Scilab Textbook Companion for Concepts Of Physics (Volume - 1) by H. C. Verma[1]

Created by
Patel Chaitanya Kishorbhai
B. Tech.
Others
Dharmsinh Desai University
College Teacher
Prarthan Mehta
Cross-Checked by
K. V. P. Pradeep

June 12, 2014

[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: Concepts Of Physics (Volume - 1)

Author: H. C. Verma

Publisher: Bharati Bhavan,india

Edition: 2

Year: 2011

ISBN: 9788177091878

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 2

Physics and Mathematics

Scilab code Exa 2.1w calculation of magnitude and direction of vector

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.1w
//calculation of magnitude and direction of vector

//given data
xcomp=25; //value of component along X axis
ycomp=60; //value of component along Y axis
theta=90; //angle between X and Y axis

//calculation
A=sqrt((xcomp*xcomp)+(ycomp*ycomp)+(2*xcomp*ycomp*
    cosd(theta)));
alpha=atand(ycomp/xcomp);

disp(A,'magnitude of the vector is');
disp(alpha,'direction of the vector is');
```

Scilab code Exa 2.2 calculation of sum of vectors and difference of the vectors

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.2
//calculation of sum of vectors and difference of
    the vectors

//given data
A=5; //magnitude(in unit) of A vector
B=5; //magnitude(in unit) of B vector
theta=60; // angle(in degree) between both vectors


//calculation
C=sqrt((A*A)+(B*B)+(2*A*B*cosd(theta))); //C=|A+B|
    sum of two vectors
thetas=180-theta; //for difference(subtraction)
    reverse direction of a vector and add it to other
D=sqrt((A*A)+(B*B)+(2*A*B*cosd(thetas))); //D=|A-B|
    difference of two vectors

disp(C,'the sum of two vectors(in unit) is');
disp(D,'the difference of two vectors(in unit) is');
```

Scilab code Exa 2.2w calculation of resultant of three vectors

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.2w
//calculation of resultant of three vectors

```

```
//given data
theta1=37; //value of angle(in degree) of first vector with X axis
theta2=0; //value of angle(in degree) of second vector with X axis
theta3=90; //value of angle(in degree) of third vector with X axis
x=5; //magnitude(in m) of first vector
y=3; //magnitude(in m) of second vector
z=2; //magnitude(in m) of third vector

//calculation
xcomp1=x*cosd(theta1); //xcomponent(in m) of first vector
ycomp1=x*sind(theta1);//ycomponent(in m) of first vector
xcomp2=y*cosd(theta2);//xcomponent(in m) of second vector
ycomp2=y*sind(theta2);//ycomponent(in m) of second vector
xcomp3=z*cosd(theta3);//xcomponent(in m) of third vector
ycomp3=z*sind(theta3);//ycomponent(in m) of third vector

xcompr=xcomp1+xcomp2+xcomp3; //xcomponent(in m) of resultant vector
ycompr=ycomp1+ycomp2+ycomp3; //ycomponent(in m) of resultant vector

r=sqrt((xcompr*xcompr)+(ycompr*ycompr)); //magnitude (in m) of resultant vector
theta=atand(ycompr/xcompr); //value of angle(in degree) of resultant vector with X axis

disp(r,'magnitude(in m) of resultant vector is');
disp(theta,'value of angle(in degree) of resultant vector with X axis');
```

Scilab code Exa 2.3 calculation of component of force in vertical direction

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.3
//calculation of component of force in vertical
    direction

//given data
F=10.5 //force(in newton) acting on the particle
theta=37 //angle(in degree) at which force acts

//calculation
Fp=F*cosd(theta); //component of force in vertical
    direction

disp(Fp,'component of force(in newton) in vertical
    direction is');
```

Scilab code Exa 2.3w calculation of resultant of the vectors

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.3w
//calculation of resultant of the vectors

//given data
//theta1=90; value of angle(in degree) of OA vector
//theta2=0; value of angle(in degree) of OB vector
```

```
//theta3=135; value of angle(in degree) of OC vector
OA=5; //magnitude(in m) of OA vector
//OB=magnitude(in m) of OB vector
//OC=magnitude(in m) of OC vector

//calculation
//xcomp1=0; xcomponent(in m) of OA vector
//ycomp1=-OA; ycomponent(in m) of OA vector
//xcomp2=OB; xcomponent(in m) of OB vector
//ycomp2=0; ycomponent(in m) of OB vector
//xcomp3=(-1/sqrt(2))*OC; xcomponent(in m) of OC vector
//ycomp3=(1/sqrt(2))*OC;ycomponent(in m) of OC vector

//xcompr=OB-((1/sqrt(2))*OC); xcomponent(in m) of resultant vector=0(given)        (1)
//therefore OB=((1/sqrt(2))*OC)        (2)
//ycompr=((1/sqrt(2))*OC)-OA; ycomponent(in m) of resultant vector
//((1/sqrt(2))*OC)=OA        (3)

OC=sqrt(2)*OA; //from equation (3)
OB=((1/sqrt(2))*OC) //from equation(2)

disp(OC,'magnitude(in m) of OC vector is');
disp(OB,'magnitude(in m) of OB vector is');
```

Scilab code Exa 2.4 calculation of work done by the force during displacement

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.4
//calculation of work done by the force during
   displacement

//given data
F=12; //force(in newton) acting on the particle
r=2; //displacement(in m) of the particle
theta=180; //angle(in degree) between force and
   displacement

//calculation
W=F*r*cosd(theta);//formula of work done

disp(W,'work done(in J) by the force,during the
   given displacement is');
```

Scilab code Exa 2.4w calculation of direction of resultant vector

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.4w
//calculation of direction of resultant vector

//given data
//OA=OB=OC=F all the three vectors have same
   magnitude
//xcompOA=F*cos30=(F*(sqrt(3)))/2
//xcompOB=F*cos360=F/2
//xcompOC=F*cos135=-F/(sqrt(2))
//xcompr=xcompOA + xcompOB + xcompOC

```

```
//ycompOA=F*cos60=F/2
//ycompOB=F*cos360=-(F*(sqrt(3)))/2
//ycompOC=F*cos135=F/(sqrt(2))
//ycompr=ycompOA + ycompOB + ycompOC

//calculation
theta=atand((1-sqrt(3)-sqrt(2))/(1+sqrt(3)+sqrt(2)))
    ;

disp(theta,'the angle(in degree) made by OA+OB-OC
    vector with X axis is');
```

Scilab code Exa 2.5 calculation of angle between two vectors from known value of their cross product

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.5
//calculation of angle between two vectors from
    known value of their cross product

//given data
C=15; //magnitude(in unit) of cross product of two
    vectors,C=|A*B|
A=5; //magnitude(in unit) of A vector
B=6; //magnitude(in unit) of B vector
//calculation
theta=asind(C/(A*B)); //formula for cross product

printf("angle(in degree) between the given two
    vectors is %d or %d",theta,180-theta);
```

