f N\n',P_v) //N
```

```
printf('The component parallel to plane is:%f N\n',P_l) //N
printf('The component perpendicular to the plane is:%f N',P_p) //N
//The decimal point accurasy might cause a small discrepancy in the answers
```

Scilab code Exa 1.9 Vectors

```
//Initilization of variables
F1=90 //lb
theta1=((40*%pi)/180) //radians
theta2=((30*%pi)/180) //radians
//Calculations
R_x=0 //lb
R_y=20 //lb
//Taking the sum of forces in the X-Direction
P=((F1*cos(theta1))/cos(theta2)) //lb
//Taking the sum of the forces in the Y-Direction
F=(P*sin(theta2))+(F1*sin(theta1))-20 //lb
//Results
clc
printf('The value of P is:%f lb\n',P) //lb
printf('The value of F is:%f lb',F) //lb
```

Scilab code Exa 1.10 Vectors

```
//Initilization of variables
x=4 //m
y=3 //m
z=2 //m
F=50 //N
//Calculations
```

```
OP=sqrt(x^2+y^2+z^2) //m
thetax=(x/OP) //radians
thetay=(y/OP) //Radians
thetaz=(z/OP) //radians
P_x=F*(thetax) //N
P_y=F*(thetay) //N
P_z=F*(thetaz) //N
//Result
clc
printf('The vector P is:%fi+%fj+%fk N',P_x,P_y,P_z)
    //N
```

Scilab code Exa 1.11 Vectors

```
//Initilization of variables
x=2
y=-4
z=1
F=100 //N
//Calculation
thetax=x/sqrt(x^2+y^2+z^2) //radians
thetay=y/sqrt(x^2+y^2+z^2) //radians
thetaz=z/sqrt(x^2+y^2+z^2) //radians
P_x=F*thetax //N
P_y=F*thetay //N
P_z=F*thetaz //N
//Result
clc
printf('The vector P is:%fi%fj+%fk N',P_x,P_y,P_z)
    //N
```

Scilab code Exa 1.13 Vectors

```
//Initilization of variables
Fx=2.63 //N
Fy=4.28 //N
Fz=-5.92 //N
//Calculation
F=sqrt(Fx^2+Fy^2+Fz^2) //N
thetax=((acos(Fx/F))*180)/%pi //degrees
thetay=((acos(Fy/F))*180)/%pi //degrees
thetaz=((acos(Fz/F))*180)/%pi //degrees
//Result
clc
printf('The magnitude of force is:%f N \n',F) //N
printf('Thetax:%f degrees\n',thetax) //degrees
printf('Thetay:%f degrees\n',thetay) //degrees
printf('Thetaz:%f degrees',thetaz) //degrees
```

Scilab code Exa 1.14 Vectors

```
//Initilization of variables
P=[4.82, -2.33, 5.47] //N
Q=[-2.81,-6.09,1.12 ] //m
//Calculations
M=P*Q' //Nm
//Results
clc
printf('Result is:%f N.m',M) //N-m
```

Scilab code Exa 1.15 Vectors

```
//Initilization of variables
x1=2 //units
x2=-2 //units
y1=3 //units
```

```
y2=4 //units
z1=0 //units
z2=6 //units
P=[2,3,-1] //units
//Calculations
X=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2) //units
eLx=(x2-x1)/X //units
eLy=(y2-y1)/X //units
eLz=(z2-z1)/X //units
Q=[eLx,eLy,eLz] //units
Z=P*Q' //units
//Result
clc
printf('The unit vector is:%fi+%fj+%fk \n',eLx,eLy,eLz)
printf('The projection of P is:%f',Z)
//Note:The final answer for the projection of P is off by 0.1 units
//The answer mentioned in the textbook is −1.41
```

Scilab code Exa 1.16 Vectors

```
//Initilization of variables
x1=2 //units
x2=5 //units
y1=-5 //units
y2=2 //units
z1=3 //units
z2=-4 //units
P=[10,-8,14] //units
//Calculations
X=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2) //units
eLx=(x2-x1)/X //units
eLy=(y2-y1)/X //units
eLz=(z2-z1)/X //units
```

```
Q=[eLx,eLy,eLz] //units
Z=P*Q' //units
//Result
clc
printf('The unit vector is:%fi+%fj%fk \n',eLx,eLy,eLz)
printf('The projection of P is:%f lb',Z) //lb
```

Scilab code Exa 1.17 Vectors

```
Px=2.85 //ft
Py=4.67 //ft
Pz=-8.09 //ft
Qx=28.3 //lb
Qy=44.6 //lb
Qz=53.3 //lb
//Calculations
X=(Py*Qz-Pz*Qy) //N.m
Y=(Pz*Qx-Px*Qz) //N.m
Z=(Px*Qy-Py*Qx) //N.m
//Result
clc
printf('The cross product is:%fi%fj%fk lb-ft',X,Y,Z) //lb-ft
```

Scilab code Exa 1.19 Vectors

```
//Initilization Of Variables
a=1 //Lower Limit oF the Integral
b=3 //Upper Limit of the Integral
n=10 //Interval of the integral
//Calculation
//Using Trapezoidal Rule for Intergration
```

```
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=t^2')
function[I2]=Trap_Composite2(f1,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I2=(h/2)*((2*sum(f1(t)))-f1(t(1))-f1(t(n+1)))
endfunction
deff('[y1]=f1(t)','y1=t*2')
z=b-a
//Result
clc
printf('%fi+%fj%fk',Trap_Composite1(f,a,b,n),
   Trap_Composite2(f1,a,b,n),-z)
```

Chapter 2

Operations With Forces

Scilab code Exa 2.1 OPF

```
//Initilization of variables
F=20 //lb
L=4.33 //ft
//Calculation
M=-F*L //lb-ft
//Result
clc
printf('The moment of force F about O is:%f lb-ft',M
  )
```

Scilab code Exa 2.2 OPF

```
//Initilization of variables
F=20 //lb
theta=((60*%pi)/180) //radians
L=5 //ft
//Calculations
F_x=F*cos(theta) //Resloving the vector
```

```
F_y=F*sin(theta) //Resloving the vector
M=-F_y*L //Appling Varignon's theorem
//Negative sign tells that moment is clockwise
//Result
clc
printf('The moment of the force about O is:%f lb-ft',M)
```

Scilab code Exa 2.3 OPF

```
//Initilization of variables
F=100 //N
x1=2 //m
x2=5 //m
y1=0 //m
y2=1 //m
z1=4 //m
z2=1 //m
//Calculations
xside=(x2-x1) //m
yside=(y2-y1) //m
zside=(z2-z1) //m
LD=sqrt(xside^2+yside^2+zside^2)
Fx=(xside/LD)*F //N
Fy=(yside/LD)*F //N
Fz=(zside/LD)*F //N
Mx=-Fy*z1 //N-m
My=Fx*x1-Fz*z1 //N-m
Mz=Fy*x1 //N-m
//Result
clc
printf('Fx is:%f N\n',Fx) //N
printf('Fy if:%f N\n',Fy) //N
printf('Fz is:%f N\n',Fz) //N
printf('Moment about X-Axis is:%f N.m\n',Mx) //N-m
```

```
printf('Moment about Y-Axis is:%f N.m\n',My) //N-m
printf('Moment about Z-Axis is:%f N.m',Mz) //N-m
```

Scilab code Exa 2.4 OPF

```
//Initilization of variables
Fx=68.7 //N
Fy=22.9 //N
Fz=-68.7 //N
rx=2 //m
ry=0 //m
rz=4 //m
rx1=5 //m
ry1=1 //m
rz1=1 //m
//Calculation
Mx=Fz*ry-Fy*rz //N-m
My=-(Fz*rx-Fx*rz) //N-m
Mz=Fy*rx-Fx*ry //N-m
Mx1=Fz*ry1-Fy*rz1 //N-m
My1=-(Fz*rx1-Fx*rz1) //N-m
Mz1=Fy*rx1-Fx*ry1 //N-m
//Result
clc
printf('Moment with respect to origin using point
    (2,0,4):%fi+%fj+%fk N.m\n',Mx,My,Mz) //N-m
printf('Moment with respect to origin using point
    (5,1,1):%fi+%fj+%fk N.m \n',Mx1,My1,Mz1) //N-m
```

Scilab code Exa 2.5 OPF

```
//Initilization of variables
Fx=2 //lb
```

```
Fy=3 //lb
Fz=-1 //lb
rx=1 //ft
ry=-4 //ft
rz=3 //ft
//Coordinates of points
ax=3 //ft
ay=1 //ft
az=1 //ft
bx=3 //ft
by=-1 //ft
bz=1 //ft
cx=2 //ft
cy=5 //ft
cz=-2 //ft
//Calculations
Rx=ax-cx //ft
Ry=ay-cy //ft
Rz=az-cz //ft
Mx=(Ry*Fz)-(Rz*Fy) //lb-ft
My=-((Rx*Fz)-(Rz*Fx)) //lb-ft
Mz=(Rx*Fy)-(Ry*Fx) //lb-ft
E_u=sqrt((bx-cx)^2+(by-cy)^2+(bz-cz)^2) //ft
ex=(bx-cx)/E_u //ft
ey=(by-cy)/E_u //ft
ez=(bz-cz)/E_u //ft
M_lx=Mx*ex //lb-ft
M_ly=My*ey //lb-ft
M_lz=Mz*ez //lb-ft
M_l=M_lx+M_ly+M_lz //lb-ft
//Result]
clc
printf('Hence the moment about line is %f lb-ft \n',
   M_l)
```

Scilab code Exa 2.6 OPF

```
//Initilization of variables
P_x=22 //N
P_y=23 //N
P_z=7 //N
p1=1 //m
p2=-1 //m
p3=-2 //m
//Calculations
Mx=(p2*P_z)-(p3*P_y) //N-m
My=-((p1*P_z)-(p3*P_x)) //N-m
Mz=(p1*P_y)-(p2*P_x) //N-m
//Result
clc
printf('The moment about the line from the origin is
   :%fi%fj+%fk N.m',Mx,My,Mz) //N-m
```

Scilab code Exa 2.8 OPF

```
//Initilization of variables
F=10 //N Force couple
a=3 //m Moment arm
//Calculations
C=-F*a //N-m
//Result
clc
printf('The resultant couple is: %fN.m \n',C)
```

Scilab code Exa 2.11 OPF

```
//Initilization of variables
C1=20 //N-m
```

```
C2=40 //N-m
C3=-55 //N-m
//Calculations
C=sqrt(C1^2+C2^2+C3^2) //N-m
thetax=C2/C
thetay=C3/C
thetaz=C1/C
Cx=C*thetax //N-m
Cy=C*thetay //N-m
Cz=C*thetaz //N-m
//Result
clc
printf('Couple in vector notation:%fi%fj+%fk N.m',Cx
    ,Cy,Cz) //N-m
```

Scilab code Exa 2.12 OPF

```
//Initilization of variables
F1=25 //lb
F2=25 //lb
L1=14 //in
L2=20 //in
//Calculations
C=F1*L1 //lb-in
M=-F2*L2 //lb-in
//Result
clc
printf('The twisting couple is:%f lb-in\n',C) //lb-
    in
printf('The bending moment is:%f lb-in',M) //lb-in
```

Scilab code Exa 2.13 OPF

```
//Initilization of variables
rx=20 //in
ry=0 //in
rz=14 //in
Fx=0 //lb
Fy=-25 //lb
Fz=0 //lb
//Calculation
Mx=ry*Fz-rz*Fy //lb-in
My=rx*Fz-rz*Fx //lb-in
Mz=rx*Fy-ry*Fx //lb-in
//Result
clc
printf('The moment of the 25-lb force is:%fi+%fj%fk
    lb.in',Mx,My,Mz) //lb-in
```

Scilab code Exa 2.14 OPF

```
//Initilization of variables
//Co-ordinates with respect to point O
x=17.9 //ft
y=6.91 //ft
z=46.3 //ft
Fz=-4000 //lb
Fy=0 //lb
//Calculation
Mx=y*Fz-z*Fy //lb-ft
//Result
clc
printf('The scalar coefficient of the i term is the
    moment about the X-Axis Mx:%f lb-ft lb-ft',Mx) //
    lb-ft
//The answer in the text book is incorrect due to
    decimal point accuracy in scilab
```

Chapter 3

Resolution of Coplanar Force System

Scilab code Exa 3.1 Resolution

```
//Initilization of variables
F1=150 //lb
F2=200 //lb
F3=80 //lb
F4=180 //lb
theta1=((30*%pi)/180) //radians
theta2=((150*%pi)/180) //radians
theta3=((240*%pi)/180)  //radians
theta4=((315*%pi)/180)  //radians
//Calculations
F1x=F1*cos(theta1) //lb
F1y=F1*sin(theta1) //lb
F2x=F2*cos(theta2) //lb
F2y=F2*sin(theta2) //lb
F3x=F3*cos(theta3) //lb
F3y=F3*sin(theta3) //lb
F4x=F4*cos(theta4) //lb
F4y=F4*sin(theta4) //lb
Fx=F1x+F2x+F3x+F4x //lb
```

```
Fy=F1y+F2y+F3y+F4y //lb
R=sqrt(Fx^2+Fy^2) //lb
theta=((atan(Fy/Fx))*180)/%pi //degrees
theta_R=360+theta //degrees
//Result
clc
printf('The resultant of the force system is:%f lb\n
   ',R) //lb
printf('The resultant is at :%f degrees',theta_R) //
   degrees
```

Scilab code Exa 3.2 Resolution

```
//Initilization of variables
F1=50 //N
F2=100 //N
F3=30 //N
//Calculation
//The book has a misprint for squareroot of 1^2
F1x=F1/sqrt(2) //N
F1y=F1/sqrt(2) //N
F2x=-(F2*3)/sqrt(10) //N
F2y=(-F2)/sqrt(10) //N
F3x=F3/sqrt(5) //N
F3y=(-F3*2)/sqrt(5) //N
Fx=F1x+F2x+F3x //N
Fy=F1y+F2y+F3y //N
R=sqrt(Fx^2+Fy^2) //N
theta=atan(Fy/Fx) //radians
theta_x=180+(theta*180)/%pi //degrees
//Result
clc
printf('The resultant is:%f N\n',R) //N
printf('The resultant makes an angle of:%f degrees',
   theta_x) //degrees
```

Scilab code Exa 3.3 Resolution

```
//Initilization of variables
F1=70 //lb
F2=100 //lb
F3=125 //lb
theta1=0 //radians
theta2=((10*%pi)/180) //radians
theta3=((30*%pi)/180) //radians
//Calculations
Fx=F1-(F2*cos(theta3))-(125*sin(theta2)) //lb
Fy=125*cos(theta2)-(100*sin(theta3)) //lb
R=sqrt(Fx^2+Fy^2) //lb
theta=atan(Fy/Fx) //radians
theta_x=180+(theta*180)/%pi //degrees
//Result
clc
printf('The resultant of the force system is:%f lb\n',R) //lb
printf('The resultant is at %f with respect to the X-Axis degrees',theta_x) //degrees
```

Scilab code Exa 3.4 Resolution

```
//Initilization of variables
F1=-20 //N
F2=30 //N
F3=5 //N
F4=-40 //N
//Distances with respect to point O
x1=6 //m
```

```
x2=0 //m
x3=8 //m
x4=13 //m
//Calculations
R=F1+F2+F3+F4 //N
//Applying moment about point O equal to zero
M_O=-(F1*x1)+(F2*x2)+(F3*x3)+(F4*x4) //N-m
//Applying moment about point O equal to R*x
x=M_O/R //m
//Result
clc
printf('The resultant of force system is:%f N\n',R) //N
printf('The moment about point O is:%f N-m\n',M_O) //N-m
printf('The resultant of moment acts at %f meters from point O',x) //m
```

Scilab code Exa 3.5 Resolution

```
//Initilization of variables
F1=-100 //lb
F2=200 //lb
F3=-200 //lb
F4=400 //lb
F5=-300 //lb
//Distance with respect to point O
x1=0 //ft
x2=2 //ft
x3=5 //ft
x4=9 //ft
x5=11 //ft
//Calculation
R=F1+F2+F3+F4+F5 //lb
M_O=(F1*x1)+(F2*x2)+(F3*x3)+(F4*x4)+(F5*x5) //N-m
```

```
//Result
clc
printf('The resultant of the force system is:%i lb \n',R) //lb
printf('The moment about point O is:%i lb-ft',M_O) //lb-ft
```

Scilab code Exa 3.6 Resolution

```
//Initilization of variables
F1=20 //lb
F2=20 //lb
F3=-40 //lb
//Distance from point O
x1=3 //ft
x2=3 //ft
//Calculations
R=F1+F2+F3 //lb
M_O=-(F1*x1)+(F2*x2) //lb-ft
//Results
clc
printf('The resultant of the force system is:%i lb\n',R) //lb
printf('The Moment about point O is:%i lb-ft',M_O) //lb-ft
```

Scilab code Exa 3.7 Resolution

```
//Initilization of variables
F1=500 //N
F2=-400 //N
F3=-200 //N
C=1500 //N-m
```

```
//Distance from point O
x1=2 //m
x2=4 //m
x3=6 //m
//Calculations
R=F1+F2+F3 //N
M_O=(F1*x1)+(F2*x2)+(F3*x3)+C //N-m
//Applying Varignons theorem
x=M_O/R //m
//Result
clc
printf('The resultant of the force system is:%i N\n',R) //N
printf('The moment about point O is:%i N-m\n',M_O) //N-m
printf('The resultant acts at %i meters from point O m',x) //m
```

Scilab code Exa 3.8 Resolution

```
//Initilization of variables
F1=50 //lb
F2=100 //lb
theta1=((45*%pi)/180) //radians
//Distance from point O
x1=5 //ft
x2=4 //ft
//Calculation
F_x=F1-(F2*cos(theta1)) //lb
F_y=F1-(F2*sin(theta1)) //lb
R=sqrt(F_x^2+F_y^2) //lb
M_O=F1*x1-(x2*F1) //lb-ft
//Applying Varignons Theorem
x=M_O/R //ft
//Result
```

```
clc
printf('The resultant of the force system is:%f lb\n',R) //lb
printf('The Moment about point O is:%f lb-ft\n',M_O) //lb-ft
printf('The Resultant acts at %f feet from point O ft',x) //ft
```

Scilab code Exa 3.9 Resolution

```
//Initilization of variables
A=80 //N
B=120 //N
C=100 //N
D=50 //N
thetaA=((90*%pi)/180) //radians
thetaB=((150*%pi)/180) //radians
thetaC=((45*%pi)/180) //radians
thetaD=((340*%pi)/180) //radians
//Calculations
Ax=A*cos(thetaA) //N
Ay=A*sin(thetaA) //N
Bx=B*cos(thetaB) //N
By=B*sin(thetaB) //N
Cx=C*cos(thetaC) //N
Cy=C*sin(thetaC) //N
Dx=D*cos(thetaD) //N
Dy=D*sin(thetaD) //N
M_Ax=0 //N-m
M_Ay=0 //N-m
M_Bx=-Bx*5 //N-m
M_By=By*8 //N-m
M_Cx=-Cx*1 //N-m
M_Cy=Cy*1 //N-m
M_Dx=-Dx*-1 //N-m
```

```
M_Dy=Dy*8 //N-m
Fx=Ax+Bx+Cx+Dx //N
Fy=Ay+By+Cy+Dy //N
R=sqrt(Fx^2+Fy^2) //N
M_O=M_Dx+M_Dy+M_Cx+M_Cy+M_Bx+M_By+M_Ax+M_Ay //N-m
theta=atan(Fy/Fx) //radians
theta_x=(theta*180)/%pi //degrees
//Appliying Varignons theorem
x=M_O/R //m
//Result
clc
printf('The resultant of the force system is:%f N\n'
    ,R) //N
printf('The moment about point O is:%f N\n',M_O) //N
printf('The resultant acts at and angle of %f
    degrees with respect to X-Axis degrees\n',theta_x
    ) //degrees
printf('The resultant of the force system acts at %f
     meters from point O',x) //m
```

Scilab code Exa 3.10 Resolution

```
//Initilization of variables
F1=100 //lb
F2=80 //lb
F3=120 //lb
F4=150 //lb
theta1=((60*%pi)/180) //radians
theta2=((45*%pi)/180) //radians
theta3=((90*%pi)/180) //radians
theta4=((75*%pi)/180) //radians
//Distance from point O
x1=-5 //ft
y1=20 //ft
x2=10 //ft
```

```
y2=10 //ft
x3=25 //ft
y3=25 //ft
x4=35 //ft
y4=15 //ft
//Calculations
Fx=F1*cos(theta1)+F2*cos(theta2)+F4*cos(theta4) //lb
Fy=-F1*sin(theta1)+F2*sin(theta2)-F3-F4*sin(theta4) //lb
R=sqrt(Fx^2+Fy^2) //lb
theta=atan(Fy/Fx) //radians
theta_x=(theta*180)/%pi //degrees
M_O=-(F1*cos(theta1)*y1)+(-x1)*(F1*sin(theta1))-(x2)*(F2*cos(theta2))+(y2)*(F2*sin(theta2))-(x3*F3)-(y4*F4*cos(theta4))-(x4*F4*sin(theta4)) //lb-ft
//Applying varignons theorem
x=M_O/Fy //ft
y=-M_O/Fx //ft
//Results
clc
printf('The resultant of the force system is:%f lb\n',R) //lb
printf('The resultant acts at %f degrees with respect to X-Axis \n',theta_x) //degrees
printf('The moment about point O is:%f lb-ft \n',M_O) //lb-ft
printf('The x intercept of resultant is:%f ft\n',x) //ft
printf('The y intercept of resultant is:%f ft\n',y) //ft
//Answer for angle should be negative which has not been mentioned in the tectbook but a schematic shows the angle in fourth quadrant to clarify the doubt
```

Scilab code Exa 3.11 Resolution

```
//Initilization of variables
F1=150 //lb
F2=80 //lb
F3=100 //lb
F4=50 //lb
theta1=((45*%pi)/180) //radians
r=3  //units
//Calculations
Fh=F1-F3*cos(theta1) //lb
Fv=F4-F2-F3*sin(theta1) //lb
R=sqrt(Fh^2+Fv^2) //lb
//Applying the Varignons Theorem
a=(F4*r-F1*r+F2*r-F3*r)/R //units
//Result
clc
printf('The resultant of the force system is:%f lb \
   n',R) //lb
printf('The resultant acts at %f units form the
   point O',a) //units
//Negative sign indicates a negative moment caused
   by the resultant
```

Scilab code Exa 3.12 Resolution

```
//Initilization of variables
F1=150 //lb
F2=200 //lb
F3=200 //lb
F4=225 //lb
M=900 //lb-ft
Theta1=(45*%pi)/180 //radians
Theta2=(30*%pi)/180 //radians
x1=3 //ft
```

```
x2=15 //ft
x3=12 //ft
x4=6 //ft
//Calculations
Fx=F1*cos(Theta1)+F2-F4*cos(Theta2) //Applying sum
   of all forces equal to zero in X direction
Fy=F1*sin(Theta1)-F4*sin(Theta2)+F2 //Applying sum
   of all forces equal to zero in Y direction
R=sqrt(Fx^2+Fy^2) //lb
theta=atand(Fy/Fx) //degrees
M_o=x1*F2-x2*F1*cos(Theta1)+x3*F1*sin(Theta1)-x4*F2+
   M+x4*F4*cos(Theta2)-x1*F4*sin(Theta2) //Moment
   about point O
x=M_o/Fy //Varignons Theorem
//Result
clc
printf('The x intercept of resultant position is %f\
   n',x)
printf('The Resultant is %f lb and acts at an angle
   of %f degrees',R,theta)
```

Scilab code Exa 3.13 Resolution

```
//Initilization Of Variables
a=0 //Lower Limit oF the Integral
b=6 //Upper Limit of the Integral
n=10 //Interval of the integral
l=20 //lb/ft
//Calculation
//Using Trapezoidal Rule for Intergration
R=(b-a)*l
function[I2]=Trap_Composite2(f1,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I2=(h/2)*((2*sum(f1(t)))-f1(t(1))-f1(t(n+1)))
```

```
endfunction
deff('[y1]=f1(t)','y1=20*t')
d=Trap_Composite2(f1,a,b,n)/R //ft
//Result
clc
printf('The value of R is:%i lb\n',R)
printf('The value of d is:%i ft',d)
```

Scilab code Exa 3.14 Resolution

```
//Initilization Of Variables
a=0 //Lower Limit oF the Integral
b=9 //Upper Limit of the Integral
n=10 //Interval of the integral
//Calculation
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=(t/9)*30') //y defined as a
   function of t and not x
function[I2]=Trap_Composite2(f1,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I2=(h/2)*((2*sum(f1(t)))-f1(t(1))-f1(t(n+1)))
endfunction
deff('[y1]=f1(t)','y1=(t^2/9)*30')
d=Trap_Composite2(f1,a,b,n)/Trap_Composite1(f,a,b,n)
    //m
//Result
clc
printf('The value of d is:%f m\n',d) //m
printf('The value of R is %f N',Trap_Composite1(f,a,
```

```
b,n)) //N
```

Chapter 4

Resolution of Non Coplanar Force System

Scilab code Exa 4.1 Resolution Non Coplanar

```
//Initilization Of Variables
F1=20 //lb
F2=15 //lb
F3=30 //lb
F4=50 //lb
//Co-ordinates of Forces
C1=[2;1;6]
C2=[4;-2;5]
C3=[-3;-2;1]
C4=[5;1;-2]
//Calculations
A=sqrt((C1(1,1))^2+(C1(2,1))^2+(C1(3,1)^2))
B=sqrt((C2(1,1))^2+(C2(2,1))^2+(C2(3,1)^2))
C=sqrt((C3(1,1))^2+(C3(2,1))^2+(C3(3,1)^2))
D=sqrt((C4(1,1))^2+(C4(2,1))^2+(C4(3,1)^2))
//Calculations for cos(thetax),cos(thetay) and cos(
    thetaz)
theta1=[C1(1,1)/A;C1(2,1)/A;C1(3,1)/A]
theta2=[C2(1,1)/B;C2(2,1)/B;C2(3,1)/B]
```

```
theta3=[C3(1,1)/C;C3(2,1)/C;C3(3,1)/C]
theta4=[C4(1,1)/D;C4(2,1)/D;C4(3,1)/D]
//Calculations for forces (in form of force vectors)
Fa=F1*theta1 //lb
Fb=F2*theta2 //lb
Fc=F3*theta3 //lb
Fd=F4*theta4 //lb
Fx=Fa(1,1)+Fb(1,1)+Fc(1,1)+Fd(1,1) //lb
Fy=Fa(2,1)+Fb(2,1)+Fc(2,1)+Fd(2,1) //lb
Fz=Fa(3,1)+Fb(3,1)+Fc(3,1)+Fd(3,1) //lb
R=sqrt(Fx^2+Fy^2+Fz^2) //lb
thetax=acosd(Fx/R) //degrees
thetay=180-acosd(Fy/R) //degrees
thetaz=acosd(Fz/R) //degrees
//Result
clc
printf('The resultant of the force system is %f lb \n',R)
printf('The angle of the resultant with respect to x y and z axes are %f, %f and %f degrees respectively',thetax,thetay,thetaz)
```

Scilab code Exa 4.2 Resolution Non Coplanar

```
//Initilization of variables
F=[20;-10;30] //N
//co-ordinates in meters
a=2 //m
b=4 //m
c=7 //m
d=3 //m
e=2 //m
f=4 //m
//Calculations
R=F(1,1)+F(2,1)+F(3,1) //N
```

```
M_o=F(1,1)*a+F(2,1)*b+F(3,1)*c //N-m
x=M_o/R //m
M_x=-F(3,1)*e-F(1,1)*d+F(2,1)*f //N-m
z=-M_x/R //m
//Result
clc
printf('The resultant is %f N \n',R)
printf('The moment about point O is %f N-m \n',M_o)
printf('The position of R is at %f from origin m \n'
    ,x)
printf('The moment is %f N-m\n',M_x)
printf('The z co-ordinate is %f m',z)
```

Scilab code Exa 4.3 Resolution Non Coplanar

```
//Initilization of variables
F=[100;50;-150] //Force vector N
a=2 //m
b=2 //m
c=3 //m
d=2 //m
e=4 //m
f=8 //m
//Calculations
R=F(1,1)+F(2,1)+F(3,1) //N
M_x=-F(1,1)*a+F(2,1)*b-F(3,1)*c //N-m
M_z=F(1,1)*d+F(2,1)*e+F(3,1)*f //N-m
C=sqrt(M_x^2+M_z^2) //N-m
thetax=atand(-M_x/M_z) //degrees
//result
clc
printf('The resultant is %f N \n',R)
printf('The moment about x axis is %f N.m \n',M_x)
printf('The moment about z axis is %f N.m\n',M_z)
printf('The couple acting is %f N.m\n',C)
```

```
printf('The trace makes an angle with x axis of %f
    degrees',thetax)
```

Scilab code Exa 4.4 Resolution Non Coplanar

```
//Initilization of variables
x1=-2
y1=2
z1=-2
x2=3
y2=0
z2=-4
x3=3
y3=2
z3=2
F1=40 //lb
F2=30 //lb
F3=20 //lb
Mxm=[-92.4,-48,-19.4]
Mym=[-46.2,72,9.8]
Mzm=[46.2,-36,19.4]
//Calculations
mag1=sqrt(x1^2+y1^2+z1^2)
mag2=sqrt(x2^2+y2^2+z2^2)
mag3=sqrt(x3^2+y3^2+z3^2)
thetax1=acosd(x1/mag1) //degrees
thetay1=acosd(y1/mag1) //degrees
thetaz1=acosd(z1/mag1) //degrees
thetax2=acosd(x2/mag2) //degrees
thetay2=acosd(y2/mag2) //degrees
thetaz2=acosd(z2/mag2) //degrees
thetax3=acosd(x3/mag3) //degrees
thetay3=acosd(y3/mag3) //degrees
thetaz3=acosd(z3/mag3) //degrees
//Now we will define all the components in terms of
```

```
matrices for simplicity of computation
F=[F1,F2,F3] //lb
COSthetax=[cosd(thetax1);cosd(thetax2);cosd(thetax3)]
COSthetay=[cosd(thetay1);cosd(thetay2);cosd(thetay3)]
COSthetaz=[cosd(thetaz1);cosd(thetaz2);cosd(thetaz3)]
Fx=F*COSthetax //lb
Fy=F*COSthetay //lb
Fz=F*COSthetaz //lb
R=sqrt(Fx^2+Fy^2+Fz^2) //lb
thetax=acosd(Fx/R) //degrees
thetay=acosd(Fy/R) //degrees
thetaz=acosd(Fz/R) //degrees
//Moment calculations
Mx=Mxm(1)+Mxm(2)+Mxm(3) //lb-ft
My=Mym(1)+Mym(2)+Mym(3) //lb-ft
Mz=Mzm(1)+Mzm(2)+Mzm(3) //lb-ft
C=sqrt(Mx^2+My^2+Mz^2) //lb-ft
//Direction cosines
PHIx=acosd(Mx/C) //degrees
PHIy=acosd(My/C) //degrees
PHIz=acosd(Mz/C) //degrees
//Result
clc
printf('The result of the force is %f lb\n',R)
printf('The angles with respect to X-Axis,Y-Axis and Z-axis are %f ,%f and %f degrees respectively\n',thetax,thetay,thetaz)
printf('The magnitude of resultant couple is %f lb-ft\n',C)
printf('The angles are as follows Cosphix=%f degrees,Cosphix=%f degrees and Cosphiz=%f degrees',PHIx,PHIy,PHIz)
```

Scilab code Exa 4.5 Resolution Non Coplanar

```
//Initilization of variables
F=[150;90;160] //lb force vector kind of decleration
//Co-ordinates defined as [x;y;z] all the co-ordinates are in feet
C_1=[2;0;0]
C_2=[0;0;1]
C_3=[0;-2;-1]
C_4=[-1;0;-1]
//Calculations
A=C_2-C_1
B=C_4-C_3
F1=(F(1,1)*A)/sqrt(A(1,1)^2+A(2,1)^2+A(3,1))
F2=(F(2,1)*B)/sqrt(B(1,1)^2+B(2,1)^2+B(3,1))
R=F1+F2
//The calculations for this is done differently
C1x=det([1 0 0;C_1(1,1) C_1(2,1) C_1(3,1);F1(1,1) F1(2,1) F1(3,1)])
C1y=det([0 1 0;C_1(1,1) C_1(2,1) C_1(3,1);F1(1,1) F1(2,1) F1(3,1)])
C1z=det([0 0 1;C_1(1,1) C_1(2,1) C_1(3,1);F1(1,1) F1(2,1) F1(3,1)])
C2x=det([1 0 0;C_3(1,1) C_3(2,1) C_3(3,1);F2(1,1) F2(2,1) F2(3,1)])
C2y=det([0 1 0;C_3(1,1) C_3(2,1) C_3(3,1);F2(1,1) F2(2,1) F2(3,1)])
C2z=det([0 0 1;C_3(1,1) C_3(2,1) C_3(3,1);F2(1,1) F2(2,1) F2(3,1)])
C3z=[0;0;F(3,1)]
sC1=[C1x;C1y;C1z]
sC2=[C2x;C2y;C2z]
C=sC1+sC2+C3z
//Result
```

```
clc
printf('The resultant force couple is %fi%fj+%fk lb-ft',C(1,1), C(2,1), C(3,1))
```

Chapter 5

Equilibrium of Coplanar Force System

Scilab code Exa 5.1 Equilibrium of CFS

```
//Initilization of variables
D=[6/sqrt(40) -4/sqrt(20);2/sqrt(40) 2/sqrt(20)]
B=[0;25] //lb
//Calculations
X=inv(D)*B
//Result
clc
printf('The tension in  cable AB is %flb and the
   tension in cable AC is %f lb',X(2),X(1))
```

Scilab code Exa 5.2 Equilibrium of CFS

```
//Initilization of variables
F1=100 //lb
R=16 //in
//Calculations
```

```
theta=asind(14/16) //degrees
N=100/sind(theta) //lb
P=N*cosd(theta) //lb
//Result
clc
printf('The value of normal reaction offered is %flb
    and the push required is %f lb',N,P)
```

Scilab code Exa 5.3 Equilibrium of CFS

```
//Initilization of variables
L=20 //m
M=1200 //kg
g=9.81 //m/s^2
H=10 //m
//Calculations
AB=sqrt(L^2-H^2) //Applying Pythagoras Theorem
costheta=17.3/20
F1=M*g*H/AB //N
F2=M*g/costheta //N
//Result
clc
printf('Force F1 is %f N and Force F2 is %f N',F1,F2
  )
//Decimal accuracy causes discrepancy in answers
   compared to the textbook answers
```

Scilab code Exa 5.4 Equilibrium of CFS

```
//Initilization of variables
Fx=1000 //lb
Fy=1000 //lb
costheta=9/15
```

```
cosbeta=12/15
sintheta=4/5
sinbeta=3/5
//Calculations
//Matrix solution
A=[costheta -cosbeta;sintheta sinbeta]
B=[-1000;1000]
X=inv(A)*B
//Result
clc
printf('Thus force in AB is %i lb compression and BC
    has %i lb compression',X(1),X(2))
```

Scilab code Exa 5.5 Equilibrium of CFS

```
//Initilization of variables
w=10 //lb/ft
L=12 //ft
theta=30 //degrees
//Calculation
//Matrix Calculations
A=[cosd(30) -cosd(30);sind(30) sind(30)]
B=[0;120]
X=inv(A)*B
//Result
clc
printf('The tension in the cable is %i lb and the
   reaction at B is %i lb', X(1) ,X(2))
```

Scilab code Exa 5.6 Equilibrium of CFS

```
//Initilization of variables
W1=40 //lb
```

```
W2=30 //lb
theta1=30 //degrees
//Calculations
//Summing the forces parallel to 30 degree plane
T=40*sind(theta1)
theta=asind(T/W2)
//Result
clc
printf('The tension in the cable is %flb and the
    angle theta is %f degrees',T,theta)
```

Scilab code Exa 5.8 Equilibrium of CFS

```
//Initilization of variables
F1=125 //N
F2=200 //N
F3=340 //N
F4=180 //N
x1=4 //m
x2=3 //m
x3=10 //m
x4=15 //m
x5=17 //m
//Calculations
Rb=(-F1*x1+F2*x2+F3*x3+F4*x4)/x5//moment about point
    A
Ra=(F1*(x1+x5)+F3*(x5-x3)+F2*(x5-x2)+F4*(x5-x4))/x5
    //moment about point B
//Result
clc
printf('The reaction at A is %fN and reaction at B
    is %fN',Ra,Rb)
```

Scilab code Exa 5.9 Equilibrium of CFS

```
//Initilization of variables
F1=1000 //lb
F2=1200 //lb
F3=2000 //lb
x1=1 //ft
x2=7 //ft
x4=2 //ft
x3=6 //ft
//Calculation
//Equilibrium equations
Rn=(F3*(x1+x2+x3)+F2*(x1+x2)+F1*x1)/(x1+x3+x2+x4)//
    Moment about point M
Rm=(F1*(x2+x3+x4)+F2*(x3+x4)+F3*x4)/(x1+x2+x3+x4)//
    Moment about point N
//Result
clc
printf('The reaction at M is %flb and reaction at N
    is %flb',Rm,Rn)
//Decimal Accuracy causes discrepancy in answers
    between computation and textbook
```

Scilab code Exa 5.10 Equilibrium of CFS

```
//Initilization of variables
P=10 //kg
g=9.81 //m/s^2
//Calculations
//equilibrium at fig b
T1=P*g/2 //N
//equilibrium at fig c
T2=T1/2 //N
//equilibrium at fig d
P=T2
```

```
//Result
clc
printf('The force P is %fN',P)
```

Scilab code Exa 5.11 Equilibrium of CFS

```
//Initilization of variables
k=20 //lb/in
w=20 //lb/ft
x1=4 //ft
x2=10 //ft
x3=8 //ft
x4=6 //ft
x5=9 //ft
F1=1920 //lb.rad
F2=3360 //lb.rad
//calculations
theta=(w*x2*x5)/(F1*x3+F2*(x3+x4)) //radians
FB=F1*theta
FC=F2*theta
A=(w*x2)-FB-FC
//Result
clc
printf('The force in spring B is %flb and spring C
    is %f lb and the reaction at A is %f lb',FB,FC,A)
```

Scilab code Exa 5.12 Equilibrium of CFS

```
//Initilization of variables
L=3.8 //m
w=10 //kg/m
P=1000 //N
t=0.8 //m
```

```
g=9.81 //m/s^2
//Calculations
Gf=L*w*g //N
A=(P*L+Gf*L*0.5)/t //N Taking moment about point B
B=(P*(L-t)+Gf*(0.5*L-t))/t //N Taking moment about
    point A
//Result
clc
printf('The reaction at point A and B are %f N and
    %f N respectively',A,B)
//Decimal accuracy causes discrepancy in answers
    compared to the textbook
```

Scilab code Exa 5.13 Equilibrium of CFS

```
//Initilization of variables
Wa=400 //lb
Wb=200 //lb
theta=30 //degrees
//Calculations
Ta=Wa*sind(theta) //lb
Tb=Wb*sind(theta) //lb
//Taking moment about point O
P=(Tb*12+Ta*6)/24 //lb
//Result
clc
printf('The value of Ta is %f lb and that of Tb is
    %f lb, also P is %f lb',Ta,Tb,P)
```

Scilab code Exa 5.15 Equilibrium of CFS

```
//Initilization of variables
```

```
F=[5;2;3;1.5] //kN Forces are defined as a cloumn
    matrix
theta=[90;60;45;80] //degrees angles are also
    defined as a column matrix
d=[2;6;13;17] //distances from point C of each force
c=[17;15;11;4] //distance form point D of each force
//Calculations
//Summing horizontal forces
Ch=-F(3,1)*cosd(theta(3,1))+F(2,1)*cosd(theta(2,1))+
    F(4,1)*cosd(theta(4,1)) //kN "which indidcates
    that Ch acts to the left instead of the assumed"
//Taking moment about point C
D=(F(1,1)*d(1,1)+F(2.1)*sind(theta(2,1))*d(2,1)+F
    (3,1)*sind(theta(3,1))*d(3,1)+F(4,1)*sind(theta
    (4,1))*d(4,1))/d(4,1) //kN
//Taking moment about point D
Cv=(F(1,1)*c(2,1)+F(2,1)*sind(theta(2,1))*c(3,1)+F
    (3,1)*sind(theta(3,1))*c(4,1))/c(1,1) //kN
//Result
clc
printf('The values of Ch,D and Cv are %f kN,%f kN
    and %f kN respectively',Ch,D,Cv)
```

Scilab code Exa 5.16 Equilibrium of CFS

```
//Initilization of variables
w=100 //N/m
F1=200 //N
M=500 //N.m
Lw=2 //m
//Distance from point A
d=[1;2;3;4;5] //m
//Distance from point B
b=[5;4;3;2;1] //m
//Calculations
```

```
//Taking moment aboout point A
Ra=(w*Lw*d(1,1)+F1*d(3,1)-M)/d(4,1) //N
//Taking moment about point B
Rb=(w*Lw*b(3,1)+F1*b(5,1)+M)/b(2,1) //N
//Result
clc
printf('The value of reaction at A and B are %f N
   and %f N respectively',Ra,Rb)
```

Scilab code Exa 5.17 Equilibrium of CFS

```
//Initilization of variables
L=14 //feet
W=18 //lb
theta1=60 //degrees
theta2=30 //degrees
L1=10 //ft
//Calculations
//Taking moment about point B
Rd=(W*(L/2)*cosd(theta1)*cosd(theta2))/L1 //lb
//Summing all the forces in the horizontal direction
T=Rd*cosd(theta2) //lb
//Result
clc
printf('The value of Rd and T is %f lb and %f lb
   respectively',Rd,T)
```

Scilab code Exa 5.18 Equilibrium of CFS

```
//Initilization of variables
theta=[60;60;45] //degrees theta has been defined in
    the form of a matrix
d=[4.46;3.54;2] //feet defined as a matrix
```

```
F=400 //lb
//Calculations
//Taking moment about point A
Re=(F*(8-d(2,1)))/8 //lb
Ra=400-Re //lb here i have used the summation of forces in the vertical direction
//Taking moment about point B
Dv=(-F*3.644)/5.77 //lb
//Taking moment about point D
Bv=(F*2.126)/5.77 //lb
//Taking summation of forces in the vertical direction
Cv=-223-Dv //lb
//Taking moment about point D
Ch=(223*d(3,1)*cosd(theta(3,1))-Cv*5.173*cosd(theta(3,1)))/(5.173*sind(theta(3,1))) //lb
//Taking summation of forces in the horizontal direction
Dh=-Ch //lb
//Taking sum of forces in horizontal direction
Bh=-Dh //lb
//Result
clc
printf('The Floor reactions are \n')
printf('Ra=%f lb and Re=%f lb\n',Ra,Re)
printf('Pin reaction at C on CE are\n')
printf('Ch=%f lb and Cv=%f lb\n',Ch,Cv)
printf('The pin reactions at B on AC are\n')
printf('Bh=%f lb and Bv=%f lb',Bh,Bv)
```

Scilab code Exa 5.19 Equilibrium of CFS

```
//Initilization of variables
r=0.5 //m
m=10 //kg
```

```
g=9.81 //m/s^2
theta=60 //degrees
//Calculations
//Due to symmetry the reaction will be shared by the structure
A=m*g*0.5 //N
B=A //N
//Vertical forces summed
N1=m*g/(2*sind(theta/2)) //N
//Taking moment about point C
T=(N1*0.866+B*sind(theta*0.5))/(1.5*cosd(theta*0.5))
//Result
clc
printf('The value of N1 and T are %f N and %f N respectively',N1,T)
```

Chapter 6

Equilibrium of Non Coplanar Force System

Scilab code Exa 6.1 Equilibrium of NCFS

```
//Initilization of variables
H=30 //ft
F=150 //lb
theta1=10 //degrees
theta2=30 //degrees
theta3=60 //degrees
//Calculations
//Matrix solution of simultaneous equations
X=[cosd(theta3)*sind(theta2) -cosd(theta3)*sind(
    theta2);cosd(theta3)*cosd(theta2) cosd(theta3)*
    cosd(theta2)]
Y=[0;F*cosd(theta1)]
R=inv(X)*Y
//To find P,sum the forces vertically along the y-
    axis
P=F*sind(theta1)+2*R(1,1)*sind(theta3) //lb
    Copression
//Result
clc
```

```
printf('The value of A and B is %f lb and that of P
    is %f lb',R(1,1),P)
```

Scilab code Exa 6.2 Equilibrium of NCFS

```
//Initilization of variables
F=150 //lb
theta1=10 //degrees
theta2=30 //degrees
theta3=60 //degrees
//Calculations
A=[-cosd(theta3)*cosd(theta2);-sind(theta3);cosd(
    theta3)*sind(theta2)]
B=[-cosd(theta3)*cosd(theta2);-sind(theta3);cosd(
    theta3)*sind(theta2)]
//150lb force is actually a vector
F_v=[F*cosd(theta1);F*sind(theta1);0] //lb
//Postion vector relative to C
r=[0;30;0]
//Moment about point C is zero
//solution by matrix
X=[7.5 -7.5;13 13]
Y=[0;4470]
R=inv(X)*Y
A=R(1,1) //lb
B=R(2,1) //lb
//Summing forces in y direction
Cy=0.866*A+0.866*B+25.9 //lb
//Result
clc
printf('The value of A and B is %f lb and that of Cy
     is %f lb',A,Cy)
```

Scilab code Exa 6.3 Equilibrium of NCFS

```
//Initililization of variables
m=6.12 //kg
g=9.81//m/s^2
x1=3 //m
x2=4 //m
y=6 //m
z1=2 //m
z2=4 //m
AB=5
//Calculations
AD=sqrt(x1^2+y^2+z1^2)
AC=sqrt(x2^2+z1^2)
//Sum of forces in the y direction
T1=(m*g*AD)/6 //N
//sum of forces in the x and z direction
//Matrix solution of the folllowing simultaneous
    equations
X=[x2/AC, -x1/AB;-z1/AC, z2/AB]
Y=[T1*(x1/AD);T1*(z1/AD)]
R=inv(X)*Y
T2=R(1) //N
T3=R(2) //N
//Result
clc
printf('The values of T1,T2 and T3 are %f N,%f N and
    %f N respectively',T1,T2,T3)
```

Scilab code Exa 6.4 Equilibrium of NCFS

```
//Intilization of variables
F=[0;60;0] //Force defined as a matrix
t1=[-3/7;6/7;-2/7] //Tension defined as a matrix
t2=[4/4.47;0;-2/4.47] //tension defined as a mtrix
```

```
t3=[-3/5;0;4/5] //Tension defined as a matrix
//Calculations
//Summation of forces in the y-direction
T1=F(2,1)/t1(2,1) //N
//Summation of forces in the x-direction and z direction
M1=[t2(1,1) t3(1,1);t2(3,1) t3(3,1)]
M2=-1*[t1(1,1)*T1;t1(3,1)*T1]
R=inv(M1)*M2
//Result
clc
printf('The tension in the strings are T1=%f,T2=%f and T3=%f',T1,R(1),R(2))
```

Scilab code Exa 6.5 Equilibrium of NCFS

```
//Initilization of variables
m=80 //kg
g=9.81 //m/s^2
//Co-ordinates of points in Meters
A=[1,3,0]
B=[3,3,-4]
C=[4,3,0]
D=[2,0,-1]
//Calculations
//Tension in DC will be
a=[C(1)-D(1),C(2)-D(2),C(3)-D(3)]
h=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2+(C(3)-D(3))^2)
c=a/h
//Unit vector calculations
e=[B(1)-A(1),B(2)-A(2),B(3)-A(3)]
v=sqrt((B(1)-A(1))^2+(B(2)-A(2))^2+(B(3)-A(3))^2)
e_ab=e/v
//Position vector AD
r_ad=[D(1)-A(1),D(2)-A(2),D(3)-A(3)]
```

```
//Moment Calculations
O=[1,0,0;1,-3,-1;0,-m*g,0]
P=[0,1,0;1,-3,-1;0,-m*g,0]
Q=[0,0,1;1,-3,-1;0,-m*g,0]
C1=[1,0,0;1,-3,-1;2,3,1]
C2=[0,1,0;1,-3,-1;2,3,1]
C3=[0,0,1;1,-3,-1;2,3,1]
rxF1=[det(O),det(P),det(Q)]
rxF2=[(det(C1)/h),(det(C2)/h),(det(C3)/h)]
//Final Moment calculations
rxF=rxF1+rxF2
//Taking dot product
dot1=e_ab.*rxF
dot2=e_ab.*rxF2
//equating dot product to zero to obtain C
C=-(dot1(1)+dot1(3))/dot2(3)
//Result
clc
printf('The tension in CD is %f N',C)
```

Scilab code Exa 6.6 Equilibrium of NCFS

```
//Initilization of variables
w=200 //lb
Dh=4 //ft
//Calculation
theta=atand(2/Dh) //degrees
T=w/(3*cosd(theta)) //lb
//result
clc
printf('The Tension in each rope is %f lb\n Theta=%f
    degrees',T,theta)
```

Scilab code Exa 6.7 Equilibrium of NCFS

```
//Initilization of variables
F=[100,0,0] //N
CE=5 //m
BC=sqrt(34) //m
AC=sqrt(41) //m
//Calculations
//solving as a matrix for system of linear equations
A=[3/BC,-4/AC,0;0,0,(6*4)/CE;-3/BC,-3/AC,-3/CE]
B=[0;F(1)*4;-F(1)]
C=inv(A)*B
//Result
clc
printf('The forces F1 F2 and F3 are as %f N %fN and %fN respectively\n',C(1),C(2),C(3))
printf('Here F3 is compression assumed and rest are Tension')
```

Scilab code Exa 6.8 Equilibrium of NCFS

```
//Initilization of variables
F=[100,0,0] //N
CE=5 //m
BC=sqrt(34) //m
AC=sqrt(41) //m
//Calculations
//solving as a matrix for system of linear equations
A=[3/BC,-4/AC,0;0,0,(6*4)/CE;-3/BC,-3/AC,-3/CE]
B=[0;F(1)*4;-F(1)]
C=inv(A)*B
//Result
clc
printf('The forces F1 F2 and F3 are as %f N %fN and %fN respectively\n',C(1),C(2),C(3))
```