Scilab code Exa 2.6 calculation of the slope of curve at a given point

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.6
//calculation of the slope of curve at a given point

//given data
AB=5; //length of AB line segment
BC=4; //length of BC line segment
DE=5; //length of DE line segment
EF=-4; //length of EF line segment

//calculation
m1=AB/BC; //formula of slope,m1=dy/dx at x=2
//m2=0 since tangent to curve at x=6 is parallel to
   x axis
m2=0;
m3=DE/EF; //formula of slope,m2=dy/dx at x= 10

disp(m1,'the slope of the curve at x=2 is');
disp(m2,'the slope of the curve at x=6 is');
disp(m3,'the slope of the curve at x=10 is');
```

Scilab code Exa 2.6w calculation of angle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.6w
//calculation of angle

//given data
xcompOA=4; //magnitude(in m) of x component of OA
```

```
    vector
//xcompOB=6*cos(theta) magnitude(in m) of x
    component of OB vector

//calculation
theta=acosd(-xcompOA/6);//since xcompOA + xcompOB=0
        where xcompOB=6*cos(theta)

disp(theta,'the value of angleAOB(in degree) is');
```

Scilab code Exa 2.7w calculation of unit vector

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.7w
//calculation of unit vector

//given data
ax=5; //x component of A vector
ay=1; //y component of A vector
az=-2; //z component of A vector

//calculation
A=sqrt((ax*ax)+(ay*ay)+(az*az));
uax=ax/A; //x component of unit vector of A vector
uay=ay/A; //y component of unit vector of A vector
uaz=az/A; //z component of unit vector of A vector

disp(uax,'x component of unit vector of A vector');
disp(uay,'y component of unit vector of A vector');
disp(uaz,'z component of unit vector of A vector');
```

Scilab code Exa 2.9 evaluation of a integral

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.9
//evaluation of a integral

//given data
//function of x=(2*x^2)+(3*x)+5)
//limit=3 to 6

//calculation
y=integrate('((2*x^2)+(3*x)+5)','x',3,6)

disp(y,'value of the given integral is')
```

Scilab code Exa 2.9w calculation of angle between two vectors

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.9w
//calculation of angle between two vectors

//given data
ax=2; //xcomponent of A vector
ay=3; //ycomponent of A vector
az=4; //zcomponent of A vector

bx=4; //xcomponent of B vector
by=3; //ycomponent of B vector
bz=2; //zcomponent of B vector

//calculation
```

```
adotb=((ax*bx)+(ay*by)+(az*bz));
a=sqrt((ax*ax)+(ay*ay)+(az*az));
b=sqrt((bx*bx)+(by*by)+(bz*bz));
theta=acosd(adotb/(a*b)); //formula of dot product

disp(theta,' angle(in degree) between given two
   vectors is')
```

Scilab code Exa 2.10 calculation of round off value upto three digits

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.10
//calculation of round off value upto three digits.

//given data
a=15462
b=14.745
c=14.750
d=14.650*10^12

//calculation
//since round off upto three digit is required, we
   have to sort the numerics with the number of
   significant figures i.e. 3
na=15500
nb=14.7
nc=14.8
nd=14.6*10^12

printf('the value of %f rounded upto three
   significant digits is %d',a,na);
printf('\nthe value of %f rounded upto three
   significant digits is %3.2f',b,nb);
```

```
printf('\nthe value of %f rounded upto three
   significant digits is %3.2f',c,nc);
printf('\nthe value of %3.4e rounded upto three
   significant digits is %3.2e',d,nd);
```

Scilab code Exa 2.10w calculation of value of the given scalar

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.10w
//calculation of value of the given scalar

//given data
ax=2; //xcomponent of A vector
ay=-3; //ycomponent of A vector
az=7; //zcomponent of A vector

bx=1; //xcomponent of B vector
by=0; //ycomponent of B vector
bz=2; //zcomponent of B vector

cx=1; //xcomponent of C vector
cy=0; //ycomponent of C vector
cz=2; //zcomponent of C vector

//calculation
//D=B*C
dx=(by*cz)-(cy*bz);
dy=-((bx*cz)-(cx*bz));
dz=(bx*cy)-(cx*by);

//R=A.(B*C)
R=(ax*dx)+(ay*dy)+(az*dz);

```

```
disp(R,'value of the given scalar is');
```

Scilab code Exa 2.11 calculation of value

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.11
//calculation of value

//given data
x=25.2;
y=1374;
z=33.3;

//calculation
temp=(x*y)/z
//since x,z has three significant figures and y has four significant figures
//we have to sort the answer with the minimum number of significant figures i.e. 3
//results into temp=1039.7838  we need to consider only 3 significant figures, hence

ntemp=1040

printf('value is %f,considering only 2 significant figures value is %d',temp,ntemp);
```

Scilab code Exa 2.11w calculation of change in volume of sphere as radius is changed

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 2.11w
//calculation of change in volume of sphere as
    radius is changed

//given data
R=20; //initial radius(in cm) of sphere
Rdash=20.1; //final radius(in cm) of sphere
function v=f(R)
    v=(4*%pi*R^3)/3;
endfunction

//calculation
function v=f(R)
    v=(4*%pi*R^3)/3;
endfunction

deltaR=Rdash-R;
deltav=(derivative(f,R))*deltaR

disp(deltav,'the change in volume(in cm cube) of
    sphere is')
```

Scilab code Exa 2.12 calculation of value

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.18w
//calculation of value

//given data
x=24.36;
y=0.0623;
```

```
z=256.2;

//calculation

//since after point the value of z is in one digit , thus consider only one digit after point.
//the other values can be thus written as
x=24.4;
y=.1;
z=256.2;

temp=x+y+z
printf('the value is %3.1f',temp);
```

Scilab code Exa 2.13 calculation of average focal length of concave mirror considering uncertainity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.13
//calculation of average focal length of concave mirror considering uncertainity

//given data
fi=[25.4 25.2 25.6 25.1 25.3 25.2 25.5 25.4 25.3 25.7]; //focal length(in cm)
N=length(fi);

//calculation
fbar=mean(fi) //average of fi
fnew=fi-fbar;
sfnew=sum(fnew.*fnew)
sigma=sqrt(sfnew/N) //uncertainity(in cm) in focal length
```

```

printf("the focal length of the given concave mirror (in cm) is %f or %f",fbar+sigma,fbar-sigma);
```

Scilab code Exa 2.13w calculation of maximum and minimum value of a given function

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.13w
//calculation of maximum and minimum value of a given function

//given data
function y=f(x)
    y=x+(1/x);
endfunction

//calculation
//dy/dx=1-(1/x^2)=0 for maximum or minimum
//x=1 or -1
//at x=0 y=infinite is maximum value
//minimum value of y at x=1
ymin=f(1);

disp(ymin,'maximum value of given function is infinite and minimum value is ')
```