```
printf('Here F3 is compression assumed and rest are
    Tension')
```

Scilab code Exa 6.9 Equilibrium of NCFS

```
//Initilization of variables
//here forces will be defines as matrices along with
    their co−ordinates
//Force in N and co−ordinates in mm
F1=[30 200 300]
F2=[10 400 200]
F3=[20 200 500]
F4=[50 400 500]
//Calculations
//solving as system of linear equations
A=[1 1 1;-600 -600 0;0 600 600]
B=[F1(1)+F2(1)+F3(1)+F4(1);-(F3(1)*F3(3)+F1(1)*F1(3)
    +F4(1)*F4(3)+F2(1)*F2(3));-(-F3(1)*F3(2)-F1(1)*F1
    (2)-F4(1)*F4(2)-F2(1)*F2(2))]
C=inv(A)*B
//Result
clc
printf('The reactions are as R1=%fN, R2=%fN and R3=
    %fN',C(1),C(2),C(3))
```

Scilab code Exa 6.11 Equilibrium of NCFS

```
//Initilization of variables
w=50 //lb wind load
W=60 //lb weight of door
//Calculations
//Calculation as system of linear equations
A=[0 0 33;1 1 -1;28 10 -28]
```

```
B=[50*18;-50;-50*24]
C=inv(A)*B
P=C(3)/cosd(20)
D=[-28 -10;1 1]
E=[1080-(28*(P*sind(20)));P*sind(20)]
F=inv(D)*E
By=60
//Result
clc
printf('The forces are as follows: \n')
printf('Az=%flb Bz=%flb Pz=%flb Ax=%flb Bx=%flb P=
    %flb By=%flb',C(1),C(2),C(3),F(1),F(2),P,By)
```

Scilab code Exa 6.12 Equilibrium of NCFS

```
//Initilization of variables
m=1 //kg
g=9.81 //m/s^2
t1=45 //degrees
t2=30 //degrees
//Calculations
//Solving as system of linear equations
A=[1 0 -cosd(t1) 0;0 1 0 3/5;-5 g*m*cosd(t1)*cosd(t2
    ) 0 0;-1 0 0 4/5]
B=[0;g*m;g*m*5*cosd(t1)*cosd(t2);0]
C=inv(A)*B
//Result
clc
printf('The forces are Nb=%fN Nc=%fN Tc=%fN Tb=%fN',
    C(1),C(2),C(3),C(4))
```

Scilab code Exa 6.13 Equilibrium of NCFS

```
//Initilization of variables
l=200 //lb
//Calculations
P=l*5/12 //lb
//Solving as system of linear equations
A=[0 -36 0 0;0 0 0 36;0 0 1 1;1 1 0 0]
B=[-P*cosd(25)*48;l*20+P*sind(25)*48;P*sind(25)+200;
    P*cosd(25)]
C=inv(A)*B
//Result
clc
printf('The forces are Az=%flb Bz=%flb Ay=%flb By=
    %flb',C(1),C(2),C(3),C(4))
```

Scilab code Exa 6.14 Equilibrium of NCFS

```
//Initilization of variables
A=80 //lb
B=40 //lb
C=60 //lb
l1=2 //in
l2=4 //in
l3=6 //in
l4=9 //in
l5=3 //in
l6=7 //in
//Calculations
P=-(-A*l1+B*l2-C*l2)/l1
By=-(A*l3+P*l3)/l4
Ay=(-A*l5-P*l5)/l4
Bz=-(-C*l1-B*l1)/l4
Az=(C*l6+B*l6)/l4
//Result
clc
printf('The forces are Ay=%flb By=%flb Az=%flb Bz=
```

```
%flb',Ay,By,Az,Bz)
```

Scilab code Exa 6.15 Equilibrium of NCFS

```
//Initilization of variables
W=138 //lb
w=80 //lb
//Calculations
u=sqrt(3*3+4*4+6*6)
a=[-3/u 4/u -6/u]
v=sqrt(3*3+3*3+3*3)
c=[3/v 3/v -3/v]
P=[1 0 0;0 0 8;0 -W 0]
Q=[0 0 1;0 0 8;0 -W 0]
R=[1 0 0;0 0 4;0 -w 0]
S=[0 0 1;0 0 4;0 -w 0]
T=[1 0 0;0 0 6;a(1) a(2) a(3)]
U=[0 1 0;0 0 6;a(1) a(2) a(3)]
V=[1 0 0;0 0 3;c(1) c(2) c(3)]
Y=[0 1 0;0 0 3;c(1) c(2) c(3)]
//Solving for A and C
MAT1=[det(T) det(V);det(U) det(Y)]
MAT2=[det(P)+det(R);0]
res=-inv(MAT1)*MAT2
A=[a(1)*res(1) a(2)*res(1) a(3)*res(1)]
C=[c(1)*res(2) c(2)*res(2) c(3)*res(2)]
E=[-(A(1)+C(1)) -(-w-W+A(2)+C(2)) -(A(3)+C(3))]
//Result
clc
printf('The force vectors are as follows\n')
printf('A=%fi+%fj%fk lb and C=%fi+%fj%fk lb also Ex=%f Ey=%flb Ez=%flb',A(1),A(2),A(3),C(1),C(2),C(3),E(1),E(2),E(3))
//Decimal accuracy causes discrepancy in the answers
```

Chapter 7

Trusses and Cables

Scilab code Exa 7.1 Truss and Cable

```
//Initilization of variables
F1=2000 //lb
F2=4000 //lb
l1=10 //ft
l2=30 //ft
l3=20 //ft
l4=40 //ft
t=60 //degrees
//Calculations
//Taking moment about point B and A
Ra=(F1*l2+F2*l1)/l4
Rb=(F2*l2+F1*l1)/l4
//Consider fig 7-4(c)
A=[1 -cosd(t);0 -sind(t)]
B=[0;-2500]
C=inv(A)*B
//Consider figure 7-4(d)
A1=[1 cosd(t);0 -sind(t)]
B1=[-C(2)*cosd(t);-C(2)*sind(t)+F1]
C1=inv(A1)*B1
//Consider figure 7-4(e)
```

```
CD=577
CE=C(1)+C1(2)*cosd(t)+CD*cosd(t)
//Consider figure 7-4(f)
DE=Rb/sind(t)
//Result
clc
printf('The reactions are Ra=%flb and Rb=%flb\n',Ra,
   Rb)
printf('Force in member AB=%flb and AC=%flb\n',C(2),
   C(1))
printf('Force in member BC=%flb and BD=%flb\n',C1(2)
   ,C1(1))
printf('Force in member CD=%flb and CE=%flb\n',CD,CE
   )
printf('Force in member DE=%flb',DE)
//Decimal Accuracy causes discrepancy in answers
```

Scilab code Exa 7.2 Truss and Cable

```
//Initilization of variables
s=4 //m length of sides
l=2 //kN load acting on each node
r=7 //kN by inspection reaction at A
//Calculation
//Taking Moment about point G
FH=(-r*12+2*10+2*6+2*2)/(2*tand(60)) //kN
   Compressive
//Taking moment about point H
GI=(r*14-2*12-2*8-2*4)/(2*tand(30)) //kN Tension
//Summing forces in the vertical direction
HG=-(r-(l*3))/sind(60) //kN Compression
//Taking moment about point J yields
IK=(-2*4-2*8+r*10)/(2*tand(60)) //kN
//Result
clc
```

```
printf('The value of the forces in the components
   are as follows\n')
printf('FH=%fkN,GI=%fkN,HG=%fkN and IK=%fkN\n',FH,GI
   ,HG,IK)
printf('The answer in the text book for GI is wrong'
   )
```

Scilab code Exa 7.3 Truss and Cable

```
//Initilization of variables
theta=30 //degrees
EF=40000 //lb
l=36 //feet
//Calculation
//Taking moment about point D and setting EF=40000
   lbs
P=-(EF*sind(theta)*l)/l //lb
//Result
clc
printf('The maximum value of P is %flb\n',P)
printf('The negative sign indicates the downward
   direction')
```

Scilab code Exa 7.4 Truss and Cable

```
//Initilization of variables
l=12 //m
theta1=30 //degrees
F1=1000 //N
F2=2000 //N
//Calculation
FG=l*cosd(theta1) //m
DG=(l+(l/2))/cosd(theta1) //m
```

```
//Taking moment about point G
A=(F1*l+F2*FG+F1*DG)/(l*3) //N
//Summing forces in horizontal direction
G_x=(2*F1+F2)*sind(theta1) //N
//Summing forces in the vertical direction
G_y=(2*F1+F2)*cosd(theta1)+F1-A //N
//Taking moment about point C
BD=-(A*l)/(l/2) //N
//Taking moment about point D
CE=(A*(l+(l/2)))/FG //N
theta=atand((l/2)/FG) //degrees
//Summing forces in the vertical direction
CD=(A+(BD*cosd(60)))/cosd(theta) //N
//Result
clc
printf('The values of  the forces are as follows\n')
printf('A=%fN,G_x=%fN,G_y=%fN,BD=%fN,CE=%fN and CD=
    %fN',A,G_x,G_y,BD,CE,CD)
//Decimal Accuracy causes discrepancy in answers
```

Scilab code Exa 7.5 Truss and Cable

```
//Initilization of variables
A=2000 //lb
E=2000 //lb
theta=60 //degrees
theta1=30 //degrees
//Sign convention positive means Tension and
    negative means Compression
//Taking sum of forces along x and y direction in
    fig7-13
AB=-A/sind(theta) //lb
AG=-AB*cosd(theta) //lb
//Taking sum of forces along x and y direction in
    fig7-14
```

```
BG=((-AB*cosd(theta1))-1000)/(cosd(theta1)) //lb
BC=((AB*sind(theta1))-(BG*sind(theta1))) //lb
//Taking sum of forces along x and y direction in fig7-15
GC=-(BG*sind(theta))/sind(theta) //lb
GF=AG+BG*cosd(theta)-GC*(cosd(theta)) //lb
//By symmetry of structure
DE=AB //lb
FE=AG //lb
DF=BG //lb
CD=BC //lb
//Result
clc
printf('The forces in the truess are\n')
printf('AB=DE=%flb,AG=FE=%flb,BG=DF=%flb,BC=CD=%flb and CG=CF=%flb',AB,AG,BG,BC,GC)
```

Scilab code Exa 7.6 Truss and Cable

```
//Initilization of variables
F=500 //N
A=1000 //N
theta=60 //degrees
l=20 //m
//Calculations
//Taking moment about point G
R_c=(20*3*A+50*F+30*F+10*F)/40 //N
//Returning to fig7-17
//Taking moment about point C
BD=(l*A+(l/2)*F)/(l*sind(theta)) //N
//Taking sum of forces in vertical direction
CD=(A+F-R_c)/sind(theta) //N
//Result
clc
printf('The forces in the members are as follows\n')
```

```
printf('BD=%fN and CD=%fN, Also the reaction at C is
    %fN',BD,CD,R_c)
//Decimal accuracy causes discrepancey in answers
```

Scilab code Exa 7.7 Truss and Cable

```
//Initilization of variables
w=800 //lb/ft
a=600 //ft
d=40 //ft
//Calculations
T=0.5*w*a*(sqrt(1+(a^2/(16*d^2)))) //lb
H=(w*a^2)/(8*d) //lb
//Taking the first two terms of the series
l=a(1+(8/3)*(d/a)^2-(32/5)*0.00002) //ft
//Result
clc
printf('The value of T=%flb and that of H=%flb, Also
    l=%fft',T,H,l)
//Deciaml accuracy causes discrepancy in answers
```

Scilab code Exa 7.8 Truss and Cable

```
//Initilization of variables
l=800*300 //lb
//Calculations
//Summing forces in horizontal and vertical
    direction
theta=atand(40/150) //degrees
H=l/tand(theta) //lb
T_max=sqrt(l^2+H^2) //lb
//Result
clc
```

```
printf('The maximun tension is %flb and H=%flb',
    T_max,H)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 7.9 Truss and Cable

```
//Initilization of variables
//The variable decleration is taken to simplify the
    solution
x1=10 //m
x2=20 //m
m=3 //kg/m
g=9.8 //m/s^2
//For simplicity a1 and a2 values are being
    considered as constant free of H
a_1=sqrt(x1/(m*g*0.5))
a_2=sqrt((x1+x2)/(m*g*0.5))
y=10 //m
//Calculations
H=(300/(a_1+a_2))^2 //N
//Now reconsidering a1 and a2 actual values
a1=a_1*sqrt(H) //m
a2=a_2*sqrt(H) //m
//Theta calculations
x=a1
theta=atand(2*y/x)
//T calculations
T=sqrt((864*a2^2)+H^2) //N
//result
clc
printf('The tension in the cable is %fN',T)
```

Scilab code Exa 7.10 Truss and Cable

```
//Initilization of variables
T=140000 //N
w=2000 //N/m
a=20 //m
//Calculations
//Calculation step by step
lhs=(140000*2)/(2000*20)
d=sqrt(1/((((lhs^2)-1)*16)/(20^2))) //m
l=a(1+(8/3)*(d/a)^2) //m
//Result
clc
printf('The sag in the cable is %fm and the required
    length is %fm',d,l)
```

Scilab code Exa 7.11 Truss and Cable

```
//Initilization of variables
w=10/16 //lb/ft
a=80 //ft
P=500 //lb
//Calculations
lhs=(P*2)/(w*a)
d=sqrt(1/((((lhs^2)-1)*16)/(80^2))) //ft
//Result
clc
printf('The sag in the cable is %fft',d)
```

Scilab code Exa 7.12 Truss and Cable

```
//Initilization of variables
w=0.518 //lb/ft
d=50 //ft
l=500 //ft
```

```
//Plot coding
A=linspace(0,800,9) //defined x axis
B=A+50
C=[50000,(500/(2*100)),(500/(2*200)),(500/(2*300)),(500/(2*400)),(500/(2*500)),(500/(2*600)),(500/(2*700)),(500/(2*800))]
D=cosh(C)
E=[D(1)*A(1),D(2)*A(2),D(3)*A(3),D(4)*A(4),D(5)*A(5),D(6)*A(6),D(7)*A(7),D(8)*A(8),D(9)*A(9)]
plot(A,B,A,E) //plotting two lines on the same plot
//Calculations
//By close observation of plot taking c around 650
//consider c=635
c=635
T_max=w*(c+d) //lb
a=c+d
l=sqrt(4*(a*a-c*c))
//Result
clc
printf('The maximum tension is %flb and length required is %fft',T_max,l)
```

Scilab code Exa 7.13 Truss and Cable

```
//Initilization of variables
m=0.6 //kg/m
l=240 //m
d=24 //m
//Calculations
c=((((1/4)*(l^2))-(24*24)))/(2*d)
T_max=9.8*m*(c+d) //N
a=asinh((l)/(2*c))*576
//Result
clc
printf('The maximun tension is %fN and a=%fm',T_max,
```

a)

Chapter 8

Forces In Beams

Scilab code Exa 8.1 Beams

```
//Initilization of variables
R_A=100 //N
R_B=200 //N
//Calculations
//Shear force at 2m
V=100 //N
//Moment at 2m
M=R_A*2 //N.m
//Result
clc
printf('The shear force at 2m is %fN and the moment
    at 2m is %fN-m',V,M)
```

Scilab code Exa 8.2 Beams

```
//Initilization of variables
//length matrix
L1=[0,4,6] //m
```

```
//Bending moment matrix
B=[0,400,0] //N.m
//Shear force plotting
//Here the left side and right side lengths are considered as close as 4 to keep up with right and left distinctions
L=[0,3.999,4,5.99998,6]
S=[100,100,-200,-200,0]
//Calculations cum Result
subplot(221)
plot(L1,B)
xtitle("Bending Moment Diagram","Span","Bending Moment")
subplot(222)
plot(L,S,L,0)
xtitle("Shear Force Diagram","Span","Shear Force")
```

Scilab code Exa 8.3 Beams

```
//Initilization of variables
w=196 //N/m
M_app=4000 //N.m
L=6 //m
//Calculations
//Taking Moment about Point L and equating it to 0
R_r=(M_app+w*L*L*0.5)/(3*L) //N
//Taking Moment about Point R and equating it to 0
R_l= ((((2*L)+(L/2))*(w*L))-(M_app))/(3*L) //N
//finding point of zero shear
a=R_l/w
//defining x
x0=[0,18]
x=[0,0.5,1,1.5,2,2.5,3,3.5,a,4,4.5,5,5.5,6] //for 0<x<6
x1=[6,12] //for6<x<12
```

```
x2=[12,18] //for 12<x<18
xv=[6,12,18] //specially for shear force
xo=[12.001,12.002] //Straight line plot
//Shear Force Calculations
//Summing forces in vertical direction and equating to 0
V1=R_l-w*x //N for 0<x<6
V2=R_l-w*L //N for 6<x<18
//Bending Moment Calculations
M1=R_l*x-(w*x^2*0.5) //N.m for 0<x<6
M2=R_l*x1-((w*L)*(x1-3)) //N.m for 6<x<12
M3=R_l*x2-((w*L)*(x2-3))+M_app //N.m for 12<x<18
Mo=[-1464.8652,2509.3333]
//Maximum bending moment
M_max=R_l*a*0.5 //N.m
//Plotting
subplot(221)
plot(x,V1,xv,V2,x0,0)
xtitle('Shear Force Diagram',"Span","Shear Force")
subplot(222)
plot(x,M1,x1,M2,x2,M3,x0,0,xo,Mo)
xtitle('Bending Moment Diagram',"Span","Bending Moment")
//Result
clc
printf('The value of reactions are R_l=%fN and R_r=%fN\n',R_l,R_r)
printf('The point of maximum bending moment is %f meters from left support nad maximum bending moment is %fN-m\n',a,M_max)
printf('The bending moment and shear force diagrams have been plotted')
```

Scilab code Exa 8.5 Beams

```
//Initlization of  variables
F1=2000 //lb
w=100 //lb/ft
//Calculations
R_r=(-F1*5+w*14*13)/20 //lb
R_l=(F1*25+w*14*7)/20 //lb
//Shear Force matrix
V=[-2000,-2000,990,990,-410,0] //lb
//Bending Moment matrix
B=[0,-10000,-4060,840,0]
//Length matrix for shear force
X_v=[0,5,5.0001,11,20.89999,20.9]
//Length matrix for bendimg moment
X_b=[0,5,11,19.9,20.9]
//Plotting
subplot(221)
plot(X_v,V,X_v,0)
xlabel("Shear Force Diagram","Span","Shear Force")
subplot(222)
plot(X_b,B,X_b,0)
xlabel("Bending Moment Diagram","Span","Bending Moment")
//Result
clc
printf('The bending Moment and Shear Force diagrams have been plotted\n')
//Note
//The textbook does not specify the span and hence there seems to be a disagreement between the textbook and scilab solution.here the values have just been plotted
```

Chapter 9

Friction

Scilab code Exa 9.1 Friction

```
//Calculations
//Simplifying  equation (3) after substituting value
    of Nb in it we get
//m_u^2+m_u*2*tand(50)-1=0
//Solution of the equation
a=1
b=2*tand(50)
c=-1
g=sqrt(b^2-(4*a*c))
//solution
x1=(-b+g)/(2*a)
x2=(-b-g)/(2*a)
//As x2 does not make any physical sense x1 is the
   answer
//Result
clc
printf('The value of mu is %f',x1)
```

Scilab code Exa 9.3 Friction

```
//Initilization of variables
m=70 //kg
g=9.81 //m/s^2
theta=20 //degrees
//Calculations
//Solving by martix method
//Taking sum along vertical and horizontal direction
    and equating them to zero
A=[sind(theta) 1 0;-cosd(theta) 0 1;0 -1/4 1]
//RHS matrix
R=[m*g;0;0]
ans1=inv(A)*R //force vector N
//Calculation part 2
//Similar solution by matrix method
//Taking moment about point O and summing forces in
   horizontal and vertical direction and equating
   all to zero
B=[4*cosd(theta) 0 0;-cosd(theta) 1 0;sind(theta) 0
   1]
//RHS matrix
J=[m*g*1.5;0;m*g]
ans2=inv(B)*J //force Vector N
//Result
clc
printf('The value of P in first case is %iN and that
    in second case is %iN',ans1(1),ans2(1))
```

Scilab code Exa 9.4 Friction

```
//Initilization of variables
W=200 //lb
Fapp=300 //lb
mu=0.3 //coefficient of friction
theta=30 //degrees
//Calculations
```

```
//Summing forces in the plane parallel to the slope
F=-(W*sind(theta)-Fapp*cosd(theta)) //lb
N1=(W*cosd(theta)+Fapp*sind(theta)) //lb
//Max value obtained
Fprime= mu*N1
//Result
clc
printf('The value of F and N1 are %flb and %flb respectively and the maximum value obtained is %flb',F,N1,Fprime)
```

Scilab code Exa 9.5 Friction

```
//Initilization of variables
mu1=0.2 //coefficient of friction between wedges and A
mu2=1/4 //coefficient of friction between wedges
F=20 //tonnes
//Calculations
//Using the matrix method to solve
//Summing forces in vertical and horizontal direction
A=[1,-(mu1*10+1)/(sqrt(101));0, (10-mu1*1)/sqrt(101)] //force matrix
B=[mu2*F*1000;F*1000] //lb
//Solving both matrices
R=inv(A)*B //lb
//Result
clc
printf('The forces N2 and P are %ilb and %ilb respectively',R(2),R(1))
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 9.6 Friction

```
//Initilization of variables
theta=45 //degrees
mu1=1/4 //coefficient of friction between A and B
mu2=1/3 //coefficient of friction between A and
    Floor
ma=14 //kg
mb=9 //kg
g=9.81 //m/s^2
//Calculations
//Summing forces in vertical direction
Nb=mb*g //N
//Also
Fprimeb=mu1*Nb //N
//Summing forces in direction
T=Fprimeb //N
//Considering the fig(c)
//Summing forces in the horizontal direction and
    vertical direction and solving by matrix method
A=[-cosd(theta) mu2;sind(theta) 1] //N
B=[-Fprimeb;(mb*g+ma*g)] //N
R=inv(A)*B //N
//Result
clc
printf('The value of P and Na are %fN and %fN
    respectively',R(1),R(2))
```

Scilab code Exa 9.7 Friction

```
//Initilization of variables
m1=40 //kg
m2=13.5 //kg
mu=1/3 //coefficient of friction
g=9.81 //m/s^2
```

```
//Calculations
//Solving by substitution
//After simplification we get
x=mu*m2*g
y=mu*(m1*g+m2*g)
theta=atand((x+y)/(m1*g)) //degrees
//Result
clc
printf('The value of the angle is %f degrees',theta)
```

Scilab code Exa 9.8 Friction

```
//Initilization of variables
W=350 //lb
theta=30 //degrees
phi=15 //degrees
//Calculations
//Solving by the matrix method
A=[cosd(theta) sind(phi);-sind(theta) cosd(phi)]
B=[W*sind(theta);W*cosd(theta)]
an=inv(A)*B //lb
//Result
clc
printf('The value of P and R are %flb and %flb
    respectively',an(1),an(2))
```

Scilab code Exa 9.9 Friction

```
//Initilization of variables
theta=45 //degrees
m1=45 //kg
m2=135 //kg
g=9.81 //m/s^2
```

```
mu=0.25 //coefficient of riction
//Calculations
N2=m2*g //N
T=mu*N2 //N
N1=m1*g*cosd(theta) //N
Fprime1=N1*mu //N
P=T+Fprime1-(m1*g*sind(theta)) //N
//Result
clc
printf('The values are N2=%fN,T=%fN,N1=%fN,Fprime1=%fN and P=%fN',N2,T,N1,Fprime1,P)
```

Scilab code Exa 9.10 Friction

```
//Initilization of variables
mu=0.2 //coefficient of friction
F1=150 //lb
F2=100 //lb
theta=60 //degrees
//Calculations
N1=F1*cosd(theta) //lb
T=(mu*N1+(F1*cosd(theta/2))) //lb considering positive
//Equilibrium for 100lb
//Eliminating N2 from both equations
//Taking derivative we get
theta2=atand(mu) //degrees
//Hence P becomes
P=(F2*mu+T)/(cosd(theta2)+(mu*sind(theta2))) //lb
//Result
clc
printf('The minimum value of P is %flb',P)
```

Scilab code Exa 9.11 Friction

```
//Initilization of variables
F=180 //N
m=100 //kg
g=9.81 //m/s^2
mu=0.25 //coeffiecient of friction
//Calculations
//Assuming F2 is maximum
N2=F*2/(1+mu) //N
F2=mu*N2 //N
N1=m*g-F2 //N
F1=F-F2 //N
//Result
clc
printf('The vaules are N2=%fN,F2=%fN,N1=%fN and F1=%fN',N2,F2,N1,F1)
```

Scilab code Exa 9.12 Friction

```
//Initilization of variables
N=4/3 //Normal reaction after solving without the mg term in it
u_N=1/2 //Frictional force without the mg term in it
//Calculations
u=u_N/N //Coefficient of friction
//Result
clc
printf("The frictional co-efficient is %f",u)
//The answer in the textbook and the code differs due to multiplication of fractions
```

Scilab code Exa 9.13 Friction