Scilab code Exa 2.14w calculation of the area under curve

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 2.14w
//calculation of the area under curve

//given data
function y=f(x)
    y=x*x;
endfunction

//calculation
A=integrate('f','x',0,6)

disp(A,'the area under curve is')
```

Scilab code Exa 2.18w calculation of value

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 2.18w
//calculation of value

//given data
x=21.6003;
y=234;
z=2732.10;
a=13;

//calculation
//since a has least significant figures that is 2,
    we have to sort the other numerics with the same
    number of significant figures i.e. 2
x=22;
y=234;
z=2732;
```

```
a=13;
temp=(x+y+z)*13
//results into temp=38844. Again we need to consider
    only 2 significant figures , hence
ntemp=39000

printf('value is %d,considering only 2 significant
   figures value is %d',temp,ntemp);
```

Chapter 3

Rest and Motion Kinematics

Scilab code Exa 3.1 calculation of distance and displacement

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.1
//calculation of distance and displacement

//given data
r=40; //radius(in m) of the circle

//calculation
dist=%pi*r; //distance travelled(in m)
displ=2*r; //displacement(in m)

disp(dist,'distance travelled(in m) by the person is
   ');
disp(displ,'displacement(in m) of the person from
   initial to final point is');
```

Scilab code Exa 3.1w calculation of average speed of the walk

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.1w
//calculation of average speed of the walk

//given data
v1=6//speed(in km/h) of the man
v2=8//speed(in km/h) of the man
d1=1//distance(in km) travelled at v1 speed
d2=1//distance(in km) travelled at v2 speed
d=2//given distance(in km)

//calculation
t=(v1/d1)+(v2/d2);//total time(in s) taken
vavg=d/t;//formula for average velocity

disp(vavg,'the average velocity(in km/h) of the man
    is');
```

Scilab code Exa 3.2 calculation of average speed and instantaneous speed

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.2
//calculation of average speed and instantaneous
    speed

//given data
function s=f(t)
    s=2.5*t^2;
endfunction
t=5; //time (in s)

```

```
//calculation
vav=f(t)/t; //average speed(in m/s)
vinst=derivative(f,t); //instantaneous speed(in m/s)

disp(vav,'the average speed(in m/s) of the particle
    is');
disp(vinst,'the instantaneous speed(in m/s) of the
    particle is');
```

Scilab code Exa 3.2w calculation of average speed and average velocity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.2w
//calculation of average speed and average velocity

//given data
w=40//length(in ft)of the wall
t=50//time(in min) taken
rnd=10//number of rounds taken

//calculation
dist=2*w*rnd;
avgspeed=dist/t;
avgvelocity=0//average velocity(in ft/min) since
    displacement=0  as he is at the same door from
    where he has started

printf('the average speed of the teacher is %3.2f ft
    /min and the average velocity is %3.2f ft/min',
    avgspeed,avgvelocity);
```

Scilab code Exa 3.3 calculation of distance from speed versus time graph

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.3
//calculation of distance from speed versus time graph

//given data
base=3; //time(in s) representing the base of graph(triangle)
height=6; //speed(in m/s) representing the height of the graph(triangle)
//calculation
dist=(1/2)*base*height; //distance travelled is the area of the graph(triangle)

disp(dist,'the distance(in m) travelled by the particle is');
```

Scilab code Exa 3.3w calculation of average velocity and average acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.3w
//calculation of average velocity and average acceleration

//given data
A=1//given value of constant A
B=4//given value of constant B
C=-2//given value of constant C
```

```
D=5//given value of constant D
t=4//time(in s)
t1=0//initial time(in s) for calculation of average velocity and average acceleration
t2=4//final time(in s) for calculation of average velocity and average acceleration

function x=f(t)
    x=(A*(t^3))+(B*(t^2))+(C*t)+D
endfunction

function a=f1(t)
        a=(6*A*t)+(2*B)
endfunction

//calculation
v=derivative(f,t)//formula of velocity
na=f1(t)//formula of acceleration

x1=f(t1);//formula of position of the particle at t1 time
x2=f(t2);//formula of position of the particle at t2 time
vavg=(x2-x1)/(t2-t1);//formula of average velocity

v1=derivative(f,t1);//formula of velocity of the particle at t1 time
v2=derivative(f,t2);//formula of velocity of the particle at t2 time
aavg=(v2-v1)/(t2-t1);//formula of average acceleration

printf('\nthe velocity of particle at t=4 s is %3.2f m/s',v);
printf('\nthe acceleration of particle at t=4 s is %3.2f m/s^2',na)
printf('\nthe average velocity of the particle between t=0 s and t=4 s is %3.2f m/s',vavg);
```

```
printf('\nthe average acceleration of the particle
   between t=0 s and t=4 s is %3.2f m/s^2',aavg);
```

Scilab code Exa 3.4 calculation of average velocity of the tip of minute hand in a table clock

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.4
//calculation of average velocity of the tip of
   minute hand in a table clock

//given data
R=4; //length(in cm) of the minute hand = radius(in
   cm) of the circle representing the clock
t1=1800; //time(in second) elapsed between 6.00 a.m
   and 6.30 a.m       30*60
t2=45000; //time(in second) elapsed between 6.00 a.m
    and 6.30 p.m       (12*60*60) + (30*60)

//calculation
vav1=(2*R)/t1; //average velocity(in cm/s) in first
   case
vav2=(2*R)/t2; //average velocity(in cm/s) in second
    case

disp(vav1,'average velocity(in cm/s) of the tip of
   minute hand in time elapsed between 6.00 a.m and
   6.30 a.m is');
disp(vav2,'average velocity(in cm/s) of the tip of
   minute hand in time elapsed between 6.00 a.m and
   6.30 p.m is');
```

Scilab code Exa 3.4w calculation of distance travelled displacement and acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.4w
//calculation of distance travelled,displacement and acceleration

//given data
//graph of velocity(in m/s) versus time(in s)

//calculation
d1=(2*10)/2;//distance(in m) travelled during t=0 s to t=2 s = area of OAB
d2=(2*10)/2;//distance(in m) travelled during t=2 s to t=4 s = area of BCD
d=d1+d2;//distance(in m) travelled during t=0 s to t=4 s
dis=d1+(-d2);//displacement(in m) during t=0 s to t=4 s
a1=(10-0)/(1-0);//acceleration(in m/s^2) at t=1/2 s = slope of OA
a2=(-10-0)/(3-2);//acceleration(in m/s^2) at t=2 s = slope of BC

disp(d1,'distance(in m) travelled during t=0 s to t=2 s is');
disp(d2,'distance(in m) travelled during t=2 s to t=4 s is');
disp(d,'distance(in m) travelled during t=0 s to t=4 s');
disp(dis,'displacement(in m) during t=0 s to t=4 s')
```

```
;
disp(a1,'acceleration(in m/s^2) at t=1/2 s');
disp(a2,'acceleration(in m/s^2) at t=2 s');
```

Scilab code Exa 3.5 calculation of distance travelled in given time time taken to reach a particular velocity and distance covered to reach particular velocity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.5
//calculation of distance travelled in given time, time taken to reach a particular velocity and distance covered to reach particular velocity

//given data
u=2.5; //initial velocity(in m/s) of the particle
t=2; //time(in s) for which the particle has travelled
v=7.5; //final velocity(in m/s) of the particle
a=.5; //acceleration(in m/s^2) of the particle

//calculation
x=(u*t)+((1/2)*a*t*t); //Equation of motion with constant acceleration
t1=(v-u)/a; //Equation of motion with constant acceleration
x1=((v*v)-(u*u))/(2*a); //Equation of motion with constant acceleration

disp(x,'distance(in m) travelled by the particle in the first two seconds is');
disp(t1,'time(in s) taken by particle to reach 7.5 m/s velocity is');
```

```
disp(x1,'distance(in m) covered by particle to reach
    7.5 m/s velocity is');
```

Scilab code Exa 3.5w calculation of acceleration and distance travelled

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.5w
//calculation of acceleration and distance travelled

//given data
v1=100//speed1(in m/s)
v2=150//speed2(in m/s)
t=1//change in time (in s)

//calculation
a=(v2-v1)/t;//formula of acceleration
x=((v2*v2)-(v1*v1))/(2*a);//distance travelled in (t
   +1)th second

printf('acceleration of the particle is %3.2f m/s^2'
   ,a)
printf('\ndistance travelled in (t+1)th second is %3
   .2f m',x)
```

Scilab code Exa 3.6 calculation of displacement of particle in last 1 second

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.6
```

```
//calculation of displacement of particle in last 1 second

//given data
u=5; //initial velocity(in m/s) of the particle
a=2; //constant acceleration(in m/s^2) of the particle
t=10; //time(in s)

//calculation
//s = u*t+((1/2)*a*t^2)....equation of motion
//sdash = u*(t-1)+((1/2)*a*(t-1)^2)
//st = s-sdash =u+((a/2)*(2*t-1));

st=u+((a/2)*(2*t-1)); //formula of displacement in last one second
disp(st,'displacement(in m) of particle in last 1 second');
```

Scilab code Exa 3.6w calculation of acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.6w
//calculation of acceleration

//given data
u=0//initial velocity(in m/s)
v=2.2//final velocity(in m/s)
d=.24//distance(in m) travelled

//calculation
a=((v*v)-(u*u))/(2*d);//formula of acceleration

```

```
printf('the acceleration of the stone is %3.3f m/s^2
    ',a)
```

Scilab code Exa 3.7 calculation of maximum height reached by the ball

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.7
//calculation of maximum height reached by the ball

//given data
u=4; //initial velocity(in m/s) of the ball
a=-10; //acceleration(in m/s^2) of the ball

//calculation
y=-((u*u)/(2*a)); //formula for vertical height(in m
    )

disp(y,'maximum height(in m) reached by the ball is'
    );
```

Scilab code Exa 3.8 calculation of velocity and position of the particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.8
//calculation of velocity and position of the
    particle

//given data
a=1.5; //acceleration(in m/s^2) of the particle
```

```
theta=37; //angle(in degree) made by particle with X axis
ux=8; //x component of initial velocity(in m/s) of the particle
uy=0; //y component of initial velocity(in m/s) of the particle
t=4; //time(in s)

//calculation
ax=a*cosd(theta);
ay=a*sind(theta);

vx=ux+(ax*t); //formula of x component of final velocity
vy=uy+(ay*t); //formula of y component of final velocity
v=sqrt((vx*vx)+(vy*vy));
thetav=atand(vy/vx);

x=(ux*t)+((ax*t*t)/2); //formula for x coordinate of particle at time t
y=(uy*t)+((ay*t*t)/2); //formula for y coordinate of particle at time t

printf('the velocity of the particle at t=4 s is %f m/s and angle made with X axis is %f degree',v,thetav)
printf('the particle is at(%f,%f)m at  time t=4 s',x,y)
```

Scilab code Exa 3.8w calculation of total distance and number of trips

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 3.8w
//calculation of total distance and number of trips

//given data
dcar=20//distance(in km) travelled by the car
vcar=40//speed(in km/h) of the car
vfly=100//speed(in km/h) of the fly

//calculation
tcar=dcar/vcar;//time(in h) taken by the car to cover given distance
tfly=tcar;
dfly=tfly*vfly;//distance(in m) travelled by the fly
//number of trips made by fly can be infinite

printf('total distance travelled by the fly is %3.2f km and number of trips made by fly can be infinite',dfly);
```

Scilab code Exa 3.9 calculation of horizontal range of the projectile

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.9
//calculation of horizontal range of the projectile

//given data
u=12//initial velocity(in m/s) of the projectile
theta=45//angle(in degree) made by the projectile with X axis
g=10//gravitational acceleration(in m/s^2)

//calculation
h=(u*u*sind(2*theta))/g;//formula for horizontal
```

```
    range of a projectile

printf('the ball hits the field at %f m from the
    point of projection',h);
```

Scilab code Exa 3.9w drawing graph of x versus t v versus t and a versus t

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.9w
//drawing graph of x versus t, v versus t and a
    versus t

//given data
h=19.6//height(in m) from where the ball is dropped
//evaluating value for equation    x=(u*t)+((1/2)*a*
    t^2)

//calculation
t=[0 1 2 2 3 4]
x=[0 4.9 19.6 19.6 4.9 0]//values of x(in m)
    obtained on evaluating equation    x=(u*t)+((1/2)
    *a*t^2) along with direction of motion
v=[0 9.8 19.6 -19.6 -9.8 0]//values of v(in m)
    obtained on evaluating equation    v=u+(a*t)
    along with direction of motion
a=9.8//constant acceleration(m/s^2)

subplot(221);
plot(t,x);
xlabel('time(in s)')
ylabel('distance(in m)')

```

```
subplot(222);
plot(t,v);
xlabel('time(in s)')
ylabel('velocity(in m/s)')

subplot(223);
plot(t,a);
xlabel('time(in s)')
ylabel('acceleration (in m/s^2)')
```

Scilab code Exa 3.10 calculation of velocity of the swimmer with respect to ground

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.10
//calculation of velocity of the swimmer with
    respect to ground

//given data
vsr=4//velocity(in km/h) of the swimmer with respect
     to water
vrg=3//velocity(in km/h) of the river water with
    respect to ground

//calculation
vsg=sqrt((vsr*vsr)+(vrg*vrg));//formula for relative
     velocity    vsg = vsr + vrg
theta=atand(4/3);

printf('the velocity of the swimmer with respect to
    ground is %f km/h and angle made by him with X
    axis is %f degree',vsg,theta);
```