```
//Initilization of variables
mu_ca=0.3 //ceofficient of friction between copper
   block A and aluminium block B
mu_af=0.2 //coefficient of friction between
   aluminium block B and Floor
ma=3 //kg
mb=2 //kg
g=9.81 //m/s^2
//Calculations
//For A
//Taking sum of forces along X and Y direction
Na=ma*g //N
P=mu_ca*Na //N
//For B
//Taking sum of forces along X and Y direction
Nb=Na+mb*g //N
Fb=mu_ca*Na //N
//Now largest value of friction before slip is
Fprimeb=mu_af*Nb //N
//Now as Fb<F'b hence initial assumption is
   incorrect and P=Fb
P=Fb //N
//Result
clc
printf('The value of force that will cause motion is
    %fN',P)
```

Scilab code Exa 9.15 Friction

```
//Initilization of variables
d_m=2 //in  mean diameter of the screw
p=1/4 //in
mu=0.15 //coefficient of friction
l=2 //ft
L=4000 //lb
```

```
//Calculations
phi=atand(mu) //degrees
beta=atand(p/(%pi*l)) //degrees
//Force to raise the load
P=(L*tand(phi+beta))/(d_m*12) //lb
//Force to lower the load
P2=(L*tand(phi-beta))/(d_m*12) //lb
//Result
clc
printf('The force to raise the load is %flb and to
   lower is %flb',P,P2 )
```

Scilab code Exa 9.16 Friction

```
//Initilization of variables
r_m=2.338 //in
d_m=3.25 //in
mu=0.06 //coefficient of friction
P=1500 //lb
p=1/4 //pitch
//Calculation
phi=atand(mu) //degrees
beta=atand(p/(2*%pi*r_m)) //degrees
M=P*r_m*tand(phi+beta)+mu*P*(d_m/2) //lb.in
//Result
clc
printf('The moment required is %flb-in',M)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 9.17 Friction

```
//Initilization of variables
d=750 //mm diameter
```

```
alpha=%pi //wrap angle    radians
mu=0.25 //coefficient of friction
T_t=200 //N tension on the tight side
//Calculation
T2=T_t/(exp(mu*alpha)) //N
//Result
clc
printf('The tension of the slack side is %fN',T2)
```

Scilab code Exa 9.18 Friction

```
//Initilization of variables
d=635 //mm diameter of the drum
P=178 //N
mu=1/3 //coefficient of friction
l1=100 //mm
l2=660 //mm
theta1=60 //degrees
GD=d/2 //mm
//Calculations
//Taking moment about point C
Tb=(P*(l1+l2))/(l1*sind(theta1)) //N
CD=((d/2)-(l1*cosd(theta1/2)))/sind(theta1/2) //mm
//from fig 9-22(b)
theta=asind(GD/CD) //degrees
//from fig9 -22(c)
w_d=180+30+theta //degrees
w=(w_d)*(%pi/180) //radians
//As Tc is greater than Tb
Tc=Tb*(exp(mu*w)) //N
M=(Tc-Tb)*GD //N.mm
an=M/1000 //N.m
//Result
clc
printf('The braking moment required is %fN-m',an)
```

```
//Note the unit of the final enswer carefully
```

Scilab code Exa 9.19 Friction

```
//Initilization of variables
L=1000 //lb
P=10 //lb
//Calculations
mu=log(L/P)/(4*2*%pi)
//Result
clc
printf('The coefficient of friction is %f',mu)
```

Scilab code Exa 9.20 Friction

```
//Initilization of variables
m=900 //kg
mu=0.2 //coefficient of friction
g=9.8 //m/s^2
//Calculations
T2=m*g/(exp(2*2*%pi*mu)) //N
//Result
clc
printf('The force needed to hold the mass is %fN',T2
  )
```

Scilab code Exa 9.21 Friction

```
//Initilization of variables
d=760 //mm
```

```
W=500 //N
a=0.305 //mm coefficient of rolling resisatnce
r=d/2 //mm
//Calculations
P=(W*a)/r //N
//Result
clc
printf('The force necessary is P=%fN',P)
```

Chapter 10

First Moment and Centroid

Scilab code Exa 10.5 1st Moment and CG

```
//Initilization of variables
r=50 //mm
L1=75 //mm
L2=%pi*r //mm
L3=61.2 //mm
theta1=45 //degrees
theta2=60 //degrees
//Calculations
x_bar=[(L1/2)*cosd(theta1),L1*cosd(theta1)+r,L1*cosd
    (theta1)+100+(L3/2)*cosd(theta2)] //mm
y_bar=[(L1/2)*sind(theta1),L1*sind(theta1)+(2*r)/%pi
    ,(L3/2)*sind(theta2)] //mm
//Centroid Calculations
x=(L1*x_bar(1)+L2*x_bar(2)+L3*x_bar(3))/(L1+L2+L3)
    //mm
y=(L1*y_bar(1)+L2*y_bar(2)+L3*y_bar(3))/(L1+L2+L3)
    //mm
//Result
clc
printf('The centroid is as follows x=%f mm and y=
    %fmm',x,y)
```

Scilab code Exa 10.6 1st Moment and CG

```
//Initilization of variables
theta=75 //degrees
alpha=(150*%pi)/180 //rad
r=1
theta1=30 //degrees
lhor=14 //in
//calculations
a=((2*r)/alpha)*sind(theta) //in
y=-a*sind(90-theta) //in
//Length of arc
l=r*alpha //in
//Slope length calculations
DF=7 //in
AB=DF //in
BC=1 //in
BF=BC*cosd(theta1) //in
FC=BC*sind(theta1) //in
DC=DF+FC //in
EC=DC/cosd(theta1) //in
//Centroid of EC is at G
yslope=0.5*EC*sind(theta1)+BF //in
//Y of composite figure
Y=((2*l*y)+14*-1+(2*EC*yslope))/(2*l+lhor+2*EC) //in
//Result
clc
printf('The centroid is at Y=%f in',Y)
```

Scilab code Exa 10.11 1st Moment and CG

```
//Initilization of variables
a=100 //mm
b=150 //mm
A1=2*10^4 //mm^2
A2=5*10^3 //mm^2
A3=(%pi*(a/2)^2)/2 //mm^2
//Calculations
x=(A1*a+A2*(133.3)-A3*b)/(A1+A2-A3) //mm
y=(A1*a*0.5+A2*(116.66)-A3*((4*a*0.5)/(3*%pi)))/(A1+
    A2-A3) //mm
//Result
clc
printf('The centroidal distances are x=%f mm and y=
    %f mm',x,y)
```

Scilab code Exa 10.16 1st Moment and CG

```
//Initilization of variables
V=[1728*10^3,432*10^3,7.54*10^3]
x_bar=[60,140,60] //mm
y_bar=[30,20,30] //mm
//Calculations
x=(V(1)*x_bar(1)+V(2)*x_bar(2)+V(3)*x_bar(3))/(V(1)+
    V(2)+V(3)) //mm
y=(V(1)*y_bar(1)+V(2)*y_bar(2)+V(3)*y_bar(3))/(V(1)+
    V(2)+V(3)) //mm
z=120 //mm from symmetry
//Result
clc
printf('The centroid is at x=%f mm y=%f mm and z=
    %fmm',x,y,z)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 10.17 1st Moment and CG

```
//Initilization of variables
tx=30 //degrees
ty=45 //degrees
tz=60 //degrees
//Calculations
V=[10,15,25] //in^3
x_bar=[4,12,24] //in
y_bar=[4*cosd(tx),-6*cosd(ty),-4*cosd(tz)]
z_bar=[-4*sind(tx),6*sind(ty),-4*sind(tz)]
//Centroid calculations
x=(V(1)*x_bar(1)+V(2)*x_bar(2)+V(3)*x_bar(3))/(V(1)+
    V(2)+V(3)) //in
y=(V(1)*y_bar(1)+V(2)*y_bar(2)+V(3)*y_bar(3))/(V(1)+
    V(2)+V(3)) //in
z=(V(1)*z_bar(1)+V(2)*z_bar(2)+V(3)*z_bar(3))/(V(1)+
    V(2)+V(3)) //in
//Result
clc
printf('The centroid is at x=%f in,y=%f in and z=%f
    in',x,y,z)
```

Scilab code Exa 10.26 1st Moment and CG

```
//Initilization Of Variables
a=0 //Lower Limit oF the Integral
b=8 //Upper Limit of the Integral
n=10 //Interval of the integral
//Calculation
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
```

```
endfunction
deff('[y]=f(t)','y=75*t^2')
Rr=Trap_Composite1(f,a,b,n)/(2*8) //lb
//Moment calculations
M=Trap_Composite1(f,a,b,n) //ft-lb
//Result
clc
printf('The reaction is %f lb and Moment is %f lb-ft',Rr,M)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 10.27 1st Moment and CG

```
//Initilization Of Variables
a=0 //Lower Limit oF the Integral
b=0.3 //Upper Limit of the Integral
n=10 //Interval of the integral
g=9.8 //m/s^2
rho=1000 //kg/m^3
//Calculation
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=(g*0.6*rho*1.2*t)-(0.6*g*rho*t^2)')
B=Trap_Composite1(f,a,b,n)/(2*b) //N
//Result
clc
printf('The value of B is %f N',B)
```

Scilab code Exa 10.28 1st Moment and CG

```
//Initilization of variables
l=62.4 //lb/ft^3
h=12 //ft
f=105 //lb/ft^3
//Calculations
p1=l*h //lb/ft^2
//Total force on left side
//Simplfying the equation we get a three degree
   equation in d
//solving for d
p=[1/3 0 -144 467]
r=roots(p)
d=r(3) //ft
//Result
clc
printf('The value of d is %f ft',d)
```

Chapter 11

Virtual Work

Scilab code Exa 11.5 V W

```
//Initilization of variables
//Simplification constants
a=90
b=30
c=60
//Calculations
//Allowing for only the cos and sin terms to be zero
    after simplification
theta1=atand(a/(b+2*c)) //degrees
theta2=atand(a/(b+c)) //degrees
theta3=atand(a/b) //degrees
//Result
clc
printf('The values of theta1,theta2 and theta3 are
   %f,%f and %f respectively in Degrees',theta1,
   theta2,theta3)
```

Scilab code Exa 11.6 V W

```
//Initilization of variables
N=100 //lb
mu=0.3 //coefficient of friction
l=5 //in compressed to length
//Calculations
//Simplfying the calculations we obtain
M=8*(N+N*mu) //lb-in
//Result
clc
printf('The Moment is %i lb-in',M)
```

Scilab code Exa 11.7 V W

```
//Initilization of variables
m=10 //kg
g=9.8 //m/s^2
F=200 //N
l=3 //m
//Calculations
//Applying Virtual work principle
By=m*g*0.5 //N
Bx=F*(2/3) //N
//By equations of equilibrium
Ax=-Bx-F //N negative sign indictaes the LEFT
    orientation
Ay=m*g-By //N
//Result
clc
printf('The values are Ax=%fN,Ay=%fN,Bx=%fN and By=
    %fN',Ax,Ay,Bx,By )
```

Scilab code Exa 11.10 V W

```
//Initilization of variables
l=2 //ft
W=20 //lb
k=144 //lb/ft
r=0.5 //ft
theta=44.1 //degrees
//Calculations
//Simplfying the solution to obtain
sinetheta=(k*r^2)/(W*l) //in terms of theta
//By trial and error theta=44.1 degrees
//Check for stable equilibirum
Check=-W*cosd(theta)*l+k*r^2
//Result
clc
printf('the Check Value is %f which indicates Stable
    Equilibirum',Check)
```

Chapter 12

Kinematics of a Particle

Scilab code Exa 12.1 Kin of a Part

```
//Initilization of variables
t=4 //seconds
//Calculations
//Displacement
x=3*t^3+t+2 //ft
//Velocity
v=9*t^2+1 //ft/s
//Acceleration
a=18*t //ft/s^2
//Result
clc
printf('The dipalacemnt is %f ft and the velocity is
    %f ft/s and Acceleration is %f ft/s^2',x,v,a)
```

Scilab code Exa 12.2 Kin of a Part

```
//Initilization of variables
t1=4 //s
```

```
t2=5 //s
//Calculation
v1=9*t1^2+1 //ft/s
v2=9*t2^2+1 //ft/s
a=(v2-v1)/(t2-t1) //m/s^2
//Result
clc
printf('The acceleration during fifth second is %f
   ft/s^2',a)
```

Scilab code Exa 12.3 Kin of a Part

```
//Defining Matrices
t=[0 1 2 3 4 5 10] //s
//Displacement matrix
s=[8*t^2+2*t] //m
//Velocity Matrix
v=[16*t+2] //m/s
//Acceleration Matrix
a=16 //m/s^2
//Plotting the curves
//S-T curve
subplot(221)
plot(t,s)
xlabel('t(s)')
ylabel('s(m)')
subplot(222)
plot(t,v)
xlabel('t(s)')
ylabel('v(m/s)')
subplot(223)
plot(t,a)
xlabel('t(s)')
ylabel('a(m/s^2)')
//Result
```

```
clc
printf('The graphs are the solutions')
```

Scilab code Exa 12.4 Kin of a Part

```
//Initilization of variables
v_o=0 //ft/s
v_f=88//ft/s
t=28 //s
//Calculations
k=(v_f-v_o)/t //ft/s^2
s=((v_f-v_o)/2)*t //ft
//Result
clc
printf('The constant k is %f and displacement is %f
   ft',k,s)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 12.5 Kin of a Part

```
//Initilization of variables
v_o=0 //ft/s
v_f1=30 //ft/s
v_f2=0 //ft/s
t1=3 //s
t2=2 //s
//Calculations
//Plotting the v-t curve
//Velocity matrix
v=[v_o,v_f1,v_f2]
//Time matrix
t=[0,3,5]
plot(t,v)
```

```
xlabel('t')
ylabel('v')
//Part "b"
//Acceleration at 3s
a1=(v_f1-v_o)/t1 //ft/s^2
//Acceleration at 5s
a2=(v_f2-v_f1)/t2 //ft/s^2
//Part "c"
s=(v_f1*t1*0.5)+(v_f1*t2*0.5) //ft
//Part "d"
//Simplfying the equation we get
//7.5t^2-30t+5=0
a=7.5
b=-30
c=5
q=sqrt(b^2-4*a*c)
x1=(-b+q)/(2*a)
x2=(-b-q)/(2*a)
//As x1 is greater than 2 it does not hold as a
    solution
t=x2 //s
//Hence total time is
T=t1+t //s
clc
//Result
printf('The acceleration at 3s and 5s are %f ft/s^2
    and %f  ft/s^2 respectively\n',a1,a2)
printf('The displacement is %f ft\n',s)
printf('The total time is %f s',T)
```

Scilab code Exa 12.6 Kin of a Part

```
//Initilization of variables
v_o=2 //m/s
y_o=120 //m
```

```
g=9.8 //m/s^2
//Calculations
//Solve  using ground as datum
y=0
//Simplfying the equation
a=4.9
b=-2
c=-120
q=sqrt(b^2-4*a*c)
x1=(-b+q)/(2*a) //s
x2=(-b-q)/(2*a) //s
//Result
clc
printf('The time required is %f s',x1)
//As x2 is negative and negative time does not make
   any physical sense
```

Scilab code Exa 12.7 Kin of a Part

```
//Initilization of variables
Vo1=80 //ft/s
Vo2=60 //ft/s
g=32.2 //ft/s^2
//Calculations
//Simplfying by equating the two times
t=(-(Vo2*2)-(g*0.5*4))/(Vo1-Vo2-(g*0.5*4)) //s
//Substituting this t in s we get
s=(Vo1*t)-(0.5*g*t*t) //ft
//Result
clc
printf('The time obtained is %f s and the balls meet
    at %f ft',t,s)
```

Scilab code Exa 12.8 Kin of a Part

```
//Initilization of variables
theta=40 //degrees
x=100 //ft
ay=32.2 //ft/s^2
//Calculations
//Simplfying the equation
t=sqrt((tand(theta)*x)/(ay/2)) //s
//Velocity calculations
Vo=100/(cosd(theta)*t) //ft/s
//Result
clc
printf('The initial speed should be %f ft/s',Vo)
```

Scilab code Exa 12.9 Kin of a Part

```
//Initilization of variables
t=[0,1,2,3,4,5,6] //s
//Solving the Differential Equations we obtain
s=(t+1)^3 //ft
v=3*(t+1)^2 //ft/s
a=6*(t+1) //ft/s^2
//Plotting
subplot(221)
plot(t,s)
xlabel('t(s)')
ylabel('s(ft)')
subplot(222)
plot(t,v)
xlabel('t(s)')
ylabel('v(ft/s)')
subplot(223)
plot(t,a)
xlabel('t(s)')
```

```
ylabel('a(ft/s^2)')
//Result
clc
printf('The result are the gplots that have been
    generated')
```

Scilab code Exa 12.10 Kin of a Part

```
//Initilization of variables
t=3 //s
//Calculations
//After solving the differential equation
s=(1/3)*(t+2)^3 //ft
v=(t+2)^2 //ft/s
a=2*(t+2) //ft/s^2
//Result
clc
printf('The displacement,velocity and acceleration
    at t=3s are %f ft,%f ft/s and %f ft/s^2
    respectively',s,v,a)
```

Scilab code Exa 12.12 Kin of a Part

```
//Initilization of variables
//Calling upward direction positive
xdot1=6 //ft/s
xdot3=3 //ft/s
xdoubledot=2 //ft/s^2
xdoubledot3=-4 //ft/s^2
//Calculations
xdot=-xdot1 //ft/s
xdot2=2*xdot-xdot3 //ft/s
xdoubledot2=2*xdoubledot-xdoubledot3 //ft/s^2
```

```
//Result
clc
printf('The value of velocity is %f ft/s and
   acceleration is %f ft/s^2',xdot2,xdoubledot2)
```

Scilab code Exa 12.14 Kin of a Part

```
//Initilization of variables
x=2 //m/s
//Differentation constant
dc=3.6*2*2*2
//Calculations
//y=3.6*x^2
//Taking the derivative twice of both x and y
   quantities we get
a=dc //m/s^2
//The rest of the solution is theoritical hence not
   coded
//Result
clc
printf('The acceleration is:%f m/s^2',a)
```

Scilab code Exa 12.16 Kin of a Part

```
//Initilization of variables
t=4 //s
//Calculations
//Part (a)
x=t^3 //in
y=-2*t^2 //in
z=2*t //in
//Part (b)
//Theory question
```

```
//Part(c)
//Unit vector calculation
m=sqrt(4^2+1^1+(-3)^2)
e_l=[4/m,1/m,-3/m]
v=[3*t^2,-4*t,2] //in/s
//Projection of v on n at t=4s
dot=v.*e_l //in/s
a=dot(1)+dot(2)+dot(3) //in/s
//Result
clc
printf('The co-ordinates of position are x=%fin,y=
    %fin,z=%fin\n',x,y,z)
printf('The projection of v on n at t=4s is %fin/s',
    a)
```

Scilab code Exa 12.17 Kin of a Part

```
//Initilization of variables
theta=%pi/3 //rad
//Calculations
//Method (a)
t=sqrt(theta) //s
r=2*theta
rdot=4*t
thetadot=2*t
//Velocity calculations
x=r*thetadot
v=sqrt((rdot)^2+x^2) //ft/s
//Theta calculations
thetax=30+atand(rdot/x) //degrees
//Method (b)
x=2*theta*cos(theta) //ft
y=2*theta*sin(theta) //ft
xdot=4*t*((cos(t^2)))+2*t^2*(-sin(t^2))*(2*t) //ft/s
ydot=4*t^2*sin(t^2)+2*t^2*cos(t^2)*2*t //ft/s
```

```
v=sqrt(xdot^2+ydot^2) //ft/s
thetax=atand(ydot/-xdot) //degrees
//Result
clc
printf('By both the methods we obtain v=%f ft/s and
    thetax as %f degrees',v,thetax)
```

Scilab code Exa 12.18 Kin of a Part

```
//Initilization of variables
theta=%pi/3 //rad
//Calculations
t=sqrt(theta) //s
thetadot=2*t
thetadoubledot=2
r=2*t^2
rdot=4*t
rdoubledot=4
ax=rdoubledot-(r*thetadoubledot*thetadoubledot) //ft
    /s^2
ay=2*rdot*thetadot+r*thetadoubledot //ft/s^2
a=sqrt(ax^2+ay^2)
thetax=30+atand(ax/ay) //degrees
//Solving by cartesian co-ordinate system yields
    same solution
//Result
clc
printf('The acceleration is %f ft/s^2 and thetax=%f
    degrees',a,thetax)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 12.20 Kin of a Part

```
//Initilization of variables
theta=45 //degrees
l=0.5 //m
w=10 //rad/s
//Calculations
//PART a
//Here the theta derivative with respect to time is
    angular speed w
Vp1=l*(secd(theta)^2)*w //m/s
//Part b
//Radial Component
r=l*secd(theta)*tand(theta)*w //m/s
//Transverse Component
t=l*secd(theta)*w //m/s
//Total
Vp2=sqrt(r^2+t^2) //m/s
//Result
clc
printf('The velocity is:%fm/s\n',Vp1)
printf('The velocity in part b is %fm/s',Vp2)
```

Scilab code Exa 12.21 Kin of a Part

```
//Initilization of variables
Va=5 //ft/s
theta=70 //degrees
l=6.24 //ft
//Calculations
Vb=-cotd(theta)*Va //ft/s
//Result
clc
printf('The value of Vb is %fft/s',Vb)
```

Scilab code Exa 12.25 Kin of a Part

```
//Initilization of variables
theta=linspace(0,360,13)
//Calculations
//Defining everything in terms of matrices
t=(theta*%pi)/(180*6) //s converting degrees to
    radians
costheta=cosd(theta)
sintheta=sind(theta)
x=2*costheta //ft
v=-12*sintheta //ft/s
a=-72*costheta //ft/s^2
//Plotting
subplot(221)
plot(t,x)
xlabel('t(s)')
ylabel('Displacement x(ft)')
subplot(222)
plot(t,v,t,0)
xlabel('t(s)')
ylabel('Velocity v(ft/s)')
subplot(223)
plot(t,a)
xlabel('t(s)')
ylabel('Acceleration a(ft/s^2)')
//Result
clc
printf('The results are the plots')
```

Scilab code Exa 12.26 Kin of a Part

```
//Initilization of variables
d=1.2 //m
w0=0 //rpm
```

```
w=2000 //rpm
t=20 //s
//Calculations
alpha=(w-w0)/t
alpha_rad=(alpha*2*%pi)/60 //converting to radians/s
   ^2
//Result
clc
printf('The angular acceleration is %frad/s^2',
   alpha_rad)
```

Scilab code Exa 12.27 Kin of a Part

```
//Initilization of variables
w0=0 //rad/s
w=209 //rad/s
t=20 //s
//Calculations
theta=0.5*(w+w0)*t //rad
theta_rev=round(theta/(2*%pi)) //revolutions
   rounding off
//Result
clc
printf('The flywheel makes %i revolutions',theta_rev
   )
```

Scilab code Exa 12.28 Kin of a Part

```
//Initilization of variables
w0=0 //rad/s
alpha=10.5 //rad/s^2
t=0.6 //s
r=0.6 //m
```

```
//Calculations
w=w0+alpha*t //rad/s
v=r*w //m/s
a_t=r*alpha //m/s^2
a_n=r*w*w //m/s^2
a=sqrt(a_t^2+a_n^2) //m/s^2
phi=atand(a_t/a_n) //degrees
//result
clc
printf('The tangential velocity is %fm/s\n',v)
printf('the acceleration is %fm/s^2 and angle is %f
    degrees',a,phi)
```

Scilab code Exa 12.29 Kin of a Part

```
//Initilization of variables
l=4 //ft
wb=40 //rpm
we=60 //rpm
//Calculations
r=l/2 //ft
vb=r*((wb*2*%pi)/60) //ft/s
ve=r*((we*2*%pi)/60)
//Result
clc
printf('The linear speeds are %f ft/s and %f ft/s at
    b and e respectively',vb,ve)
```

Scilab code Exa 12.30 Kin of a Part

```
//Initilization of variables
wb=40 //rpm
we=60 //rpm
```

```
t1=5 //s using different symbol to avoid conflict in
    decleration
t=2 //s
r=2 //ft
//Calculations
alpha=(((we*2*%pi)/60)-((wb*2*%pi)/60))/t1 //rad/s^2
w=((wb*2*%pi)/60)+alpha*t //rad/s
//Components of acceleration are
a_t=r*alpha //ft/s^2
a_n=r*w^2 //ft/s
//result
clc
printf('The tangential and normal acceleration\n are
    %f rad/s^2 and %frad/s^2 respectively',a_t,a_n)
```

Scilab code Exa 12.31 Kin of a Part

```
//Initilization of variables
d=200 //mm
w0=(800*2*%pi)/60 //rpm
w=0 //rpm
t=600 //s
//Calculations
alpha=(w-w0)/t //rad/s^2 (deceleration)
//result
clc
printf('The angular acceleration is %frad/s^2\n The
   negative sign indicates that the wheel
   decelerates',alpha)
```