Scilab code Exa 3.10w calculation of height of balloon when stone reaches ground

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.10w
//calculation of height of balloon when stone
   reaches ground

//given data
x=-50//height(in m) of the ballon when the stone was
    dropped
u=5//velocity(in m/s) of the ballon
a=-10//acceleration(in m/s^2) of the ballon

//calculation
//from x=(u*t)+((1/2)*a*t*t) we have −5*t^2 + 5*t +
   50 = 0
a=-5//coefficient of t^2
b=5//coefficient of t
c=50//constant

t1=(-b+sqrt((b*b)-(4*a*c)))/(2*a)//value of t
t2=(-b-sqrt((b*b)-(4*a*c)))/(2*a)//value of t

if(t1>0)
    t=t1;
end

if(t2>0)
    t=t2;
end

```

```
if(t1>0 & t2>0)
    tn1=t1;
    tn2=t2;
end

tballoon=t;//during this time baloon has uniformly moved upwards
dballoon=u*t;
dtotal=dballoon+(-x);

printf('height of the ballon when the stone reaches ground is %3.2f m',dtotal);
```

Scilab code Exa 3.11 calculation of velocity of the raindrops with respect to the man

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.11
//calculation of velocity of the raindrops with respect to the man

//given data
vmanstreet=3//velocity(in km/h) of man with respect to the street
vrainstreet=4//velocity(in km/h) of rain with respect to the street

//calculation
vrainman=sqrt((vrainstreet*vrainstreet)+(vmanstreet*vmanstreet));//velocity(in km/h) of rain with respect to the man
theta=atand(vmanstreet/vrainstreet);//angle(in degree) made by rain drops with Y axis
```

```

printf('velocity of the raindrops with respect to the man is %3.2f km/h and angle made by rain drops with Y axis is %3.3f degree',vrainman,theta)
```

Scilab code Exa 3.11w calculation of time of flight horizontal range and vertical range

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.11w
//calculation of time of flight,horizontal range and vertical range

//given data
u=20//initial velocity(in m/s) of the football
theta=45//angle(in degree) made by the football with ground
g=10//gravitational acceleration(in m/s^2)

//calculation
ux=u*cosd(theta);
uy=u*sind(theta);

t=(2*uy)/g;// from equation y=(uy*t)+((1/2)*g*t*t)......taking y=0
H=((uy*uy)/(2*g));//from equation (vy*vy)=(uy*uy)-(2*g*y)       taking vy=0
x=ux*t;//horizontal distance travelled at ux velocity

printf('the time taken by the ball to strike the ground is %3.2f s',t);
```

```
printf('\nthe maximum height reached by the ball is
    %3.2f m',H);
printf('\nthe horizontal distance travelled by the
    ball before reaching the ground is %3.2f m',x);
```

Scilab code Exa 3.16w calculation of angle of the swim and time to cross the river

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.16w
//calculation of angle of the swim and time to cross
     the river

//given data
vrg=2//velocity(in km/h) of the river with respect
    to ground
vmr=3////velocity(in km/h) of the man with respect
    to river
d=.5//width(in km) of the river

//calculation
theta=asind(vrg/vmr);//from equation of relative
    velocity   vmg=vmr+vrg...taking components along
    X axis
vmg=vmr*cosd(theta);//taking component along Y axis
time=d/vmg;

printf('swimmer should try to swim,making an angle
    of %3.2f degree with Y axis',theta);
printf('\ntime taken by the swimmer to cross the
    river is %3.2f h',time);
```

Scilab code Exa 3.17w calculation of time taken and position of the arrival on opposite bank

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.17w
//calculation of time taken and position of the
   arrival on opposite bank

//given data
dyaxis=.5//displacement(in km) along Y axis
vrg=2//velocity(in km/h) of the river with respect
   to ground
vmr=3////velocity(in km/h) of the man with respect
   to river
theta1=30//angle(in degree) of vmr with Y axis
theta2=90//angle(in degree) of vrg with Y axis

//calculation
vyaxis=(vmr*cosd(theta1))+(vrg*cosd(theta2));//
   velocity along Y axis i.e taking y component in
   equation  vmg=vmr+vrg
t=dyaxis/vyaxis;
vxaxis=(-vmr*sind(theta1))+(vrg*sind(theta2));//
   velocity along X axis i.e taking x component in
   equation  vmg=vmr+vrg
dxaxis=vxaxis*t;

printf('time taken by the swimmer to cross the river
    is %3.2f hour',t);
printf('\ndisplacement of the swimmer along X axis
   is %3.4f km',dxaxis);
```

Scilab code Exa 3.18w calculation of speed of raindrops with respect to road and the moving man

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 3.18w
//calculation of speed of raindrops with respect to
   road and the moving man

//given data
vmg=10//velocity(in km/h) of the man with respect to
    the ground
theta=30//angle(in degree) made by vrg with Y axis

//calculation
vrg=vmg/sind(theta);// from equation of relative
   velocity  vrg=vrm+vmg...taking horizontal
   components
vrm=vrg*cosd(theta);// from equation of relative
   velocity  vrg=vrm+vmg...taking vertical
   components

printf('the speed of raindrops with respect to the
   ground is %3.2f km/h and with respect to the  man
    is %3.2f km/h',vrg,vrm);
```

Scilab code Exa 3.19w calculation of speed and direction of rain with respect to the road

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 3.19w
//calculation of speed and direction of rain with respect to the road

//given data
vmanroad=8//velocity(in km/h) of the man with respect to the road

//calculation
//from equation of relative velocity    vrainroad = vrainman + vmanroad
//taking horizontal components          vrainroad*sind(aplha)=8                 1
//taking components along line OA       vrainroad*sind(30+alpha)=12*cosd(30)     2
//from    1    and    2

alpha=acotd(sqrt(3)/2);
vrainroad=vmanroad/sind(alpha);//from equation 2

printf('the speed of the rain with respect to the road is %3.2f km/h and makes angle of %3.2f degree with Y axis',vrainroad,alpha);
```

Chapter 4

The Forces

Scilab code Exa 4.1 calculation of coulomb force

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 4.1
//calculation of coulomb force

//given data
np=26//number of protops in an iron atom
na=6*10^26//number of atome in 58 kg iron
mi=58//mass(in kg) of iron
e=1.6*10^(-19)//charge(in coulomb) on an electron
perdiff=1//percentage of charge of electron less
    than that of proton
r=1//separation(in m) between the two blocks

//calculation
poschrg=(na*np*e*perdiff)/(mi*100)
fc=(9*10^9*poschrg*poschrg)/(r*r)

disp(fc,'the coulomb force(in newton) between the
    two blocks is newton')
```

Scilab code Exa 4.3w calculation of ratio of the electrical force to the gravitational force between two electrons

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 4.3w
//calculation of ratio of the electrical force to
   the gravitational force between two electrons

//given data
me=9.1*10^-31//mass(in kg)of an electron
e=1.6*10^-19//charge(in coulomb)of an electron
k=9*10^9//value of ratio 1/(4*%pi*epsilonzero)   (in
   N m^2/C^2)
G=6.67*10^-11//value of universal gravitational
   constant   (in N m^2/kg^2)

//calculation
ratio=(k*e*e)/(G*me*me)//ratio = electric force /
   gravitational force

disp(ratio,'the ratio of electric to gravitational
   force between two electrons is')
```

Chapter 5

Newton s Laws of Motion

Scilab code Exa 5.1 calculation of force exerted by the string on a particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
  clc;clear;
//example 5.1
//calculation of force exerted by the string on a
    particle

//given data
m=.5//mass(in kg) of the particle
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
T=m*g//tension in the string is equal to the
    downward force exerted by earth

printf('the force extered by the string on particle
     in vertically upward direction is %3.2f N',T);
```

Scilab code Exa 5.3w calculation of the force exerted by the tree limb on the bullet

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 5.3w
//calculation of the force exerted by the tree limb on the bullet

//given data
u=250//initial velocity(in m/s) of the bullet
v=0//final velocity(in m/s) of the bullet
x=.05//penetration(in m) by the bullet in the tree limb
m=.01//mass of bullet(in kg)

//calculation
a=((u*u)-(v*v))/(2*x)//formula of horizontal acceleration in case of uniform linear motion
F=m*a;

printf('the force exerted by the tree limb on the bullet is %3.2f N',F)
```

Scilab code Exa 5.4w calculation of the position of a particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 5.4w
//calculation of the position of a particle

//given data
m=.01//mass(in kg) of the particle
```

```
Fx=10//component of force(in N) along X axis
Fy=5//component of force(in N) along Y axis
ux=0//x component of initial velocity(in m/s) of the particle
uy=0//y component initial velocity(in m/s) of the paticle
t=5//time(in s) at which position is to be determined

//calculation
ax=Fx/m;
x=(ux*t)+((1/2)*ax*t*t);//formula of horizontal position in case of uniform linear motion
ay=Fy/m;
y=(uy*t)+((1/2)*ay*t*t);//formula of vertical position in case of uniform linear motion

printf('at t=5 s position of the particle is (i%3.2f + j%3.2f)m',x,y)
```

Scilab code Exa 5.7w calculation of acceleration with which ring starts moving if released from rest at an angle theta

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 5.7w
//calculation of acceleration with which ring starts moving if released from rest at an angle theta

//given data
//m=mass of the ring
theta=30//angle(in degree)of the release
m=1//assume for obtaiming the solution
M=2*m //mass of the block
```

```
g=9.8//gravitational acceleration(in m/s^2) of the earth

//calculation
//M*g-T=M*a*cosd(theta)........equation of motion of the block...(1)
//T*cosd(theta)=m*a............equation of motion of the ring....(2)
//solving above equations we get
a=(M*g*cosd(theta))/(m+M*(cosd(theta)*cosd(theta)))

printf('the acceleration with which ring starts moving if released from rest at an angle theta is %3.2f m/s^2',a)
```

Scilab code Exa 5.8w calculation of the maximum acceleration of the man for safe climbing

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 5.8w
//calculation of the maximum acceleration of the man for safe climbing

//given data
m=60//mass(in kg) of the man
theta=30//angle(in degree) made by the rope with ground
fgmax=360//maximum force(in N0 that can be applied to the wooden clamp
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
```

```
T=fgmax/sind(theta)//since t*sin(theta)=upward force
a=(T-(m*g))/m//from equation of motion

printf('the maximum acceleration of the man for safe
    climbing is %3.2f m/s^2',a)
```

Chapter 6

Friction

Scilab code Exa 6.1 calculation of the angle made by the contact force with the vertical and the magnitude of contact force

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.1
//calculation of the angle made by the contact force
    with the vertical and the magnitude of contact
   force

//given data
M=.4//mass(in kg) of the body
f=3//frictional force(in N)
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
N=M*g//formula of normal force
theta=atand(f/N)//angle made by the contact force
   with the vertical
F=sqrt((N*N)+(f*f))

```

```
printf('the angle made by the contact force with the
    vertical is %3.2f degree \n the magnitude of
   contact force is %3.2f N',theta,F)
```

Scilab code Exa 6.1w calculation of the maximum angle to prevent slipping

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.1w
//calculation of the maximum angle to prevent
   slipping

//given data
mus=.3//coefficient of static friction

//calculation
thetamax=atand(mus)

printf('the maximum angle to prevent slipping is %3
   .2f',thetamax)
```

Scilab code Exa 6.2 calculation of the force of friction exerted by the horizontal surface on the box

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.2
//calculation of the force of friction exerted by
   the horizontal surface on the box

```

```
//given data
M=20//mass(in kg) of the box
muk=.25//coefficient of kinetic friction
g=9.8//gravitational acceleration(in m/s^2) of the earth

//calculation
fk=muk*M*g//formula of kinetic friction

printf('the force of friction exerted by the horizontal surface on the box,in opposite direction to the pull is %3.2f N',fk)
```

Scilab code Exa 6.2w calculation of frictional force and minimum value of coefficient of static friction

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.2w
//calculation of frictional force and minimum value of coefficient of static friction

//given data
m=4//mass(in kg) of the block
f=20//frictional force(in N)=horizontal force(in N)
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
N=m*g//normal force
musmin=f/N

printf('the frictional force on the block,in opposite direction to the applied force is %3.2f
```

```
N',f)
printf('\nthe coefficient of static friction between
    the block and the table is greater than or equal
    to %3.2f',musmin)
```

Scilab code Exa 6.3 calculation of the force of friction exerted by the horse and condition of boy for sliding back

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.3
//calculation of the force of friction exerted by
   the horse and condition of boy for sliding back