Scilab code Exa 12.32 Kin of a Part

```
//Initilization of variables
```

```
//The symbols used here differ from the textbook
    solution to avoid conflict
t1=0 //s
t2=0.5 //s
t3=2.5 //s
t4=1/3 //s
w=200 //rpm
w0=0 //rpm
//Calculations
theta1=0.5*(w0+w/60)*t2 //rev
theta2=(w/60)*(t3-t2) //rev
theta3=0.5*(w/60+w0)*t4 //rev here the values of w
    and w0 are interchanged but essentially the value
     comes out to be the same hence the decleration
    has not been changed
theta=theta1+theta2+theta3 //rev
//Result
clc
printf('The wheel undergoes %f of rotations',theta)
```

Scilab code Exa 12.34 Kin of a Part

```
//Initilization of variables
t=1 //s
r=4 //m
//Calculations
s=t^3+3 //m
theta=s/r //rad
dtheta_dt=0.75*t^2 //rad/s
Vx=-4*sin(theta)*dtheta_dt //m/s
Vy=4*cos(theta)*dtheta_dt //m/s
V=sqrt(Vx^2+Vy^2) //m/s
//Result
clc
printf('The components of  velocity are Vx=%fm/s  ,Vy
```

```
=%fm/s and V=%fm/s',Vx,Vy,V)
```

Scilab code Exa 12.35 Kin of a Part

```
//Initilization of variables
t=1 //s
theta=1 //rad
//Calculations
dtheta_dt=0.75*t^2 //rad/s
acc=1.5*t //rad/s^2
ax=-4*cos(theta)*dtheta_dt^2-(4*sin(theta)*acc) //m/
   s^2 (to left)
ay=-4*sin(theta)*dtheta_dt^2+(4*cos(theta)*acc) //m/
   s^2 (up)
a=sqrt(ax^2+ay^2) //m/s^2
//result
clc
printf('The acceleration is %fm/s^2',a)
```

Scilab code Exa 12.36 Kin of a Part

```
//Initilization of variables
t=2 //s
//Calculations
//Velocity
vx=8*t-3 //ft/s
vy=3*t^2 //ft/s
v=sqrt(vx^2+vy^2) //ft/s
theta_x=atand(vy/vx) //degrees
//Acceleration
ax=8 //ft/s^2
ay=6*t //ft/s^2
a=sqrt(ax^2+ay^2) //ft/s^2
```

```
phi_x=atand(ay/ax) //degrees
//Result
clc
printf('The velocity is %fft/s and angle is
   %fdegrees\n The acceleration is %fft/s^2 and
   angle it makes is %fdegrees',v,theta_x,a,phi_x)
```

Scilab code Exa 12.37 Kin of a Part

```
//Initilization of variables
V_ao=29.3 //ft/s
OA=50 //ft
theta=45 //degrees
OB=50*sqrt(2) //ft
//Calculations
w_ao=V_ao/OA //rad/s
V_bo=V_ao*cosd(theta) //ft/s
w_bo=V_bo/OB //rad/s
//Result
clc
printf('The angular velocity with respect to the
   observer is %frad/s\n The angular velocity after
   moving 50ft is %frad/s',w_ao,w_bo)
```

Scilab code Exa 12.38 Kin of a Part

```
//Initiliztaion of variables
theta=30 //degrees
r=[100*tand(theta),100] //ft
v=17.6 //ft/s
//Calculations
v_1=100*secd(theta)*secd(theta)
w=v/v_1 //rad/s (clockwise)
```

```
//result
clc
printf('The angular velocity is %frad/s',w)
```

Scilab code Exa 12.39 Kin of a Part

```
//Initilization of variables
t=2 //s
//Calculations
Vx=20*t+5 //m/s
Vy=t^2-20 //m/s
//As indefinite integral is not possible
x=10*t^2+5*t+5 //m
y=0.5*t^2-20*t-15 //m
ax=20 //m/s^2
ay=2*t //m/s^2
//Result
clc
printf('The displacement components are x=%fm,y=%fm\n The velocity components are Vx=%fm/s,Vy=%fm/s\n The acceleration components are ax=%fm/s^2 and ay=%fm/s^2',x,y,Vx,Vy,ax,ay)
```

Scilab code Exa 12.40 Kin of a Part

```
//Initilization of variables
d=0.1 //m
v=20 //m/s
a_g=6 //m/s^2
d2=0.150 //m
//Calculations
r=d/2 //m
w=v/r //rad/s
```

```
vb=d2*0.5*w //m/s
alpha=a_g/r //rad/s^2
a_t=d2*0.5*alpha //rad/s^2 tangential acceleration
a_n=d2*0.5*w*w //m/s^2 normal acceleration
a=sqrt(a_t^2+a_n^2) //m/s^2 linear acceleration
//Result
clc
printf('The linear velocity is %fm/s and the
   acceleration is %fm/s^2',vb,a)
```

Scilab code Exa 12.41 Kin of a Part

```
//Initilization of variables
theta=40 //degrees
x=100 //ft
ax=0 //ft/s^2
ay=-32.2 //ft/s^2
//Calculations
//vox=vocos40....(1)
//voy=vox*t-1/2(32.2)t^2...(2)
//Simplyfying eq (1) and eq(2)
t_f=sqrt((x*tand(theta))/(0.5*(-ay))) //s time of
   flight
Vo=x/(cosd(theta)*t_f) //ft/s
//As the max height occurs at half wat through the
   flight
t=t_f/2 //s
ymax=Vo*sind(theta)*t+(0.5*ay*t*t) //ft the formula
   has positive sign as ay is defined negative
//result
clc
printf('The max height the ball will reach is %f ft'
   ,ymax)
```

Chapter 13

Dynamics of a Particle

Scilab code Exa 13.1 Dyna of a Part

```
//Initilization of variables
W=2 //lb
F=1.5 //lb
g=32.2 //ft/s^2
//Angles are with respect to the plane
theta1=10 //degrees
theta2=30 //degrees
//Calculations
//Now here the forces are considered as parallel and
    perpendicular to the plane
//Applying Newtond Principle
ax=(g/2)*(F*cosd(theta1)-(W*sind(theta2))) //ft/s^2
N1=(2*cosd(theta2)-(F*sind(theta1))) //lb
//result
clc
printf('The force on the particle is %flb\n The
   acceleration is %fft/s^2',N1,ax)
```

Scilab code Exa 13.2 Dyna of a Part

```
//Initilization of variables
m=5 //kg
s=12 //m
v=4 //m/s
vo=0 //m/s
g=9.8 //m/s^2
mu=0.25
//Calculations
//Using the kinematic equations of motion
a=(v^2-vo^2)/(2*s) //m/s^2
//Using Newtons Principle
N1=g*m //N
P=m*a+mu*N1 //N
//Result
clc
printf('The value of P is %fN',P)
```

Scilab code Exa 13.3 Dyna of a Part

```
//Initilization of variables
m=2 //kg
vo=0 //m/s
v=3 //m/s
s=0.8 //m
theta=20 //degrees
g=9.8 //m/s^2
//Calculations
N=m*g*cosd(theta) //N
a=(vo^2-v^2)/(2*s) //m/s^2
u=-((2*a)+(m*g*sind(theta)))/N
//Solving for return speed
//Symbol convention is different from textbook
a_ret=((m*g*sind(theta))-(u*N))/2 //m/s^2
vf=sqrt((2*a_ret*s)) //m/s
//Result
```

```
clc
printf('The speed is %fm/s',vf)
```

Scilab code Exa 13.4 Dyna of a Part

```
//Initilization of variables
W=1800 //lb
r=2000 //ft
v=58.7 //ft/s
g=32.2 //ft/s^2
//Calculations
F=(W*v*v)/(g*r) //lb
//Result
clc
printf('The frictional force to be exerted is %f lb'
    ,F)
```

Scilab code Exa 13.7 Dyna of a Part

```
//Initilization of variables
W=10 //lb
theta=30 //degrees
l=2 //ft
w=10 //rev/min
g=32.2 //ft/s^2
//Calculations
r=l*cosd(theta) //ft
a_n=r*(((w*2*%pi)/60)^2) //ft/s^2
//Applying Newtons Principle
//Solving by matrix method
A=[cosd(theta),-sind(theta);sind(theta),cosd(theta)]
B=[(W*a_n)/g;W]
C=inv(A)*B //lb
```

```
//Result
clc
printf('The value of T is %flb',C(1))
```

Scilab code Exa 13.8 Dyna of a Part

```
//Initilization of variables
m=4 //lb
v=6 //ft/s
r=2 //ft
theta1=40 //degrees
theta2=20 //degrees
g=32.2 //ft/s^2
//Calculations
a_n=v^2/r //ft/s^2
//Applying Newtons Principle
Fi=(m*a_n)/g //lb
//Solving by matrix method
A=[cosd(theta1),cosd(theta2);sind(theta1),-sind(
    theta2)]
B=[m;Fi]
C=inv(A)*B //lb
//Result
clc
printf('The value of T and C are %flb and %flb
    respectively',C(1),C(2))
```

Scilab code Exa 13.10 Dyna of a Part

```
//Initilization of variables
m1=2 //kg
theta=20 //degrees
m2=4 //kg
```

```
t=4 //s
g=9.8 //m/s^2
vo=0 //m/s
//Calculations
//Applying Newtons Principle
//Solving by matrix method
A=[1,-2;1,4]
B=[m1*g*sind(theta);m2*g]
C=inv(A)*B
a=C(2) //m/s^2
v=vo+a*t //m/s
//Result
clc
printf('The velocity of 4kg mass is %fm/s',v)
```

Scilab code Exa 13.11 Dyna of a Part

```
//Initilization of variables
m_A=20 //lb
m_B=60 //lb
u=0.3 //coefficient of friction
t=4 //s
theta1=30 //degrees
theta2=60 //degrees
g=32.2 //ft/s^2
vo=0 //ft/s
//Calculations
N1=m_A*cosd(theta1) //lb
N2=m_B*cosd(theta2) //lb
//Solving for T and a using matrix method
A=[1,-m_A/g;-1,-m_B/g]
B=[(m_A*sind(theta1)+u*N1);(-m_B*sind(theta2)+u*N2)]
C=inv(A)*B
a=C(2) //ft/s^2
v=vo+a*t //ft/s
```

```
//Result
clc
printf('The velocity is %fft/s',v)
```

Scilab code Exa 13.12 Dyna of a Part

```
//Initilization of variables
m_A=40 //kg
m_B=15 //kg
F=500 //N
g=9.8 //m/s^2
theta=30 //degrees
//Calculations
m=m_A+m_B //kg
a=(F-m*g*sind(theta))/(m) //m/s^2
//Summing forces parallel and perpendicular to the
    plane
//Simplfying equation (1) and (2)
Nb=m_B*g+(m_B*a*sind(theta)) //N
//Substituting this in eq(1)
u=-(m_B*g*cosd(theta)-(Nb*cosd(theta)))/(Nb*sind(
    theta))
//Result
clc
printf('The value of u is %f',u)
```

Scilab code Exa 13.13 Dyna of a Part

```
//Initilization of variables
P=70 //N
m_A=16 //kg
u_AH=0.25 //coefficient of friction between Block A
    and Horizontal Plane
```

```
m_B=4 //kg
u_BH=0.5 //coefficient of friction between Block B
    and Horizontal Plane
theta=10 //degrees
g=9.8 //m/s^2
//Calculations
//Applying sum of forces to both the FBD's
//Solving by matrix method
A=[-cosd(theta),-u_AH,-m_A,0;-sind(theta),1,0,0;cosd
    (theta),0,-m_B,-u_BH;sind(theta),0,0,1]
B=[-P;m_A*g;0;m_B*g]
C=inv(A)*B //Ans
//Result
clc
printf('The Value of T is %fN',C(1) )
```

Scilab code Exa 13.14 Dyna of a Part

```
//Initilization of variables
theta=10 //degrees
v=10 //ft/s
v0=0 //ft/s
u=1/3 //coefficient of friction
g=32.2 //ft/s^2
//Calculations
//Equations of motion for box are
//Simplfying the equations by sybstitution
a=((u*cosd(theta))-sind(theta))*g //ft/s^2
//Time calculations
t=(v-v0)/a //s
//Result
clc
printf('The value of a is %fft/s^2\n The time
    required is %f seconds',a,t)
```

Scilab code Exa 13.15 Dyna of a Part

```
//Initilization of variables
g=9.8 //m/s^2
//Calculations
//Simplfying the equations we can solve for T2 and
    aA first to obtain the solution
//Solving by matrix method
A=[-1.5,-4;-3.5,24]
B=[-4*g;-24*g]
C=inv(A)*B
T2=C(1) //N
T1=T2/2 //N
T3=T2/2 //N
//Acceleration calculations
a1=1*g-T1 //m/s^2
a2=(2*g-T1)/2 //m/s^2
a3=(3*g-T3)/3 //m/s^2
a4=(4*g-T3)/4 //m/s^2
//Tension in fixed cord
T_f=2*T2 //N
//Result
clc
printf('The acceleration values are a1=%f,a2=%f,a3=
    %f and a4=%f m/s^2\n The tension in the fixed
    cord is %fN',a1,a2,a3,a4,T_f)
```

Scilab code Exa 13.16 Dyna of a Part

```
//Initilization of variables
m1=14 //kg
m2=7 //kg
```

```
theta=45 //degrees
u_1=1/4 //coefficient of friction between mass 1 and
    plane
u_2=3/8 //coefficient of friction between mass 2 and
    plane
g=9.8 //m/s^2
//Calculations
//The equations of motion for m1 are
N1=m1*g*cosd(theta) //N
F1=u_1*N1 //N
//The equations of motion for m2 are
N2=m2*g*cosd(theta) //N
F2=u_2*N2 //N
//Now to get T and a we solve using matrix method
A=[-1,-m1;1,-m2]
B=[-(m1*g*sind(theta)-F1);-(m2*g*sind(theta)-F2)]
C=inv(A)*B
//Result
clc
printf('The Value of T is %fN',C(1))
```

Scilab code Exa 13.19 Dyna of a Part

```
//Initilization of variables
W=12 //oz
k=2 //oz/in
M=0.34 //kg
K=22 //N/m
g=32.2 //ft/s^2
//Calculations
//Part(a)
a=(k*W*g)/16
b=W/16
f=(1/(2*%pi))*(sqrt(a/b)) //Hz for simplicity the
   numerator and denominator have been computed
```

```
seperately as a and b
//Part(b)
F=(1/(2*%pi))*(sqrt(K/M)) //Hz
//Result
clc
printf('The frequency in part (a) is %f Hz and in
    part(b) is %f Hz',f,F)
```

Scilab code Exa 13.20 Dyna of a Part

```
//As the entire question is theoritical
//theta is directly computed
theta=acosd(2/3) //degrees
//result
clc
printf('The value of theta is %f degrees',theta)
```

Scilab code Exa 13.28 Dyna of a Part

```
//Initilization of variables
G=6.658*10^-8 //cm^3/g.s^2
//Calculations
G1=G*((3.281*10^2)/((2.205/32.2)*10^4)) //ft^3/slug-
    s^2
G2=G1 //ft^4/lb-s^4
//Result
G1
G2
```

Scilab code Exa 13.29 Dyna of a Part

```
//Initilization of variables
//Modifying the value of C without vo^2 in it
C=5000*5280
G=3.43*10^-8 //Gravatational Constant
M=4.09*10^23 //Mass of the Earth
a=5.31*10^8
//When the orbit is circular e=0
vo1=sqrt(a) //ft/s
//When the orbit is parabolic e=1
vo2=sqrt((C*a+G*M)/C) //ft/s
//Result
clc
printf('The value of vo1=%f is smaller than vo2=%f,
    hence the\n Satellite will enter a hyperbolic
    path and never return',vo1,vo2)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 13.30 Dyna of a Part

```
//Initilization of variables
r=3940+500 //mi
phi=0 //degrees
vo=36000 //ft/s
C=4440*5280*vo
G=3.43*10^-8
M=4.09*10^23 //kg
//Calculations
e=((C*vo)/(G*M))-1
//Result
clc
printf('The value of e=%f hence the path is
    Hyperbolic',e)
```

Scilab code Exa 13.31 Dyna of a Part

```
//Initilization of variables
a=92.9*10^6 //mi
G=3.43*10^-8
T=365*24*3600 //s
c=5280
//Calculations
M=(4*%pi^2*a^3*c^3)/(G*T^2) //slugs
//Result
clc
printf('The mass of the sun is %f slugs',M)
```

Chapter 14

Kinematics of a Rigid Body in Plane Motion

Scilab code Exa 14.2 Kin of rig body in PM

```
//Initilization of variables
d=500 //mm
wo=0 //rpm
w=300 //rpm
t=20 //s
t1=2 //s
//Calculations
alpha=(2*%pi*(1/60)*(w-wo))/t //rad/s^2
w1=wo+alpha*t1 //rad/s
v=(d/(2*1000))*w1 //m/s
a_n=(d/(2*1000))*w1^2 //m/s^2
a_t=(d/(2*1000))*alpha //m/s^2
a=sqrt(a_n^2+a_t^2) //m/s^2
theta=acosd(a_n/a) //degrees
//Result
clc
printf('The computed values are\n alpha=%frad/s^2,w1
   =%frad/s,v=%frad/s\n a=%fm/s^2 and the angle made
    is %fdegrees',alpha,w1,v,a,theta)
```

Scilab code Exa 14.3 Kin of rig body in PM

```
//Initilization of variables
s_BC=2 //m
s_C=2.5 //m
//Calculations
s_B=sqrt(s_BC^2+s_C^2) //m
theta=atand(s_BC/s_C) //degrees
//Result
clc
printf('The absolute displacement is %fm and the
   angle made by the vector is %fdegrees',s_B,theta)
```

Scilab code Exa 14.4 Kin of rig body in PM

```
//Initilization of variables
V_A=20 //mi/h
V_B=70 //mi/h
theta1=60 //degrees
phi=45 //degrees
//Result
//Vector's in matrix form
v_A=[-V_A*cosd(phi),V_A*sind(phi)] //mi/h
v_B=[V_B*cosd(theta1),V_B*sind(theta1)] //mi/h
a=v_A(1)+v_B(1) //mi/h
b=v_A(2)+v_B(2) //mi/h
v_ab=sqrt(a^2+b^2) //mi/h
theta=atand(b/a) //degrees
//The relative velocity v_ba is just different in
   sign while the magnitude stays the same
//Result
```

```
clc
printf('The relative velocity is %fmi/h making an
    angle %fdegres',v_ab,theta)
```

Scilab code Exa 14.9 Kin of rig body in PM

```
//Initilization of variables
l=2.5 //m
v_A=4 //m/s
a_A=5 //m/s^2
theta=30 //degrees
//Calculations
//Vector triangle yields v_a.b=2.93 m/s
v_ab=2.93 //m/s
w=v_ab/l //rad/s (clockwise)
//Ploygon yields alpha_a/b=2.75 m/s^2
alpha_ab=2.75 //m/s^2
alpha=alpha_ab/l //rad/s^2 (counterclockwise)
//Result
clc
printf('The value of angular velocity is %frad/s and
     that of angular acceleration is %frad/s^2',w,
    alpha)
```

Scilab code Exa 14.10 Kin of rig body in PM

```
//Initilization of variables
w=(2*%pi*120)/60 //rad/s
l=24 //in
l_c=4 //in
th=30 //degrees
//Calculations
v=(l_c/12)*w //ft/s
```

```
betaa=asind((l_c*sind(th))/l) //degrees
theta=60-betaa //degrees
//Component of velocity along connecting rod is
v1=v*cosd(theta) //ft/s
v_p=v1/cosd(betaa) //ft/s
//Result
clc
printf('The absoulte velocity is %fft/s',v_p)
```

Scilab code Exa 14.13 Kin of rig body in PM

```
//Initilization of variables
v_pc=3.68 //ft/s
l=2 //ft
//Calculations
w=v_pc/l //rad/s counterclockwise
//Result
clc
printf('The angular velocity is %frad/s',w)
```

Scilab code Exa 14.14 Kin of rig body in PM

```
//This problem is a combination of numerical and
    graphical solution
//The program only deals with the numerical solution
    parts the rest can be verified by graphical
    solution
//Initilization of variables
r=4/12 //ft
w=4*%pi //rad/s
l=2 //ft
w2=1.84 //rad/s
//Calculstions
```

```
ac_n=r*w^2 //ft/s^2
a_pc_n=l*w2^2 //ft/s^2
//Result
clc
printf('The  value of ac_n is %fft/s^2 and that of
   a_pc_n is %fft/s^2',ac_n,a_pc_n)
```

Scilab code Exa 14.15 Kin of rig body in PM

```
//Initilization of variables
w_bc=10 //rad/s
AB=250 //mm
BC=150 //mm
AC=179 //mm
AD=200 //mm
theta1=45 //degrees
//Calculations
v_c=(BC/1000)*w_bc //m/s
AC=sqrt((AB^2+BC^2)-(2*AB*BC*cosd(theta1))) //m
betaa=asind((BC*sind(theta1))/AC) //degrees
gammaa=asind((AB*sind(theta1))/AC)//degrees answer
   in the textbook is incorrect
ang=60-betaa //degrees
CD=sqrt(AD^2+AC^2-(2*AD*AC*cosd(ang))) //mm
D=asind((AC*sind(ang))/CD) //degrees
theta=asind((AD*sind(D))/AC) //degrees
n=360-(theta+gammaa+90) //degrees
v_cd=v_c*cosd(n) //m/s
del=180-(90+D) //degrees
v_D=v_cd/cosd(del) //m/s
w_AD=v_D/(AD/1000) //rad/s
//Result
clc
printf('The angular Velocity of AD is %frad/s',w_AD
   ) //Negative sign indicates clockwise orientation
```

```
//Answer in the textbook is incorrect
```

Scilab code Exa 14.18 Kin of rig body in PM

```
//Initilization of variables
theta1=73.9 //degrees
V=900 //mm/s
theta2=60 //degrees
theta3=46.1 //degrees
//Calculations
BC=sqrt((350*350)+(86.6*86.6)) //mm
CD=400 //mm
v_cb=(V*sind(theta2))/(sind(theta1)) //mm/s
v_c=((V*sind(theta3)))/(sind(theta1)) //mm/s
w_dc=v_c/CD //rad/s
w_cb=v_cb/BC //rad/s
//Result
clc
printf('The angular velocities are w_dc=%frad/s,w_bc
    =%frad/s',w_dc,w_cb)
```

Scilab code Exa 14.19 Kin of rig body in PM

```
//Calculations
//After equating the i and j terms we obtain
    simplified equations
//Solving by matrix method
A=[346,86.7;200,-350]
B=[-3700;-1790]
C=inv(A)*B
//Result
clc
```

```
printf('The angular accelerations are alpha_DC=%frad
   /s^2 and alpha_BC=%frad/s^2',C(1),C(2))
//The signs only indicate that the originally
   assumed orientations are incorrect and are
   opposite to those assumed
```

Scilab code Exa 14.20 Kin of rig body in PM

```
//Initilization of variables
d=3 //m
w=8 //rad/s (clockwise)
alpha=4 //rad/s^2 (counterclockwise)
r=d/2 //m
//Calculations
vo=r*w //m/s
ao=r*alpha //m/s^2
//Here OB is r
OB=r //m
v_bo=OB*w //m/s
v_B=v_bo+vo //m/s
//Also
a_bo=r*alpha //m/s^2 (directed left)
a_bo_n=r*w^2 //m/s^2
a_h=ao+a_bo //m/s^2
a_v=a_bo_n //m/s^2
a_B=sqrt((a_h^2)+(a_v^2)) //m/s^2
phi=atand(a_h/a_v) //degrees
//Result
clc
printf('The linear velocity at B is %fm/s and the
   acceleration is %fn/s^2 making an angle of %f
   degrees with horizontal',v_B,a_B,phi)
```

Scilab code Exa 14.21 Kin of rig body in PM

```
//Initilization of variables
OA=0.6 //m
w=8 //rad/s
theta=30 //degrees
v_O=12 //m/s
alpha=4 //rad/s^2
a_O=6 //m/s^2
//Calculations
//Velocity Calculations
v_AO=OA*w //m/s
v_Ah=v_AO*sind(theta)+v_O //m/s horizontal component
v_Av=v_AO*cosd(theta) //m/s
v_A=sqrt((v_Ah^2)+(v_Av^2)) //m/s
phi=atand(v_Av/v_Ah) //degrees
//Acceleration Calculations
a_AOt=OA*alpha //m/s^2
a_AOn=OA*w^2 //m/s^2
a_Ah=-a_O-a_AOn*cosd(theta)-a_AOt*sind(theta) //m/s
   ^2
a_Av=-a_AOn*sind(theta)+a_AOt*cosd(theta) //m/s^2
a_A=sqrt((a_Ah^2)+(a_Av^2)) //m/s^2
phi2=atand(a_Av/a_Ah) //degrees
//Result
clc
printf('The velocity is %fm/s making an angle %f
   degrees with horizontal\n The acceleration is %fm
   /s^2 making an angle %fdegrees with horizontal',
   v_A,phi,a_A,phi2)
```

Scilab code Exa 14.22 Kin of rig body in PM