//given data
M=30//mass(in kg) of the boy
a=2//average acceleration(in m/s^2) of the horse
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
fs=M*a//Newton's second law
musmax=fs/(M*g)//equation of static friction

printf('the force of friction exerted by the horse
   on the boy is %3.2f N',fs)
printf('\nfor the boy sliding back during
   acceleration, the value of coefficient of static
   friction is less than %3.2f ',musmax)
```

Scilab code Exa 6.3w calculation of the maximum value of mass of the block

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.3w
//calculation of the maximum value of mass of the
    block

//given data
mus=.2//coefficient of static friction between the
    block and the table
M=2//mass(in kg) of one block
g=10//gravitational acceleration(in m/s^2) of the
    earth

//calculation
N=M*g//normal force
//T=m*g              tension in the string
                                (1)
//fs=mus*N           frictional force
                                        (2)
//f=T                from equlibrium equation of 2 kg
    block       (3)
//from above equations,we get
m=(mus*N)/g

printf('the maximum value of mass of the block is %3
    .2f kg',m)
```

Scilab code Exa 6.4 calculation of coefficient of static friction and kinetic friction between the block and the plank

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.4
```

```
//calculation of coefficient of static friction and kinetic friction between the block and the plank

//given data
theta1=18//angle of plank(in degree) with horizontal when block starts slipping
theta2=15//angle of plank(in degree) with horizontal when block slips with uniform speed

//calculation
mus=tand(theta1)//formula of coefficient of static friction
muk=tand(theta2)//formula of coefficient of kinetic friction

printf('the coefficient of static friction between the block and the plank is tan(%d)=%3.2f',theta1,mus)
printf('\n the coefficient of kinetic friction between the block and the plank is tan(%d)=%3.2f',theta2,muk)
```

Scilab code Exa 6.5w calculation of the coefficient of kinetic friction

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.5w
//calculation of the coefficient of kinetic friction

//given data
theta=30//angle(in degree)f the incline
g=10//gravitational acceleration(in m/s^2) of the earth

```

```
//calculation
a=g/4//acceleration(in m/s^2) of the block.....given
//f=m*g/4................taking parallel components to the incline
//N=m*g*cosd(theta)......taking vertical components to the incline
//from above equations,we get
muk=1/(4*cosd(theta))//     muk=f/N   equation of static friction

printf('the coefficient of kinetic friction is %3.2f',muk)
```

Scilab code Exa 6.6w calculation of the values of coefficient of static and kinetic friction

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.6w
//calculation of the values of coefficient of static and kinetic friction

//given data
M=2.5//mass(in kg) of the block
F=15//horizontal force(in N)
g=10//gravitational acceleration(in m/s^2) of the earth
x=10//displacement(in m) of the block
t=5//time(in s) required by the block

//calculation
mus=F/(M*g)
a=(2*x)/(t*t)//acceleration of the block from equation of uniform linear motion
```

```
//F-muk*M*g=M*a.....newton's second law
muk=(F-(M*a))/(M*g)

printf('the coefficient of static friction between the block and the surface is %3.2f',mus)
printf('\n the coefficient of kinetic friction between the block and the surface is %3.2f',muk)
```

Scilab code Exa 6.10w calculation of mimimum and maximum values of mass and the acceleration if given a gentle push

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 6.10w
//calculation of mimimum and maximum values of m(mass) and the acceleration if given a gentle push

//given data
mus=.28//the value of coefficient of static friction between the block and the surface
muk=.25//the value of coefficient of kinetic friction between the block and the surface
M=2//mass(in kg) of one block
g=9.8//gravitational acceleration(in m/s^2) of the earth

//calculation
//T=(M*g*(1-mus))/sqrt(2)................taking components along incline for block1......(1)
//T=(M*g*(1+mus))/sqrt(2)................taking components along incline for block2......(2)
//from above equations,we get
m1=((1-mus)*M)/(1+mus)//minimum value of m ..........................................(3)
```

```
m2=((1+mus)*M)/(1-mus)//maximum value of m obtained by taking reverse direction of friction in above equations

//(M*g/sqrt(2)) - T = M*a.........newton's second law for M block........................(4)
//T - (m*g/sqrt(2)) = m*a.........newton's second law for m block........................(5)
//adding equations (4) and (5)
//((M*g*(1-muk))/sqrt(2)) - ((m*g*(1+muk))/sqrt(2)) = (M+m)*a
a=(((M*(1-muk))-(m1*(1+muk)))*g)/(sqrt(2)*(M+m1))//calculating acceleration for minimum value of m if gently pushed......given

printf('the minimum value of m for which the system remains at rest is %3.2f kg',m1)
printf('\nthe maximum value of m for which the system remains at rest is %3.2f kg',m2)
printf('\nthe acceleration of either block for minimum value of m and if gently pushed up the incline is %3.2f m/s^2',a)
```

Chapter 7

Circular Motion

Scilab code Exa 7.1 calculation of the angular velocity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.1
//calculation of the angular velocity

//given data
v=10//linear speed(in m/s)
r=20*10^-2//radius(in cm) of the circle

//calculation
w=v/r//formula of angular velocity

printf('the angular velocity is %d rad/s',w)
```

Scilab code Exa 7.1w calculation of the maximum speed the car can take on the turn without skidding

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.1w
//calculation of the maximum speed the car can take on the turn without skidding

//given data
R=45//radius(in m) of the turn
mus=2.0//coefficient of static friction between the tyre and the road
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
//considering forces in vertical and horizontal directions an dpplying Newton's law we get
//fs = M*v*v...........................(1)
//by equation of limiting friction,we get
//fs = mus*N = mus*M*g...............(2)
//from above equations we get
v=sqrt(mus*g*R)

printf('the maximum speed the car can take on the turn without skidding is %d m/s or %3.1f km/hr',v,(v*10^-3*60*60))
```

Scilab code Exa 7.2 calculation of the angular acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.2
//calculation of the angular acceleration

```

```
//given data
v1=5//initial speed(in m/s)
v2=6//final speed(in m/s)
dt=2//change in time(in s)
r=20*10^-2//radius(in cm) of the circle

//calculation
at=(v2-v1)/dt//formula of tangential acceleration
alpha=at/r//formula of angular acceleration

printf('the angular accleration is %3.1f rad/s^2',
    alpha)
```

Scilab code Exa 7.2w calculation of the value of angle of banking

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.2w
//calculation of the value of angle of banking

//given data
r=600//radius(in m) of the track
v=180*10^3/(60*60)//speed(in m/s) of the car
g=10//gravitational acceleration(in m/s^2) of the
    earth

//calculation
//for vertical direction
//N*cosd(theta) = M*g.........................(1)
//for horizontal direction
//N*sind(theta) = M*v*v/r.....................(2)
//from above equations,we get
theta=atand((v*v)/(r*g))

```

```
printf('the value of angle of banking is %3.2f
    degree',theta)
```

Scilab code Exa 7.3 calculation of the magnitude of linear acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.3
//calculation of the magnitude of linear
    acceleration