```
//Initilization of variables
AL=5 //ft
```

```
d=10 //ft displacement
//Calculations
theta=d/AL //radians
s_o=3*theta//ft
//Result
clc
printf('The displacement  So is %i ft',s_o)
```

Scilab code Exa 14.23 Kin of rig body in PM

```
//Initilization of variables
//Speed and acceleration at the center
v=12 //in/s
a=18 //in/s^2
//Calculations
v_D=((a+v*0.5)/a)*v //in/s
//Speed at point F
v_F=((v/2)/v)*v_D //in/s
//Acceleration at D
a_D=(24/a)*a //in/s^2
//Acceleration at F
a_F=((v/2)/v)*24 //in/s^2
//Result
clc
printf('The velocity and acceleration of weight A
    are %iin/s and %iin/s^2 respectively',v_F,a_F)
```

Scilab code Exa 14.24 Kin of rig body in PM

```
//Calculations
//Speed and acceleration of D
sD=((18-6)/18)*12 //in/s
aD=(12/18)*18 //in/s^2
```

```
//Speed and acceleration of F
sF=(6/12)*8 //in/s
aF=(6/12)*12 //in/s^2
//Result
clc
printf('The velocity and acceleration of weight A
  are %iin/s and %iin/s^2 respectively',sF,aF)
```

Scilab code Exa 14.26 Kin of rig body in PM

```
//Initilization of variables
v_BG=300 //mm/s
v_G=300 //mm/s
a_BGt=500 //mm/s^2
a_BGn=3600//mm/s^2
a_Gh=500 //mm/s^2
a_Bv=1800 //mm/s^2
//Calculations
w=((75-25)/25)*6 //rad/s
alpha=((75-25)/25)*10 //rad/s^2
v_B=sqrt(v_BG^2+v_G^2) //mm/s
a_v=a_Bv-a_BGt //mm/s^2
a_h=a_BGn-a_Gh //mm/s^2
a_B=sqrt(a_v^2+a_h^2) //mm/s^2
//Result
clc
printf('The velocity and acceleration of point B are
   %imm/s and %imm/s^2 respectively',v_B,a_B)
```

Chapter 15

Moment of Inertia

Scilab code Exa 15.11 MI

```
//Initilization of variables
y1=1 //in
y2=4 //in
d1=2.2-1 //in
d2=4-2.2 //in
A1=12 //in^2
A2=8 //in^2
b1=6 //in
b2=2 //in
h1=2 //in
h2=4 //in
//Calculations
y_bar=(A1*y1+A2*y2)/(A1+A2) //in
I1=(1/12)*(b1)*(h1^3) //in^4
I2=(1/12)*(b2)*(h2^3) //in^4
//Using Parallel Axes Theorem
I=(I1+A1*d1^2)+(I2+A2*d2^2) //in^4
//Result
clc
printf('The moment of inertia is %f in^4',I)
```

Scilab code Exa 15.12 MI

```
//Initilization of variables
d=60 //mm diameter of the hole
//Areas
At=100*100 //mm^2
Ab=200*100 //mm^2
Ac=((%pi/4)*d^2) //mm^2
bt=100 //mm
ht=100 //mm
bb=200 //mm
hb=100 //mm
//Distance of centroids of each area
yt=150 //mm
yb=50 //mm
yc=150 //mm
//Calculations
y_bar=((At*yt)+(Ab*yb)-(Ac*yc))/(At+Ab-Ac) //mm
//Distances
dt=yt-y_bar //mm
db=y_bar-yb //mm
dc=yc-y_bar //mm
//Values of Inertia
It=(1/12)*(bt)*(ht^3) //mm^4
Ib=(1/12)*(bb)*(hb^3) //mm^4
Ic=(1/4)*(%pi)*((d/2)^4) //mm^4
//Moment of inertia
I=(It+At*dt^2)+(Ib+Ab*db^2)-(Ic+Ac*dc^2) //mm^4
//Result
clc
printf('The moment of inertia is %f mm^4',I)
```

Scilab code Exa 15.14 MI

```
//Initilization of variables
b1=2 //in
b2=4 //in
h1=8 //in
h2=2 //in
bo=8 //in
ho=8 //in
bi=4 //in
hi=4 //in
//Calculations
I1=(1/12)*(b1)*(h1^3) //in^4
I2=(1/12)*(b2)*(h2^3) //in^4
I=2*(I1+I2) //in^4
Io=(1/12)*(bo)*(ho^3) //in^4
Ii=(1/12)*(bi)*(hi^3) //in^4
I_bar=Io-Ii //in^4
//Result
clc
printf('The moment of inertia is %f in^4',I_bar)
```

Scilab code Exa 15.15 MI

```
//Initilization of variables
b1=75 //mm
b2=12 //mm
h1=12 //mm
h2=162 //mm
d1=75 //mm
//Calculations
A=(h2*b2)+(2*b1*h1) //mm^2
I1=(1/12)*(b1)*(h1^3)+(b1*h1*d1^2) //mm^4
I2=(1/12)*(b2)*(h2^3) //mm^4
I_bar=2*I1+I2 //mm^4
```

```
k=sqrt(I_bar/A) //mm
//Result
clc
printf('The radius of gyration is %f mm',k)
```

Scilab code Exa 15.20 MI

```
//Initilization of variables
r=50 //mm
//Calculations
Ixy=(1/8)*(50^4) //mm^4
//Result
clc
printf('The moment of inertia is %f mm^4',Ixy)
```

Scilab code Exa 15.24 MI

```
//The notation has been changed for ease
//Calculations
x=(5*1*3.5+8*1*0.5)/(5*1+8*1) //in
y=(5*1*0.5+8*1*4)/13 //in
//Moment of inertia
Ix=(1/12)*(5)*(1^3)+(5*2.15*2.15)+(1/12)*(1*8^3)
    +(8*1.35^2) //in^4
Iy=(1/12)*(1)*(5^3)+(5*1.85*1.85)+(1/12)*(8)*(1^3)
    +(8*1.15^2) //in^4
Ixy=(8*1*(-1.15)*1.35)+(5*1*1.85*(-2.15)) //in^4
//Mohr circle calculations
d=0.5*(Ix+Iy) //distance to center of the cirlce
r=sqrt((21^2)+(32.3^2))
maxI=d+r //in^4
theta=atand(32.3/21) //degrees maxI occurs at this
    angle
```

```
minI=d-r //in^r
//Result
clc
printf('The moment of inertias are as follows\n Ix=
   %f in^4,Iy=%f in^4,Ixy=%f in^4\n Imax=%fin^4 and
   Imin=%f in^4',Ix,Iy,Ixy,maxI,minI)
```

Scilab code Exa 15.25 MI

```
//Notations have been changed
//Calculations
x=-(25*125*0.5*125+25*100*0.5*25)/(25*125+25*100) //
   mm
y=(25*125*0.5*25+25*100*75)/5625 //mm
Iy=(1/12)*25*125^3+25*125*(62.5-40.3)^2+(1/12)
   *100*25^3+100*25*(40.3-12.5)^2 //mm^4
Ix=Iy //mm^4 for L-section
//The second computation checks the first
Ixy=(125*25*22.2*27.8)+(100*25*(-27.8)*(-34.7)) //mm
   ^4
//Mohr Circle analysis
Imax=Ix+Ixy //mm^4
Imin=Ix-Ixy //mm^4
//Result
clc
printf('The values of moment of inertia are\n Ix=
   %fmm^4,Iy=%fmm^4,Ixy=%fmm^4\n Imax=%fmm^4 and
   Imin=%fmm^4',Ix,Iy,Ixy,Imax,Imin)
```

Scilab code Exa 15.30 MI

```
//Initilization of variables
rho=490 //lb/ft^3
```

```
t=0.02 //in
d=4 //in
r=d/2 //in
g=32.2 //ft/s^2
//Calculations
W=(%pi*r^2*t*rho)/1728 //lb
//Mass
m=W/g //slugs
//Momemt of inertia
I=(1/4)*m*(r/12)^2 //slug-ft^2
//Result
clc
printf('the moment of inertia is %fslug-ft^2',I)
```

Scilab code Exa 15.36 MI

```
//Initilization of variables
//The integration involves variables hence the
    direct formula is being used in this coding
m=500 //kg
R=0.25 //m
h=0.5 //m
//Calculations
Ix=(3/10)*m*R^2 //kg.m^2
Iy=(3/5)*m*((1/4)*R^2+h^2) //kg.m^2
//Result
clc
printf('Hence proved that Ix=%fkg-m^2 and Iy=%fkg-m
    ^2',Ix,Iy)
```

Scilab code Exa 15.37 MI

```
//Initilization of variables
```

```
del=450 //lb/ft^3
h1=9/12 //ft
h2=10/12 //ft
ro1=4/12 //ft
ri1=2/12 //ft
ro2=18/12 //ft
ri2=16/12 //ft
a=2.5/24 //ft
b=3.5/24 //ft
l=1 //ft
g=32.2 //ft/s^2
//Calculations
Whub=(%pi*ro1^2-%pi*ri1^2)*h1*del //lb
Wrim=(%pi*ro2^2-%pi*ri2^2)*h2*del //lb
//For one spoke
Wspoke=(%pi*a*b*l*del) //lb
//Moment of inertia calculations
Ihub=0.5*(Whub/g)*(ro1^2+ri1^2) //lb-s^2-ft
Irim=0.5*(Wrim/g)*(ro2^2+ri2^2) //lb-s^2-ft
Ispoke=6*((1/12)*(Wspoke/g)*l^2+(Wspoke/g)*h2^2) //
    lb-s^2-ft
Iwheel=Ihub+Irim+Ispoke //lb-s^2-ft
//result
clc
printf('The moment of inertia of the wheel is %flb-s
    ^2-ft',Iwheel)
```

Chapter 16

Dynamics of a Rigid Body in Plane Motion

Scilab code Exa 16.2 Dyna of rig body in PM

```
//Intilization of variables
W=600 //lb
d=30 //in
theta=25 //degrees
g=32.2 //ft/s^2
//Calculations
m=W/g //lb-s^2/ft
//Moment of inertia
I=0.5*m*((d/2)/12)^2 //lb-s^2-ft
//Applying Newtons law and coservation of angular
    momentum and rolling
//Solving by matrix method
A=[1,m,0,0;0,0,0,1;((d/2)/12),0,-I,0;0,1,-((d/2)/12)
    ,0]
B=[W*sind(theta);W*cosd(theta);0;0]
C=inv(A)*B
//Result
clc
printf('The Frictional Force is %f lb and the
```

```
    acceleration is %f ft/s^2',C(1),C(2))
```

Scilab code Exa 16.3 Dyna of rig body in PM

```
//Initilization of variables
m=18 //kg
d=0.6 //m
vo=3 //m/s
theta=20 //degrees
g=9.8 //m/s^2
//Calculations
//Moment of Inertia
I=0.5*m*(d/2)^2 //
//Applying Newtons second Law a
A=[1,m,0,0;0,0,1,0;d/2,0,0,-I;0,1,0,(-d/2)]
B=[g*m*sind(theta);g*m*cosd(theta);0;0]
C=inv(A)*B
//Storing the answers in variables
F=C(1) //N
ax=C(2) //m/s^2
Na=C(3) //N
alpha=C(4) //rad/s^2
//Time Calculations
v=0 //m/s^2
t=(vo)/ax //s
//Result
clc
printf('It takes %f s to reach the highest point of
    travel',t)
```

Scilab code Exa 16.5 Dyna of rig body in PM

```
//Initilization of variables
```

```
m=20 //kg
F1=40 //N
ro=0.6 //m
ri=0.45 //m
g=9.8 //m/s^2
//Calculations
//Moment of inertia
I=(2/5)*m*ro^2 //kg-m^2
//Applying Newtons Law and conservation of angular Momentum
//Solving by matrix method
A=[1,m;ro,-I/ro]
B=[F1;F1*ri]
C=inv(A)*B
//Storing answers in variables
F=C(1) //N
a=C(2) //m/s^2
//Result
clc
printf('The acceleration is %f m/s^2 and F=%f N',a,F)
//The solution in the textbook is incorrect
```

Scilab code Exa 16.6 Dyna of rig body in PM

```
//Initilization of variables
W=16.1 //lb
u=0.10 //co-efficient of friction
g=32.2 //ft/s^2
theta=30 //degrees
F=1.39 //lb
//Calculations
//Applying Newtons Second Law
//Using F=1.39 lb
a=(W*sind(theta)-F)/(W/g) //ft/s^2
```

```
alpha=(F*0.5*5/2)/((W/g)*(0.5^2)) //rad/s^2
//Result
clc
printf('The value of a is %f ft/s^2 and alpha is %f
   rad/s^2.\n Hence the sphere will both roll and
   slip.',a,alpha)
```

Scilab code Exa 16.8 Dyna of rig body in PM

```
//Initilization of variables
theta=30 //degrees
W=80 //lb
Ww=100 //lb
I=4 //slug-ft^2
r=0.5 //ft
v= 20 //ft/s
vo=0 //ft/s
g=32.2 //ft/s^2
//Calculations
//Using Equations of motion
//Solving the system of linear equatinons by matrix
   method
A=[-1,0,-W/g;1,-1,-Ww/g;0,r,-2*I]
B=[-W;Ww*sind(theta);0]
C=inv(A)*B
//Storing values in variables
T=C(1) //lb
F=C(2) //lb
a=C(3) //ft/s^2
//Time calculations
t=(v-vo)/a //s
//Result
clc
printf('The time required is %f s',t)
```

Scilab code Exa 16.9 Dyna of rig body in PM

```
//Initilization of variables
M=70 //kg
ko=0.4 //m
ri=0.45 //m
ro=0.6 //m
theta=30 //degrees
m=35 //kg
g= 9.8 //m/s^2
//Calculations
I=M*ko^2 //kg-m^2
//Using Equations of motion
//Solving the equations by matrix method
A=[-1,-m*0.15,0;1,-M*ro,-1;-ri,-I,ro]
B=[-m*g;M*g*sind(theta);0]
C=inv(A)*B
F=C(3) //N
Na=M*g*cosd(theta) //N
//Required coefficient of friction
u=F/Na //coefficient of friction
//Result
clc
printf('The value of alpha is %f rad/s^2 and Tension
    is %f N\n F=%f N and Na=%f N also u=%f',C(2),C
   (1),F,Na,u)
```

Scilab code Exa 16.10 Dyna of rig body in PM

```
//Initilization of variables
m=200 //kg
g=9.8 //m/s^2
```

```
r=1.2 //m
F1=1000 //N
F2=1400 //N
//Calculations
N=m*g //N
I=(2/5)*(m)*r^2 //kg-m^2
//Using equations of motion
//Solving for F and alpha using matrix method
//Applying equations of motion
A=[1,-m;-r,-I/r]
B=[F1-F2;F1*r]
C=inv(A)*B
//Storing values
F=C(1) //N
alpha=C(2) //rad/s^2
a=r*alpha //m/s^2
//Result
clc
printf('The value of a is %f m/s^2 and F is %f N',a,
    F)
//The negative signs indicate that the direction is
    opposite to what was origninally assumed
```

Scilab code Exa 16.11 Dyna of rig body in PM

```
//Initilization of variables
Wa=161 //lb
Wb=193.2 //lb
Wc=300 //lb
ka=3 //ft
kb=2.5 //ft
theta1=30 //degrees
theta2=45 //degrees
g=32.2 //ft/s^2
//Calculations
```

```
//Moment of inertia Calculations
Ia=(Wa/g)*ka^2 //lb-s^2-ft
Ib=(Wb/g)*kb^2 //lb-s^2-ft
//Using equations of motion for A and B and C
//Solving by matrix method
A=[1,1,-Wa/g,0,0;1,-4,-Ia*(1/4),0,0;-2,0,-Ib*(5/8)
    ,4,0;0,0,-(Wc/g)*(5/2),-1,-0.25;0,0,0,0,1]
B=[Wa*sind(theta1);0;0;-Wc*cosd(theta2);Wc*sind(
    theta2)]
C=inv(A)*B
//Storing values in the variables
T1=C(1) //lb
T2=C(4) //lb
a=C(3) //ft/s^2
//Result
clc
printf('The value of a is %f ft/s^2 and T1=%f lb and
    that of T2 is %f lb',a,T1,T2)
```

Scilab code Exa 16.12 Dyna of rig body in PM

```
//Initilization of variables
W=644 //lb
F=30 //lb
theta=30 //degrees
r=1.5 //ft
g=32.2 //ft/s^2
//Calculations
//Using equations of motion
//Solving by matrix method
A=[1,-W/g;-r,-(1/2)*(W/g)*(2*2)*(1/r)]
B=[W*sind(theta)-F*cosd(theta);-F*2]
C=inv(A)*B
a=C(2) //ft/s^2
//Result
```

```
clc
printf('The value of a is %f ft/s^2',a)
```

Scilab code Exa 16.14 Dyna of rig body in PM

```
//Initilization of variables
W=20 //lb
g=32.2 //ft/s^2
vb=0.5 //rad/s
//Calculations
//Using equations of motion
//Solving the three equations simultaneously by
   matrix method
X=[0,1,-(W/g)*5.2;-1,0,-(W/g)*3;3,-3,-(1/12)*(W/g)
   *12^2]
Y=[-0.75*(W/g);(W/g)*1.3-W;0]
C=inv(X)*Y
A=C(1) //lb
B=C(2) //lb
alpha=C(3) //rad/s^2
//Result
clc
printf('The value of alpha is %f rad/s^2 and of A
   and B are %f lb \nand %f lb respectively',alpha,A
   ,B)
```

Scilab code Exa 16.16 Dyna of rig body in PM

```
//Initilization of variables
mc=7.25 //kg
d=0.9 //m
la=0.2 //m
ma=9 //kg
```

```
F=45 //N
ay=0 //m/s^2
g=9.8 //m/s^2
//Calculations
I=2*(0.5*mc*(d/2)^2)+0.5*ma*(la/2)^2 //kg-m^2
//Using the equations of motion
Na=(2*mc+ma)*g //N
//Simplfying using radial velocity formula
//Solving the two equations using matrix method
A=[-1,-(2*mc+ma);(d/2),-I/(d/2)]
B=[-F;F*(la/2)]
C=inv(A)*B
F=C(1) //N
ax=C(2) //m/s^2
//Result
clc
printf('The computation yields ax=%f m/s^2',ax)
```

Scilab code Exa 16.18 Dyna of rig body in PM

```
//Initilization of variables
r=0.05 //m cylinder radius
g=9.8 //m/s^2
//Calculations
//Here the equation has been solved in terms of the
    veriables
//Hence we directly consider the final result
av=(2*g)/3 //m/s^2
//Result
clc
printf('The value of av is %f m/s^2',av)
```

Scilab code Exa 16.21 Dyna of rig body in PM

```
//initilization of variables
W=16.1 //lb
v=9 //ft/s
phi=30 //degrees
r=0.5 //ft
g=32.2 //ft/s^2
OG=4.5 //ft
//Calculations
//Using equations of motion
an=v^2/OG //ft/s^2
//Solving for alpha we get
N=(W/g)*an+W*cosd(phi) //lb
//Using equations of motion
A=[1,-r;-1,-r*r]
B=[W*sind(phi);0]
C=inv(A)*B
F=C(1) //lb
at=C(2) //ft/s^2
//Result
clc
printf('The value of N and F are %f lb and %f lb
    respectively',N,F)
```

Scilab code Exa 16.24 Dyna of rig body in PM

```
//Initilization of variables
m_abc=20 //kg
m_cd=10 //kg
l_abc=3 //m
l_cd=2 //m
x=0.75 //m
y=1.5 //m
theta=60 //degrees
F=1000 //N
g=9.8 //m/s^2
```

```
//Simplfying constants
a=26
b=28.3
c=49
//Calculations
//After the rigorous simplification we arrive at the
    following
//Bx=26*alpha
//By=49-28.3*alpha
//Summing moments about A
alpha=(m_abc*g*x+F*y+c)/((1/3)*m_abc*l_abc^2+a*tand(
    theta)+b) //rad/s
//Result
clc
printf('The value of angular acceleration is: %frad/
    s^2',alpha)
```

Scilab code Exa 16.25 Dyna of rig body in PM

```
//Initilization of variables
W=50 //lb
P=10 //lb
t=5 //s
vo=0 //ft/s
g=32.2 //ft/s^2
//Calculations
//Using equations of motion
ax=(P*g)/W //ft/s^2
//Solving by matrix method for A and B
F=[1,1;-4,4]
Q=[W;P]
R=inv(F)*Q
//Velocity calculations
v=vo+ax*t //ft/s
A=R(1) //lb
```

```
B=R(2) //lb
//Result
clc
printf('The velocity of the door after 5s is %f ft/s
    and A=%f lb and B=%f lb',v,A,B)
```

Scilab code Exa 16.26 Dyna of rig body in PM

```
//Initilization of variables
AB=2 //m
m=2 //kg
F=20 //N
g=9.8 //m/s^2
//Calculations
//Using equation of motion
a=F/m //m/s^2
//Solving by matrix method for Na and Nb
A=[1,-1;4/5,4/5]
B=[m*g;F*(3/5)]
C=inv(A)*B
//Result
clc
printf('The value of a is %f m/s^2 and the reactions
    are\n Na=%f N and Nb=%f N',a,C(1),C(2))
```

Scilab code Exa 16.27 Dyna of rig body in PM

```
//Initilization of variables
vo=0 //ft/s
//Calculations
s=(0.011*5280*2)/(2*0.004)
//Result
clc
```

```
printf('It travels %f ft before coming to rest',s)
//Answer in the textbook is incorrect by 20ft
```

Scilab code Exa 16.28 Dyna of rig body in PM

```
//Initilization of variables
u=0.3 //coefficient of friction
m=70 //kg
g=9.8 //m/s^2
//Calculations
//CASE 1
//Using equations of motion
Na=m*g //N
ah=(u*Na)/m //m/s^2
//CASE 2
//Applying sum of moments equal to zero
F=(Na*0.3)/1.2 //N
a_h=F/m //m/s^2
//Result
//Intutive insights can be attained after we get these results
clc
printf('The value of Na is %f N and that of acceleration are \ntwo values 1)%f m/s^2 2)%f m/s^2 each for tipping and sliding respectively\n F is %f N',Na,ah,a_h,F)
```

Scilab code Exa 16.29 Dyna of rig body in PM

```
//Initilization of variables
m=60 //kg
me=660 //kg
a=6 //m/s^2
```

```
g=9.8 //m/s^2
//Calculations
//Using equations of motion
P=m*a+m*g //N
//Scale reading
R=P/g //kg
//Increase in mass
I=R-m //kg
//Tension
T=me*a+me*g //N
//Result
clc
printf('The value of P is %f N \n Apparent icrease
    in weight is %f kg\n Tension in the cable is %f N
    .',P,I,T)
//Answer in the textbook is off by 28 //N in Tension
```

Scilab code Exa 16.30 Dyna of rig body in PM

```
//Initilization of variables
u=0.2 //coefficient of friction
ma=1.2 //kg
mb=2 //kg
g=9.8 //m/s^2
//Calculations
Nb=mb*g //N
F=u*Nb //N
//Using equations of motion
//Solving for T and a
A=[-1,-ma;1,-mb]
B=[-ma*g;F]
C=inv(A)*B
T=C(1) //N
a=C(2) //m/s^2
//Taking the sum of the moments
```

```
x_m=-(F*0.15+T*0.15)/Nb //m
x=x_m*1000 //mm
//Result
clc
printf('The acceleration of block A is %f m/s^2 and
   Nb acts at a distance %f mm\n Negative sign
   indictaes that the side assumed is incorrect',a,x
   )
```

Scilab code Exa 16.31 Dyna of rig body in PM

```
//Initilization of variables
a=2.5 //m/s^2
mA=3 //kg
mB=7 //kg
g=9.8 //m/s^2
//Calculations
F=(mA+mB)*a //N
//Using equations of motion
Py=mB*g //N
//Solving for Px and H
A=[1,1;-0.0375,0.0375]
B=[mB*a;Py*0.05]
C=inv(A)*B
Px=C(1) //N
H=C(2) //N
//Result
clc
printf('The value of H is %f N',H)
```

Scilab code Exa 16.32 Dyna of rig body in PM

```
//Initilization of variables
```

```
m=20 //kg
g=9.8 //m/s^2
vo=3 //m/s
v=0 //m/s
s=4 //m
//Calculations
//Using equations of motion
Na=m*g //N
F=(Na*0.075)/0.125 //N
a=F/m //m/s^2
//Displacement
d=-(v^2-vo^2)/(2*a) //m
displ=s-d //m
v_f=sqrt(2*a*displ) //m/s
//Result
clc
printf('The final velocity is %f m/s',v_f)
```

Scilab code Exa 16.33 Dyna of rig body in PM

```
//Initilization of variables
mA=30 //kg
mB=45 //kg
u_ab=1/3 //coefficient of friction between two
    blocks
u_bp=1/10 //coefficient of friction between block
    and horizontal plane
g=9.8 //m/s^2
//Calculations
//By inspection
Na=mA*g //N
Nb=Na+mB*g //N
a=(u_ab*Na-u_bp*Nb)/mB //m/s^2
P=(mA*a+u_ab*Na) //N
//For block A
```

```
//Solving for P,F and a
A=[1,-1,-mA;-0.05,-0.075,0;0,1,-mB]
B=[0;-Na*0.050;Nb*u_bp]
C=inv(A)*B
P_new=C(1) //N
//Result
//As p < p_new
clc
printf('The maximum value of P is %f N',P)
```

Scilab code Exa 16.34 Dyna of rig body in PM

```
//Initilization of variables
Vo=1.5 //m/s
V=0 //m/s
g=9.8 //m/s^2
//Calculations
a=(g*0.2)/0.75 //m/s^2
t=-(V-Vo)/a //s
//Result
clc
printf('The maximum accelerayion is %f m/s^2 and
   minimum time is %f s',a,t)
```

Scilab code Exa 16.36 Dyna of rig body in PM

```
//Initilization of variables
vo=0 //mi/h
v=60 //mi/h
t=13.8 //s
W=3385 //lb
xb=46 //in
xf=66 //in
```

```
xv=31 //in
g=32.2 //ft/s^2
//Calculations
a=(((v*88*60)/3600)-vo)/t //ft/s^2
//Summing horizontal forces
F=(W/g)*a //lb
//Solving for Rf and Rr
A=[1,1;-xf,xb]
B=[W;-F*xv]
C=inv(A)*B
Rr=C(1) //lb
Rf=C(2) //lb
//Result
clc
printf('The value of reactions are Rf=%f lb and Rr=
    %f lb',Rf,Rr)
```

Scilab code Exa 16.43 Dyna of rig body in PM

```
//Initilization of variables
W=161 //lb
F=16.1 //lb
r=18 //ft radius
t=2 //s
g=32.2 //ft/s^2
wo=0 //rad/s
//Calculations
//Using equations of motion
//Solving for T and alpha
A=[r/12,-0.5*(W/g)*(r/12)^2;-1,-F/g]
B=[0;-F]
C=inv(A)*B
alpha=C(2) //rad/s^2
w=wo+alpha*t //rad/s
//Result
```

```
clc
printf('The angular speed is %f rad/s',w)
//The decimal accuray causes the discrepancy
```

Scilab code Exa 16.47 Dyna of rig body in PM

```
//Initilization fo variables
r=2000 //ft
g=32.2 //ft/s^2
d=4.71 //ft
v=176 //ft/s
//Calculations
e=(d*v^2)/(g*r) //ft
//Result
clc
printf('The superelevation is %f ft',e)
//Watch the unit in the final answer
```

Scilab code Exa 16.48 Dyna of rig body in PM

```
//Initilization of variables
a=5 //ft/s^2
C=50 //lb-ft
W=161 //lb
g=32.2 //ft/s^2
//Calculations
T=0.5*(W/g)*1^2*a+C //lb
Ox=-T*(2/sqrt(a)) //lb
Oy=T*(1/sqrt(a))+W //lb
Wa=T/(1-(a/g)) //lb
//Result
clc
```

```
printf('The values are T=%f lb,Wa=%f lb,Ox=%f lb and
    Oy=%f lb',T,Wa,Ox,Oy)
```

Scilab code Exa 16.49 Dyna of rig body in PM

```
//Initilization of variables
m=100 //kg
mr=20 //kg
w=8 //rad/s
l1=300 //mm
l2=600 //mm
g=9.8 //m/s^2
//Calculations
r_bar=(mr*l1+m*750)/120 //mm
I=(1/3)*mr*(l2/1000)^2+(2/5)*m*(l1/2000)^2+m*(0.75)
    ^2 //kg.m^2
alpha=(m+mr)*g*(r_bar/1000)/I //rad/s^2
On=(m+mr)*(r_bar/1000)*w^2 //N
Ot=((m+mr)*(r_bar/1000)*alpha)-(m+mr)*g //N
//Result
clc
printf('The angular acceleration is %f rad/s^2 and
    On=%f N and Ot=%f N',alpha,On,Ot)
//Due to decimal accuracy there is discrepancy in
    answers with the textbook
```

Scilab code Exa 16.50 Dyna of rig body in PM

```
//Initilization of variables
W=40 //lb
w=10 //rad/s
alpha=2 //rad/s^2
r=2 //in
```

```
g=32.2 //ft/s^2
//Calculations
//Using equations of motion
On=(W/g)*(1/6)*w^2 //lb
Ot=(W/g)*(1/6)*alpha
Io=(0.5*(W/g)*0.5^2)*2+((W/g)*(1/6)^2)*2
//Result
clc
printf('The reaction components are On=%f lb and Ot=
    %f lb',On,Ot)
```

Scilab code Exa 16.51 Dyna of rig body in PM

```
//Initilizatin of variables
W=6 //lb
l=8 //ft
v=10 //ft/s
g=32.2 //ft/s^2
theta1=60 //degrees
theta2=30 //degrees
//Calculations
Fe=(W*v^2)/(g*l*0.5) //lb
//Using equations of motion
//Solving for C and T
A=[cosd(theta1),-cosd(theta2);cosd(theta2),cosd(
    theta1)]
B=[-Fe;W]
P=inv(A)*B //lb
C=P(1) //lb
T=P(2) //lb
//Result
clc
printf('The value of C is %f lb and T is %f lb',C,T)
```

Scilab code Exa 16.52 Dyna of rig body in PM

```
//Initilization of variables
W=32.2 //lb
T=120 //lb
m=1 //slug
r=6/12 //ft
//Calculations
w=sqrt((T*(3/5)*4)/(m*r*3)) //rad/s
//Result
clc
printf('The angular speed permissible is %f rad/s',w
   )
```

Scilab code Exa 16.53 Dyna of rig body in PM

```
//Initilization of variables
m=30 //kg
k=0.45 //m
g=9.8 //m/s^2
//Using equations of motion
//Solving for T1,T2 and alpha
A=[1,0,-m;0,-1,-45;-0.6,0.3,-m*k^2]
B=[50*g;-150*g;0]
C=inv(A)*B
//Result
clc
printf('The values are T1=%f N T2=%f N and alpha=%f
   rad/s^2 ',C(1),C(2),C(3))
```

Scilab code Exa 16.54 Dyna of rig body in PM

```
//Initilization of variables
Wc=28 //lb
v=16 //ft/s
Ib=12 //ft-lb-s^2
u=0.4 //coefficient of friction
t=2 //s
g=32.2 //ft/s^2
//Calculations
T=Wc+(Wc/g)*8 //lb
alpha=8/(15/12) //rad/s^2
F=(Ib*alpha+T*1.25)/t //lb
N=F/u //lb
//Summing moments about D
P=(N*8+F*3)/40 //lb
//Summing forces horizontally and vertically
Dx=151-P //lb
Dy=-F //lb
//Result
clc
printf('The reactions at D are Dx=%f lb and Dy=%f lb
    ',Dx,Dy)
```

Scilab code Exa 16.55 Dyna of rig body in PM

```
//Initilization of variables
m=8 //kg
n=90 //rpm
g=9.8 //m/s^2
//Calculations
Fg=m*g //N
w=2*%pi*n/60 //rad/s
//using equations of motion
By=m*g //N
```

```
//Solving for Bx and C
A=[1,1;-0.3,0.9]
B=[m*0.3*w^2;By*0.3]
C=inv(A)*B //N
//Result
clc
printf('The solution is Bx=%f N ,By=%f N and C=%f N'
    ,C(1),By,C(2))
```

Scilab code Exa 16.56 Dyna of rig body in PM

```
//Initilization of variables
m=8 //kg
n=90 //rpm
g=9.8 //m/s^2
r=0.3 //m
//calculations
w=2*%pi*n/60 //rad/s
//Using equations of motion
C=(m*g*0.3+m*r*w^2*r)/1.2 //N
Bx=C-m*r*w^2 //N
By=m*g //N
//Result
clc
printf('The solution is Bx=%f N ,By=%f N and C=%f N'
    ,Bx,By,C)
```

Scilab code Exa 16.57 Dyna of rig body in PM

```
//Initilization of variables
Na=294 //N
Nb=735 //N
//Calculations
```

```
a=(1/10*Nb-1/3*Na)/45 //m/s^2
P=(1/3*Na)-30*a //N
//result
clc
printf('The solution is P=%f N and a=%f m/s^2',P,a)
```

Scilab code Exa 16.58 Dyna of rig body in PM

```
//Initilization of variables
W=50 //lb
g=32.2
//Calculations
//Using equations of motion
a=(10/(W/g)) //ft/s^2
B=((2.5*(W/g)*a)+4*W-1.5*10)/8 //lb
A=50-B //lb
//Result
clc
printf('The solution is A=%f lb B=%f lb and a=%f ft/
    s^2',A,B,a)
```

Scilab code Exa 16.60 Dyna of rig body in PM

```
//Initilization of variables
g=9.8 //m/s^2
r1=0.3 //m
m1=20 //kg
m2=100 //kg
r2=0.75 //m
//Calculations
alpha=(m1*g*r1+m2*g*r2)/(m1*r1^2+(m1/12)*0.6^2+m2*r2
    ^2+(2/5)*m2*0.15^2) //rad/s^2
//Result
```

```
clc
printf('The angular acceleration is %f rad/s^2',
   alpha)
```

Scilab code Exa 16.61 Dyna of rig body in PM

```
//Initilization of variables
r=15/12 //ft
W=600 //lb
theta=25 //degrees
//calculations
ax=(r*W*sind(theta))/((1/r)*14.5+r*18.6) //ft/s^2
F=(W*sind(theta))-18.6*9.09 //lb
//Result
clc
printf('The solution is F=%f lb and ax=%f ft/s^2',F,
   ax)
```

Scilab code Exa 16.62 Dyna of rig body in PM

```
//Initilization of variables
m=7 //kg
g=9.8 //m/s^2
r=0.5 //m
I=0.875 //kg.m^2
//Calculations
//Solving for alpha and T
alpha=(m*g*r)/(I+m*r*0.5) //rad/s^2
T=(I*alpha)/r //N
//Result
clc
printf('The soultion is alpha =%f rad/s^2 and T=%f N
   ',alpha,T)
```

Chapter 17

Work and Energy

Scilab code Exa 17.4 Work

```
//Initilization Of Variables
s1=2 //Lower Limit oF the Integral
s2=5 //Upper Limit of the Integral
n=10 //Interval of the integral
k=20 //lb/in
//Calculation
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,s1,s2,n)
    h=(s2-s1)/n
    s=linspace(s1,s2,n+1)
    I1=(h/2)*((2*sum(f(s)))-f(s(1))-f(s(n+1)))
endfunction
deff('[y]=f(s)','y=k*s')
//Result
clc
printf('The work done is %f in-lb',Trap_Composite1(f
   ,s1,s2,n) )
```

Scilab code Exa 17.6 Work

```
//Initilization of variables
m=5 //kg
d=6 //m
theta1=30 //degrees
theta2=10 //degrees
u=0.2 //coefficient of friction
g=9.8 //m/s^2
F=70 //N
//Calculations
//Using free body diagram
Na=m*g*cosd(theta1)-(F*sind(theta2)) //N
//work done by each force
W=[F*cosd(theta2),-m*g*sind(theta1),0,-u*Na*d] //N.m
//Total Work Done
W_tot=W(1)+W(2)+W(3)+W(4) //N.m
//Using resultant
R=F*cosd(theta2)-(u*Na)-(m*g*sind(theta1)) //N
W_d=R*d //N.m (Work Done)
//Result
clc
printf('The work done is %f N.m',W_d)
```

Scilab code Exa 17.7 Work

```
//Initilization of variables
m=20 //kg
d=1.5 //m
theta=30 //degrees
u=0.25 //coefficient of friction
g=9.8 //m/s^2
F=130 //N
//Calculations
W=F*d-(m*g*sind(theta)*d) //N.m
//Result
clc
```

```
printf('The work done is %i N.m',W)
```

Scilab code Exa 17.9 Work

```
//Initilization of variables
d=6/12 //ft
l=8/12 //ft
l_c=3.2 //in
y=1.82 //in^2
//Calculations
V=(1/4)*%pi*d^2*l //ft^3
//One horizontal inch
h_i=V/l_c //ft^3
//One vertical inch
v_i=100*144 //lb/ft^2
//Then 1.82 in^2 represents
x=y*v_i*h_i //ft-lb
//Result
clc
printf('The work capacity is %f ft-lb',x)
```

Scilab code Exa 17.10 Work

```
//Initilization of variables
speed=90000 //m/h
P=100*1000 //N
//Calculations
Power=P*((speed)/3600) //J/s
//Result
clc
printf('The power developed is %fJ/s',Power)
```

Scilab code Exa 17.11 Work

```
//Initilization of variables
d=0.6 //m
T_t=800 //N
T_s=180 //N
w=200 //rpm
//Calculations
r=d/2 //m radius
//Torque
M=(T_t-T_s)*r //N.m
//Power
w_new=(2*%pi*w)/60 //rad/s
Power=M*(w_new) //W
//Result
clc
printf('The power transmitted is %f W',Power)
```

Scilab code Exa 17.12 Work

```
//Initilization of variables
P=25.6 //lb
w=600 //rpm
a=36 //in
b=12 //in
//Calculations
M=P*(((b/2)+a)/12)  //lb-ft
w_new=(2*%pi*w)/60 //rad/s
Hp=(M*w_new)/550 //hp
//Result
clc
printf('The power being transmitted is %fhp',Hp)
```

Scilab code Exa 17.13 Work

```
//Initilization of variables
Pout=3.8 //bhp
Pin=4.1 //ihp
//Calculations
Efficiency=round((Pout/Pin)*100) //Percent
//Result
clc
printf('The efficiency of the engine is %ipercent',
   Efficiency)
```

Scilab code Exa 17.15 Work

```
//Initilization Of Variables
a=3 //Lower Limit oF the Integral
b=6 //Upper Limit of the Integral
n=10 //Interval of the integral
g=9.8 //m/s^2
w=4/16
//Calculation
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    x=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(x)))-f(x(1))-f(x(n+1)))
endfunction
deff('[y]=f(x)','y=-3*x^-1')
an=-Trap_Composite1(f,a,b,n) //ft-lb
v=sqrt((an*g)/(0.5*w)) //ft/s
//Result
```

```
clc
printf('The speed of the disk is %fft/s',v)
//The answer in the textbook is incorrect
```

Scilab code Exa 17.16 Work

```
//Initilization of variables
l=2 //m
m=4 //kg
w_1=20 //rpm
w_2=50 //rpm
rev=10 //no of revolution
//Calculations
Io=(1/3)*(m)*l^2 //kg.m^2
w1=(2*%pi*w_1)/60 //rad/s
w2=(2*%pi*w_2)/60 //rad/s
theta=2*%pi*rev //rad
M=(0.5*Io*(w2^2-w1^2))/theta //N.m
//result
clc
printf('The constant moment required is %fN.m',M)
```

Scilab code Exa 17.18 Work

```
//Initilization of variables
W=1000 //lb
w_w=200 //lb weight of the individual wheel
d_w=2.5 //ft diameter of the wheel
v=22 //ft/s
t=2 //minutes
//Calculations
//T1=Initial Kinetic Energy and T2=Final Kinetic
   Energy
```

```
F=(-0.5*W*32.2^-1*v^2-4*0.5*w_w*32.2^-1*(v^2+0.5*v^2))/(10560) //lb
//Negative sign in the answer tells it oposses the motion
//Result
clc
printf('The rolling resistance is %flb',F)
```

Scilab code Exa 17.19 Work

```
//Initilization of variables
W=100 //lb
lo=4 //ft
theta=45 //degrees
g=32.2 //ft/s^2
l=8/3 //ft
//Calculations
//Taking moment about point O and equating it to zero
alpha=(W*(lo*0.5)*cosd(theta))/((W/g)*(l)*2) //rad/s^2
//Summing forces in the t direction
Ot=(W*cosd(theta))-((W/g)*lo*0.5*alpha) //lb
//Work Done
Work=W*(lo*0.5*cosd(theta)) //ft/lb
//Moment of inertia
Io=(1/3)*(W/g)*(lo^2) //kg-ft^2
//Using the concept for work done=chane in K.E
w=sqrt(Work/(0.5*Io)) //rad/s
//Summing forces along the bar
On=-(-((W/g)*lo*0.5*w^2)-(W*cosd(theta))) //lb
//Result
clc
printf('The bearing reaction at O on the rod is %flb',On)
```

Scilab code Exa 17.21 Work

```
//Initilization of variables
vo=9 //m/s
theta=30 //degrees
g=9.8 //m/s^2
//Calculations
x=((7/10)*vo^2)/(g*sind(theta)) //m
//Result
clc
printf('The ball will roll %f m up the plane',x)
//The textbook wrongly mentions the unit of
    displacement as in
```

Scilab code Exa 17.22 Work

```
//Initilization of variables
W=322 //lb
F=12 //lb
a=0 //lower limit (where the cyliner starts rolling)
b=%pi/2 //Upper Limit (where the cyliner stops
    rolling)
d=3.2 //ft
g=32.2 //ft/s^2
//Calculations
dR=1.6 //Differential Radius
d_U=2*dR*F //differential work done
//Integration Calculations
//As it is a simple integration we can resort to
    this
U=d_U*(b-a) //ft-lb
```

```
//Determination of K.E
w=sqrt(U/((0.5*(W/g)*(1/(d/2)^2))+((0.5*0.5)*(W/g)*(
    d/2)^2))) //rad/s
//Result
clc
printf('the angular velocity of the is %f rad/s',w)
```

Scilab code Exa 17.23 Work

```
//Initilization of variables
m=90 //kg
k=450 //N/m
lo=0.6 //m
r=0.15 //m
x=0.9 //m
y=0.4 //m
//Calculations
//Initial KE=0
I=0.5*m*r^2 //kg.m^2
s1=sqrt((lo^2)+(x^2)) //m
s2=sqrt((lo^2)+(y^2)) //m
V1=0.5*k*(s1-lo)^2 //N.m
V2=0.5*k*(s2-lo)^2 //N.m
//Applying Conservation of Energy
w=sqrt((V1-V2)/(0.5*m*r^2+0.5*I)) //rad/s
//Result
clc
printf('The value of  angular speed is: %f rad/s',w)
```

Scilab code Exa 17.24 Work

```
//Initilization of variables
Wa=161 //lb
```

```
Wb=193.2 //lb
Wc=322 //lb
v1=5 //ft/s
lc=6 //in
k=6 //lb/ft
l=4 //ft
u=0.2 //coefficient of friction
g=32.2 //ft/s^2
//Calculations
Ib=(1/2)*(Wb/g)*(1/2)^2 //Moment of inertia
w1=v1/0.5 //rad/s
T1=(0.5*(Wc/g)*v1^2)+(0.5*Ib*w1^2)+(0.5*(Wa/g)*v1^2)
    //ft-lb
//Work Done on the system
//The textbook is ambigious on the calculations
   hence the result is dispalyed directly
U=26.4 //ft-lb
//Velocity Calculations
v=sqrt((T1+U)/9) //ft/s
//Result
printf('The velocity of the block is %f',v)
```

Scilab code Exa 17.25 Work

```
//Initilization of variables
Mm=70 //kg
Mc=45 //kg
R=0.6 //m
g=9.8 //m/s^2
l=5 //m
theta=50 //degrees
//Calculations
//T2 calculations except for v term in it as it
   cannot be declared as a number
T2=68.7 //without the v term in it
```

```
v=sqrt((g*Mm*l-g*Mc*l*sind(theta))/T2) //m/s
//Result
clc
printf('The speed is %fm/s',v)
```

Scilab code Exa 17.26 Work

```
//The textbook has a typo in printing the question
   number
//Initilization of variables
W1=96.6 //lb
W2=128.8 //lb
v=8 //ft/s
g=32.2 //ft/s^2
theta=30 //degrees
//Calculations
//Initial KE of the system is T1=0
T2=(0.5*(W1/g)*v^2)+(0.5*(W2/g)*(v/2)^2) //ft-lb
//Work Done without s term
U=-(W1*sind(theta))+W2*0.5
//S calculations
s=T2/U //ft
//Result
clc
printf('The block attains a speed of 8ft/s in %f ft'
   ,s)
```

Scilab code Exa 17.28 Work

```
//Initilization Of Variables
a=0 //Lower Limit oF the Integral
b=6 //Upper Limit of the Integral
n=10 //Interval of the integral
```

```
m=50 //kg
l=6 //m
g=9.8 //m/s^2
//Calculation
//Gravatational Force is
Fg=g*(m/l) //dx
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=Fg*(6-t)')
//Result
clc
printf('The Work done is %f N.m',Trap_Composite1(f,a
    ,b,n))
```

Scilab code Exa 17.31 Work

```
//Initilization Of Variables
x1=150 //mm
x2=450 //mm
a=0 //Lower Limit oF the Integral
b=(x2-x1) //Upper Limit of the Integral
n=10 //Interval of the integral
k=0.044 //N/m
//Calculation
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=k*t')
```

```
//Result
clc
printf('The Work done is %f N.m',Trap_Composite1(f,a
    ,b,n)/1000)
```

Scilab code Exa 17.32 Work

```
//Initilization of variables
m=10 //kg
d=1.2 //m
g=9.8 //m/s^2
//Calculations
//Initilial KE is zero
//Final KE is(without v^2 term in it)
KE2=(3/4)*10
//Work Done
U=m*g*d //N.m
//Velocity calculations
v=sqrt(U/KE2) //m/s
//Result
clc
printf('The velocity is %fm/s',v)
```

Scilab code Exa 17.33 Work

```
//Initilization of variables
W=161 //lb
wa=150 //lb
wb=100 //lb
la=2 //ft
lb=4 //ft
//Calculations
//Work Done
```

```
T1=wb*lb-wa*la //ft-lb
//Final KE=zero
T2=0 //ft-lb
//Work Done on the system=T2-T1
//Hence the equation becomes
//50x-50x^2+100=0
//where
a=-50
b=50
c=100
//Solution
d=sqrt(b^2-4*a*c)
x1=(-b+d)/(2*a) //ft
x2=(-b-d)/(2*a) //ft
//Result
clc
printf('The stretch of the spring is %f',x2)
//Here even x1 could have been the solution,but the
   stretch in the string is elongation not
   compression hence x2 is the valid answer
```

Scilab code Exa 17.34 Work

```
//Initilization of variables
I=100 //slug-ft^2
w=4 //rad/s
theta=6 //rad
Mc=64.4 //lb
g=32.2 //ft/s^2
//Calculations
vb=2*w //ft/s
vc=0.5*w //ft/s
Mb=(0.5*I*w^2+0.5*(Mc/g)*vc^2+0.5*Mc*theta)/(2*theta
   -(0.5*vb^2*(1/g))) //lb
//Result
```

```
clc
printf('The weight of the block B is %f lb',Mb)
```

Scilab code Exa 17.35 Work

```
//Initilization of variables
Wa=96.6 //lb
Wb=128.8 //lb
g=32.2 //ft/s^2
I=12 //slug-ft^2
v=16 //ft/s
ratio=1/3 //ratio of Sb/Sa
r=3//ft
va=6 //ft/s
vb=2 //ft/s
//Calculations
//Work Done without S in it
W=Wa-(ratio*Wb)
//System has zero KE initially and final KE is given
    by
w=va/r //rad/s
T2=(0.5*(Wa/g)*va^2+0.5*I*w^2+0.5*(Wb/g)*vb^2) //ft-
   lb
//Distance Calculations
S=T2/W //ft
//Result
clc
printf('The distance through which A falls is %f ft'
   ,S)
```

Scilab code Exa 17.36 Work

```
//initilization of variables
```

```
u=0.25 //coefficient of friction
k=2800 //N/m
x=0.075 //m
g=9.8 //m/s^2
m=7 //kg
theta=30 //degrees
//Calculations
//Normal Reaction
N=g*m*cosd(theta) //N
//Frictional Force
Fr=u*N //N
//Component of force along the plane
F=g*m*sind(theta) //N
//Spring work is
W=0.5*k*x*x //N.m
s=(W+Fr*x-F*x)/(F-Fr) //m
S=round(s*1000) //mm
//Result
clc
printf('The value of S is %i mm',S)
```

Scilab code Exa 17.37 Work

```
//Initilization of variables
m=5 //kg
l=2 //m
k=10000 //N/m
x=0.1 //m
g=9.8 //m/s^2
//Calculations
drop=l+x //m mass drop length
//Work Done by Gravity
Wg=g*m*drop //N.m
//Work Done by Spring
Ws=0.5*k*x^2 //N.m
```

```
//Increase in KE is without v^2
KE=0.5*m //kg
//Velocity Calculations
v=sqrt((Wg-Ws)/KE) //m/s
//Result
clc
printf('The speed is %f m/s',v)
```

Scilab code Exa 17.38 Work

```
//Initilization of variables
l=6 //ft
k=20 //lb/in
x=8 //in
//Calculations
//Work Done by Gravity
Wg=(l*12+x) //in without W
//Work Done by Spring
Ws=0.5*k*x^2 //in-lb
//Change in the kinetic energy is zero
W=Ws/Wg //lb
//Result
clc
printf('The weight is %i lb',W)
```

Scilab code Exa 17.40 Work

```
//Initilization of variables
W=8 //lb
//Calculations
//work done by the spring woithout k
Ws=0.5*((9/12)^2-(1/12)^2)
//Work done by gravity
```

```
Wg=W*(10.5/12) //ft-lb
//Change in KE is zero
k=Wg/Ws //lb/ft
//Result
clc
printf('The value of k is %f lb/ft',k)
```

Scilab code Exa 17.41 Work

```
//Initilization of variables
Wc=100 //lb
r= 1 //ft
F=80 //lb
k=50 //lb/ft
s=6 //in
g=32.2 //ft/s^2
//Calculations
//Work done on the system
U=-0.5*k*(1)+F*(s/12) //ft-lb
//Initial KE is zero
Vo=sqrt(U/(0.5*(Wc/g+0.5*(Wc/g)*r))) //ft/s
//Result
clc
printf('The initial speed is %f ft/s',Vo)
```

Chapter 18

Impulse and Momentum

Scilab code Exa 18.8 Imp

```
//Initilization of variables
W=100 //lb
u=0.2 //coefficient of friction
t=5 //s
v1=5 //ft/s
v2=10 //ft/s
g=32.2 //ft/s^2
ll=0 //lower limit of integration
ul=5 //upper limit of integration
//Calculations
Fr=u*W //lb
//Using The impulse momentum theorem
//Since the integration is just subtraction of
   limits we can skip that
F=((W/g)*v2-(W/g)*v1+Fr*ul)/ul //lb
//Result
clc
printf('The Force is %f lb',F)
```

Scilab code Exa 18.9 Imp

```
//Initilization of variables
m=10 //kg
theta=30 //degrees
u=0.3 //coefficient of kinetic friction
t=5 //s
g=9.8 //m/s^2
//Calculations
//Asthere is no motion in the vertical direction
//Summing forces along vertical direction
Na=m*g*cosd(theta) //N
//Using impulse momentum theorem
vx=(m*g*sind(theta)-u*Na)*(t/m) //m/s
//Result
clc
printf('The spedd after 5s is %f m/s',vx)
```

Scilab code Exa 18.10 Imp

```
//Initilization Of Variables
a=1 //Lower Limit oF the Integral
b=5 //Upper Limit of the Integral
n=10 //Interval of the integral
W=80 //lb
us=0.25 //coefficient of static friction
uk=0.20 //coefficient of kinetic friction
g=32.2 //ft/s^2
//Calculation
//Limiting Force
F=W*uk //lb
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
```

```
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=t*20')
//Using Impulse momentum theorem
l=Trap_Composite1(f,a,b,n) //storing integration
   value in a variable
v=(g/W)*(l-F*(b-a)) //ft/s
//Result
clc
printf('The speed of the block is %f ft/s',v)
```

Scilab code Exa 18.11 Imp

```
//Initilization of variables
m=1.5 //kg
i1=2 //Integral constant obtained after integrating
i2=3.33 //Integral constant obtained after
   integrating
//Calculations
//As indefinite integrals are not possible to code
//We directly consider the intergral which is a
   simple integral
//v=t^2-1.11*t^3
//After we derivate this expression we obtain
t=i1/i2 //s
//Now using the formula for v we get
v=t^2-((i2/3)*t^3) //ft/s
//Result
clc
printf('The maximum velocity is:%f m/s',v)
```

Scilab code Exa 18.12 Imp

```
//Initilization of variables
m1=40 //kg
m2=10 //kg
m3=15 //kg
v0=2.5 //m/s
vf=5 //m/s
t=12 //s
u=0.1 //coefficient of friction
g=9.8 //m/s^2
theta=45 //degrees
//Calculations
//Applying Impulse-Momentum Theoroem
P=(((m1+m2+m3)*(vf-v0))+(t*(-m2*g*sind(theta)+u*g*m2
    *cosd(theta)+u*g*m3+g*m1)))/t //N
//Result
clc
printf('The value of P is %f N',P)
```

Scilab code Exa 18.13 Imp

```
//Initilization of variables
W1=4 //lb
W2=2 //lb
t2=0.04 //s
W3=-2 //lb
t3=0.02 //s
t=3 //s
g=32.2 //ft/s^2
//Calculations
//Algebraic sum of two areas
A=t2*W2+t3*W3 //lb-s
//Using Impulse Momentum Theorem
v=(A*g)/W1 //ft/s
//Result
clc
```

```
printf('The spped after 3s is %f ft/s',v)
```

Scilab code Exa 18.14 Imp

```
//Initilization of variables
f_r=1 //in/s rate of fall of mercury
ll=18 //in length of left column
lr=22 //in length of right column
rho=850 //lb/ft^3
d=1/4 //in
g=32.2 //ft/s^2
//Calculations
//Applying Impulse momentum theorem
M=((d*%pi*d^2*4)/12^3)*(rho/g)*(1/12) //lb-s
//Result
clc
printf('The upward momentum of mercury is %f lb-s',M
    )
```

Scilab code Exa 18.15 Imp

```
//Initilization of variables
m=2000 //kg
k=1.200 //m
w=120 //rpm
t=200 //s
//Calculations
//Applying Angular Momentum theorem
M=((m*k^2*(w*2*%pi))/60)/t //N.m
//Result
clc
printf('The Momentum necessary is %f N.m',M)
```

Scilab code Exa 18.17 Imp

```
//Initilization of variables
m1=5 //kg
m2=7 //kg
mp=5 //kg
r=0.6 //m
k=0.45 //m
vi=3 //m/s
vf=6 //m/s
g=9.8 //m/s^2
//Calculations
I=m1*k^2 //kg.m^2
wnet=(vf/r)-(vi/r) //rad/s
//Solving the system of linear equations
//Simplfying the equation we get
t=((I*wnet)+m1*(vf-vi)+m2*(vf-vi))*r/(r*(m2-m1)*g)
    //s
//Result
clc
printf('The time required is %f s',t)
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 18.18 Imp

```
//Initilization of variables
m=50 //kg
vo=4 //m/s
vf=8 //m/s
t=6 //s
g=9.8 //m/s^2
r=0.8 //m
```

```
u=0.25 //coefficient of friction
I=30 //kg-m^2
//Calculations
Na=m*g //N
F=u*Na //N
//Angular Speeds
wo=vo/r //rad/s
wf=vf/r //rad/s
//Applying impulse momentum theorem
mb=(I*wf+m*vf*r-I*wo-m*vo*r+F*r*t)/(vo*r+g*r*t-vf*r)
    //kg
//Result
clc
printf('The mass of block B is %f kg',mb)
```

Scilab code Exa 18.19 Imp

```
//Initilization of variables
Ws=250 //lb
Wl=500 //lb
W3=161 //lb
W4=64.4 //lb
wo=100 //rpm
wf=300 //rpm
rl=3 //ft
rs=2 //ft
g=32.2 //ft/s^2
//Calculations
//Moment Of Inertia
I=(0.5*(Wl/g)*rl^2+0.5*(Ws/g)*rs^2) //slug-ft^2
//Change in angular Momentum
change1=I*((wf-wo)*2*(%pi/60)) //lb-s-ft
//Change in angular momentum about G for 161lb
change2=2*((W3/g)*(wf-wo)*(4/60)*%pi) //lb-s-ft
//Similarly change in 64lb is
```

```
change3=3*((W4/g)*(wf-wo)*(6/60)*%pi) //lb-s-ft
//Change in linear impulse
//Without t term in it
m1=2*W3
m2=-3*W4
//Total angular impulse
t=(change1+change2+change3)/(m1+m2) //s
//Result
clc
printf('The time required is %f s',t)
```

Scilab code Exa 18.21 Imp

```
//Initilization of variables
d=3 //ft
W=300 //lb
theta=20 //degrees
F=250 //lb
t=6 //s
vo=0 //ft/s
g=32.2 //ft/s^2
//Calculations
//Applying linear impulse momentum theorem
//Solving by matrix method
A=[-W/g,1*t;-(0.5*(W/g)*d^2*0.5^2)/(d/2),-t*d/2]
B=[-F*t+W*sind(theta)*t;-F*(d/2)*6]
C=inv(A)*B
//Result
clc
printf('The speed after 6s is %f ft/s',C(1))
```

Scilab code Exa 18.22 Imp

```
//Initilization of variables
theta=30 //degrees
vo=20 //ft/s
r=4 //ft
vf=0 //ft/s
g=32.2 //ft/s^2
//Calculations
wo=vo/r //rad/s
wf=vf/r //rad/s
//Applying impulse momentum theorem
//Solving simultaneous equations
t=-((3/(2*g))*(r^2)*(wf-wo))/(r*sind(theta))//s
//Result
clc
printf('The time t is %f s',t)
```

Scilab code Exa 18.23 Imp

```
//Initilization of variables
mw=75 //kg
k=0.9 //m
wi=10 //rad/s
wf=6 //rad/s
r=1.2 //m
m=30 //kg
g=9.8 //m/s^2
//Calculations
//Initial speed
vi=-r*wi //m/s
vf=r*wf //m/s
//Initial speed of B is
vib=-0.8*wi+vi //m/s
//Similarly
vfb=12 //m/s
//Applying impulse momentum theorem
```

```
//Solving by matrix method
A=[1,-1,-(mw*(vf-vi));0.8,1.2,-(mw*(k^2)*(wf+wi))
   ;-1,0,-(m*(vfb-vi))]
B=[0;0;-g*m]
C=inv(A)*B
//Here t is calculated as 1/t for simplicity
//Result
clc
printf('The time required is %f s',1/C(3))
//Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 18.24 Imp

```
//Initilization of variables
d=8 //in
W=96.6 //lb
w=36 //rad/s
u=0.15 //coefficient of friction
g=32.2 //ft/s^2
//Calculations
r=(d/2)/12 //m
N=W //lb
F=u*N //lb
m=W/g //slugs
I=0.5*m*(r^2) //slug-ft^2
//Applying the impulse momentum theorem
//Solving the two equations simultaneously
A=[F,-m;F*r,I*(1/r)]
B=[0;w*I]
C=inv(A)*B
//Distance travelled
s=0.5*C(2)*C(1) //ft
t=C(1) //s
//Result
printf('The time required is %f s,and it travels %f
```

```
    ft',t,s)
```

Scilab code Exa 18.25 Imp

```
//Initilisation of variables
d=2/12 //ft
v=80 //ft/s
g=32.2 //ft/s^2
//Calculations
//Mass flow reate without time
m=(1/4)*%pi*d^2*v*(62.4/g)
//Let P=force of plate on mass m of water
P=m*(0-v) //lb
//Result
clc
printf('The force water exerts on the plate is %f lb
    ',-P )
```

Scilab code Exa 18.26 Imp

```
//Initilization of variables
v1=20 //ft/s
vw=80 //ft/s
d=2/12 //ft
g=32.2 //ft/s^2
//Calculations
v=vw-v1 //ft/s
//mass flow rate without t
m=(1/4)*(%pi*d^2)*(62.4/g)*v
//Applying impulse momentum theorem
P=m*v //lb
//Result
clc
```

```
printf('The force exerted by water on the plate is
    %f lb',P)
```

Scilab code Exa 18.27 Imp

```
//Initilisation of variables
W=150 //lb
v=20 //ft/s
A=0.2 //in^2
t=60 //s
g=32.2 //ft/s^2
//Calculations
//Mass flow
m=(A/12^2)*v*(62.4/g)
//Force
F=m*(0-v) //lb
//At t=60s the tank holds
Ww=(A/12^2)*v*t*62.4 //lb
//Total reading on scale
S=-F+W+Ww //lb
//Result
clc
printf('The scale reading is %f lb',S)
```

Scilab code Exa 18.28 Imp

```
//Initilisation of variables
W=150 //lb
v=20 //ft/s
A=0.2 //in^2
t=60 //s
g=32.2 //ft/s^2
//Calculations
```

```
//Mass flow
m=(A/12^2)*v*(62.4/g)
//Force
F=m*(0-v) //lb
//At t=60s the tank holds
Ww=(A/12^2)*v*t*62.4 //lb
//Total reading on scale
S=-F+W+Ww //lb
//Result
clc
printf('The scale reading is %f lb',S)
```

Scilab code Exa 18.29 Imp

```
//Initilization of variables
Wp=130 //lb
Wb=150 //lb
Wbullet=2/16 //lb
g=32.2 //ft/s^2
vbullet=1200 //ft/s
//Calculations
v=((-Wbullet/g)*vbullet)/((Wb+Wp)/g) //ft/s
//Result
clc
printf('The speed of the boat is %f ft/s',v)
//Negative sign indicates direction opposite to that
    of bullet
```

Scilab code Exa 18.30 Imp

```
//Initilization of variables
mb=0.06 //kg
ms=50 //kg
```

```
h=0.03 //m
g=9.8 //m/s^2
//Calculations
//Speed of bag+bullet
v2=sqrt(2*g*h) //m/s
//Applying conservation of momentum
v1=((mb+ms)*v2)/mb //m/s
//Result
clc
printf('The speed of bullet as it entered the bag
    was %f m/s',v1)
```

Scilab code Exa 18.32 Imp

```
//Initilization of variables
mb=0.06 //kg
vb=500 //m/s
mblock=5 //kg
vblock=30 //m/s
//Calculations
//Applying conservation of momentum
v=(mb*vb+mblock*vblock)/(mb+mblock) //m/s
//Result
clc
printf('The speed of the system is %f m/s',v)
```

Scilab code Exa 18.33 Imp

```
//Initilization Of Variables
W1=2 //lb
W2=3 //lb
a=0 //Lower Limit oF the Integral
b=2 //Upper Limit of the Integral
```

```
n=10 //Interval of the integral
k=12/12 //lb/ft
g=32.2 //ft/s^2
//Calculation
//Work Done by the spring
//Using Trapezoidal Rule for Intergration
function[I1]=Trap_Composite1(f,a,b,n)
    h=(b-a)/n
    t=linspace(a,b,n+1)
    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
endfunction
deff('[y]=f(t)','y=k*(2-t)')
W=Trap_Composite1(f,a,b,n) //ft-lb
//Solving the simultaneous equations
v3=sqrt(W/(0.5*(W2/g)+0.5*(W1/g)*(-W2/W1)^2)) //ft/s
v2=-(W2/W1)*v3 //ft/s
//Result
clc
printf('The speed of 2lb block is %f ft/s and that of 3lb block is %f ft/s',v2,v3)
```

Scilab code Exa 18.34 Imp

```
//Initilization of variables
//Here the integration is indefinite hence it will be computed manually and entered
W=10 //lb
l=4 //ft
w=2 //rad/s
g=32.2 //ft/s^2
//Calculations
//Part (a)
wf=1.5 //rad/s
t=sqrt(((W/g)*(l*w*l))-((W/g)*(l*wf*l))) //s
//Part (b)
```

```
//Applying conservation of angular momentum
r=((W/g)*l*wf*l)/((W/g)*l*w) //ft
//Result
clc
printf('The answer for part (a) is %f s\n and the
   answer for part (b) is %f ft',t,r)
```

Scilab code Exa 18.35 Imp

```
//Initilization of variables
W=2.5 //lb
w=36 //rad/s
Idisk=0.4 //slug-ft^2
g=32.2 //ft/s^2
//Calculations
Ii=Idisk+(2*(W/g)*(3/12)^2) //slug-ft^2
If=Idisk+(2*(W/g)*(11/12)^2) //slug-ft^2
//Since no external moments act
//Applying conservation of momentum
wf=(Ii*w)/If //rad/s
//Result
clc
printf('The final angular speed is %f rad/s',wf)
```

Scilab code Exa 18.39 Imp

```
//Initilization of variables
u1=6 //ft/s
u2=-8 //ft/s
e=0.8 //coefficient of restitution
//Calculations
//Solving both simultaneous equations
A=[1,-1;1,1]
```

```
B=[11.2;-2]
C=inv(A)*B //ft/s
//Result
clc
printf('The velocities are v1=%f ft/s and v2=%f ft/s
   ',C(2),C(1))
```

Scilab code Exa 18.40 Imp

```
//Initilization of variables
h1=20 //m
h2=14 //m
g=9.8 //m/s^2
//Calculations
u1=sqrt(2*g*h1) //m/s
u2=0 //m/s
v1=-sqrt(2*g*h2) //m/s
v2=0 //m/s
e=(v2-v1)/(u1-u2) //coefficient of restitution
//Result
clc
printf('The value of coefficient of restitution is
   %f',e)
```

Scilab code Exa 18.41 Imp

```
//Initilization of variables
u=6.55 //ft/s
g=32.2 //ft/s^2
L=6 //ft
W=5 //lb
c=0.7 //fraction of impulse acting in second phase
//Calculations
```

```
//Impulse
I=(W/g)*(u/3) //N.s
//Second Phase
v=(u*2-(c*(W*u*(1/3))))  //ft/s
wprime=(1.09*60+c*(W*u*(1/3)*6))/60 //rad/s
//Result
clc
printf('The value of v=%f ft/s and that of w is %f
    rad/s',v,wprime)
//The value of w is incorrect in the textbook
```

Scilab code Exa 18.42 Imp

```
//Initilization of variables
m1=9 //kg
m2=5.5 //kg
u1=-3 //m/s
u2=1.77 //m/s
e=0.8 //coefficient of restitution
//Calculations
//Solving by matrix method after we get the two
    equations
A=[-1,1;m1,m2]
B=[(e*u1-e*u2);m1*u1+m2*u2]
C=inv(A)*B //m/s
//Result
clc
printf('The 9kg ball will rebound up the speed of %f
     m/s\n and the 5.5kg ball will move to the right
    and down \nwith components of %f m/s and %f m/s
    respectively',C(1),u2,-C(2))
```

Scilab code Exa 18.46 Imp

```
//Initilization of variables
v=4 //m/s
m=9 //kg
s=1.5 //m
//Calculations
Io=(2/3)*(m*s^2) //kg.m
w=(m*v*s*0.5)/Io //rad/s
//Result
clc
printf('The angular velocity of the box is %i rad/s'
    ,w)
```

Scilab code Exa 18.48 Imp

```
//Initilization of variables
m=2000 //kg
mf=8500 //kg
vr=2000 //m/s
a=9.8 //m/s^2
//Calculations
//Applying Newtons Second Law
dm_dt=-(-((m+mf)*a)-(m+mf)*a)/(-vr) //kg/s
//Result
clc
printf('dm/dt=%f kg/s',dm_dt)
//The negative sign indicates the loss in the mass
    of the system
```

Scilab code Exa 18.50 Imp

```
//Initilization of variables
W=4000 //lb
k=3 //ft
```

```
wp=(1/60)*2*%pi //rad/s
ws=(300/60)*2*%pi //rad/s
d=3.5 //ft
g=32.2 //ft/s^2
//Calculations
I=(W/g)*k^2 //slug-ft^2
M=I*ws*wp //lb-ft
//Now equating M to Rf-Rr gives one equations and
   vertical sum yields other
//solving them by matrix method
A=[1,-1;1,1]
B=[M*(2/d);W]
C=inv(A)*B //lb
//Result
clc
printf('The weight of Rf and Rr are %f lb and %f lb
   respectively',C(1),C(2))
```

Scilab code Exa 18.51 Imp

```
//As the integration is indefinite we will directly
   consider the equation with R
//Initillization of variables
GM=1.41*10^16 //ft^3/s^2
r=2640000 //ft
theta=60 //degrees
R=21120000 //ft
//Calculations
v1=sqrt((GM*((1/R)-(1/(R+r))))/2.031) //ft/s
//Result
clc
printf('The speed required will be %f ft/s',v1)
```

Scilab code Exa 18.52 Imp

```
//Initilization of variables
k=4 //lb/ft
so=1 //ft
W=1/2 //lb
g=32.2 //ft/s^2
vo=5 //ft/s
//Calculations
m=W/g //kg
//Angular momentum is conserved
v=sqrt((0.5*k*so^2*2*2*g)+vo^2) //ft/s
//Using vd=15
d=15/v //ft
//Result
clc
printf('The ball passes %f ft close to the fixed pin
    ',d)
```

Chapter 19

Mechanical Vibration

Scilab code Exa 19.2 Vibrations

```
//Initilization of variables
k=18 //lb/in
g=386 //in/s^2
W=35 //lb
//Calculations
f=(1/(2*%pi))*sqrt((k*g/W)) //cps
period=1/f //s
//Result
clc
printf('The period of vibration is %f s',period)
```

Scilab code Exa 19.11 Vibrations

```
//Initilization of variables
ds=0.2 //m
ts=0.05 //m
rhos=7850 //kg/m^3 density of  steel
dw=0.002 //m
```

```
lw=0.9 //m
G=80*10^9 //Pa
//Calculations
//Torsional Constant
K=(%pi*dw^4*G)/(32*lw) //m/rad
//Mass Calculations
m=(1/4)*%pi*(ds^2)*ts*rhos //kg
//Moment of Inertia
Io=(1/2)*m*(ds/2)^2 //kg.m^2
//Frequency
f=(1/(2*%pi))*(sqrt(K/Io)) //Hz
//Result
clc
printf('The natural frequency of the system is %f Hz',f)
```

Scilab code Exa 19.13 Vibrations

```
//Initilization of variables
m=120 //kg
k=0.3 //m
ls=0.6 //m
ds=0.05 //m
G=80*10^9 //Pa
//Calculations
//Polar Moment of Inertia
J1=m*k^2 //kg.m^2
J2=J1 //kg.m^2
J=(1/32)*%pi*(ds^4) //m^4
//Frequency
f=(1/(2*%pi))*(sqrt((J*G*(J1+J2))/(ls*J1*J2))) //Hz
//Result
clc
printf('The natural frequency of the torsional oscillation is %f Hz',f)
```

Scilab code Exa 19.14 Vibrations

```
//Initilization of variables
ds=2 //in
L=15 //in
Wf1=300 //lb
k1=6 //in
Wf2=100 //lb
k2=4 //in
G=12*10^6 //Pa
g=386 //in/s^2
//Calculations
//Moment of inertia of flywheel
Jf=(Wf1/g)*k1^2 //lb-s^2-in
//Moment of inertia of the rotor
Jr=(Wf2/g)*k2^2 //lb-s^2-in
//Moment of inertia of the shaft cross section
J=(1/32)*%pi*ds^4 //in^4
//Frequency
f=(1/(%pi*2))*(sqrt((J*G*(Jf+Jr))/(L*Jf*Jr))) //cps
//result
clc
printf('The natural frequency of the system is %f
    cps',f)
```

Scilab code Exa 19.15 Vibrations

```
//Initilization of variables
W=10 //lb
A=2 //in^2
//Calculations
```

```
wn=sqrt(((A/144)*5*62.4*5)/2.59) //Hz
//Result
clc
printf('The frequency of oscillation is %f Hz',wn)
```

Scilab code Exa 19.16 Vibrations

```
//Initilization of variables
f=50 //cps
g=386 //in/s^2
E=30*10^6 //lb/in^2
l=4 //in
I=2.08*10^-6 //in^4
//Calculations
W=(3*E*I*g)/(((f*2*%pi)^2)*l^3) //lb
//Result
clc
printf('The value of W is %f lb',W)
```

Scilab code Exa 19.19 Vibrations

```
//Initilization of variables
F=10 //lb
v=20 //in/s
g=386 //in/s
W=12 //lb
k=20 //lb/in
//Calculations
//Coefficient of damping
c=F/v //lb-s/in
//Natural Frequency
wn=sqrt((k*g)/W) //rad/s
//Critical Damping coefficient
```

```
cr=(2*W/g)*wn //lb-s/in
//Damping Coefficient
d=c/cr
//Frequency of damped vibrations
wd=sqrt(1-d^2)*wn //rad/s
//Result
clc
printf('The frequency of damped vibrations is %f rad
    /s',wd)
```

Scilab code Exa 19.20 Vibrations

```
//Initilization of variables
wn=25.4 //rad/s
t=0.261 //s
d=0.316
//Calculations
del=d*t*wn //logarithmic decay
//Result
clc
printf('The rate of decay is %f',del)
```

Scilab code Exa 19.24 Vibrations

```
//Initilization of variables
F=9 //N
m=5 //kg
k=6000 //N/m
f1=1 //Hz
f2=5.4 //Hz
f3=50 //Hz
//Calculations
//Natural Frequency
```

```
fn=(1/(%pi*2))*(sqrt(k/m)) //Hz
deltaf=F/(k/1000) //mm
//Part(a)
r1=f1/fn
amp1=deltaf/(1-r1^2) //mm amplitude
//Part (b)
r2=f2/fn
amp2=deltaf/(1-r2^2) //mm amplitude
//Part (c)
r3=f3/fn
amp3=deltaf/(1-r3^2) //mm amplitude
//Result
clc
printf('The amplitudes in part (a),(b) and (c)
   respectively are \n %f mm,%f mm and %f mm
   respectively',amp1,amp2,amp3)
```

Scilab code Exa 19.25 Vibrations

```
//Initilization of vraiables
g=386 //in/s^2
W=20 //lb
w=600 //rpm
ratio=1/12
//Calculations
r=sqrt((1/ratio)+1)
fn=((w/60)/r) //cps
k=((fn*2*%pi)^2*W)/(g) //lb/in
//Result
clc
printf('The value of k is %f lb/in',k)
```

Scilab code Exa 19.28 Vibrations

```
//Initilization of variables
X=12 //mm
me_M=1.3 //mm
//Calculations
d=(me_M)/(2*X)
//Result
clc
printf('The damping ratio is %f',d)
```