//given data
r=10*10^-2//radius(in cm)
t=4//time(in s) taken

//calculation
d=2*%pi*r//distance covered
v=d/t//linear speed
a=(v*v)/r

printf('the linear acceleration is %3.2f m/s^2',a)
```

Scilab code Exa 7.4 calculation of the value of radial and tangential acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.4
//calculation of the value of radial and tangential
    acceleration

```

```
//given data
t=3//time(in s)
r=20*10^-2//radius(in cm) of the circle

function v1=f(t1)
    v1=2*t1
endfunction

//calculation
v=f(t)
ar=(v*v)/r//radial acceleration
at=derivative(f,t)//tangential acceleration

printf('the value of radial acceleration is %d m/s^2',ar)
printf('\nthe value of tangential acceleration is %d m/s^2',at)
```

Scilab code Exa 7.4w calculation of the value of elongation of the spring

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.4w
//calculation of the value of elongation of the spring

//given data
k=100//spring constant(N/m) of the given spring
l0=.5//natural length(in m) of the string
m=.5//mass(in kg) of the particle
w=2//angualr velocity(in rad/s) of the mass

//calculation
//from the equation of horizontal force
```

```
//k*l = m*v*v/r = m*w*w*r = =m*w*w*(10+l) ........................(1)
//from above equation we get
l=(m*w*w*10)/(k-(m*w*w))

printf('the value of elongation of the spring is %3.2f m or %3.1f cm',l,l*100)
```

Scilab code Exa 7.5 calculation of the normal contact force by the side wall of the groove

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.5
//calculation of the normal contact force by the side wall of the groove

//given data
r=25*10^-2//radius(in m) of the cirlce
m=.1//mass(in kg) of the block
t=2//time(in s) taken by the block

//calculation
v=2*%pi*r/t//speed of the block
a=(v*v)/r//acceleration of the block
N=m*a//newton's second law

printf('the normal contact force by the side wall of the groove is %3.2f N',N)
```

Scilab code Exa 7.6 calculation of the speed of vehicle on the turn

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.6
//calculation of the speed of vehicle on the turn

//given data
r=10//radius(in m) of the turn
theta=10//angle(in degree) of the bank
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
v=sqrt(r*g*tand(theta))//since tand(theta) = (v*v)/(
    r*g)

printf('for normal contact force providing the
    necessary centripetal force,the speed of vehicle
    on the turn is %3.2f m/s',v)
```

Scilab code Exa 7.7 calculation of the weight of the body if spring balance is shifted to the equator

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.7
//calculation of the weight of the body if spring
    balance is shifted to the equator

//given data
W=98//weight(in N) of the body at north pole
R=6400*10^3//radius(in m) of the earth
g=9.8//gravitational acceleration(in m/s^2) of the
    earth
```

```

//calculation
m=W/g//formula of weight
w=(2*%pi)/(24*60*60)//angular speed of the earth
We=W-(m*w*w*R)//  since We = W - (m*w*w*R)

printf('the weight of the body if spring balance is
    shifted to the equator is %3.2f N',We)
```

Scilab code Exa 7.7w calculation of the value of force exerted by the air on the plane

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.7w
//calculation of the value of force exerted by the
    air on the plane

//given data
v=900*10^3/(60*60)//speed(in m/s) of the fighter
    plane
r=2000//radius(in m)of the vertical circle
M=16000//mass(in kg)
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
//from Newton's second law
//F-M*g = M*v*v/r
//from above equation,we get
F=M*(g+(v*v/r))

printf(' the force exerted by the air,on the plane
    in upward direction is %3.2e N',F)
```

Scilab code Exa 7.8w calculation of the angular speed of rotation

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.8w
//calculation of the angular speed of rotation

//given data
L=20*10^-2//length(in m) of the rod = length(in m)of
    the string
theta=30//angle(in degree) made by the string with
   the vertical
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
//applying Newton's second law
//T*sind(theta) = m*w*w*L*(1+sind(theta))
   .............(1)
//applying Newton's first law in vertical direction
//T*cosd(theta) = m*g
   .................................(2)
//from above equations,we get
//tand(theta)=((w*w*L*(1+sind(theta)))/g)
   .............(3)
w=sqrt((g*tand(theta))/(L*(1+sind(theta))))

printf('the angular speed of rotation is %3.1f rad/s
   ',w)
```

Scilab code Exa 7.10w calculation of the minimum speed at which floor may be removed

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 7.10w
//calculation of the minimum speed at which floor
   may be removed

//given data
r=2//radius(in m) of the rotor
mus=0.2//coefficient of static friction between the
   wall and the person
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
//by applying Newton's second law for horizontal
   direction
//fs = m*g.................................(1)
//by limiting friction
//mus*N = m*g or mus*m*v*v/r = m*g...........(2)
//from above equations,we get
v=sqrt(r*g/mus)

printf('the minimum speed at which floor may be
   removed is %3.1f m/s',v)
```

Chapter 8

Work and Energy

Scilab code Exa 8.1 calculation of the work done by the spring force

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.1
//calculation of the work done by the spring force

//given data
k=50//spring constant(in N/m) of the spring
x=1*10^-2//compression(in m) from natural position

//calculation
W=(k*x*x)/2//work done in compressing a spring

printf('the work done by the spring force is %3.1e J
    ',W)
```

Scilab code Exa 8.1w calculation of the work done by the porter on the suitcase

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.1w
//calculation of the work done by the porter on the suitcase

//given data
m=20//mass(in kg) of suitcase
h=2//height(in m) above the platform
g=9.8//gravitational acceleration(in m/s^2) of the earth

//calculation
W=-m*g*h//work done by gravity
//the work done by the porter = negative of the work done by gravity

printf('the work done by the porter on the suitcase is %d J',-W)
```

Scilab code Exa 8.2 calculation of the work done by force of gravity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.2
//calculation of the work done by force of gravity

//given data
m=20*10^-3//mass(in kg) of the particle
u=10//speed(in m/s) of the particle
g=9.8//gravitational acceleration(in m/s^2) of the earth

```

```
//calculation
//from equation of motion.....(v*v)=(u*u)-(2*g*h)
    ......take v=0 we get
h=(u*u)/(2*g)
W=-m*g*h//law of conservation of energy

printf('the work done by force by gravity is %3.1f J
    ',W)
```

Scilab code Exa 8.2w calculation of the value of minimum horsepower of the motor to be used

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.2w
//calculation of the value of minimum horsepower of
    the motor to be used

//given data
m=500//mass(in kg) of the elevator
v=.20//velocity(in m/s) of the elevator
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
P=m*g*v//power = force*velocity

printf('the value of minimum horsepower of the motor
     to be used is %3.2f hp',P/746)
```

Scilab code Exa 8.3w calculation of the power delivered by the pulling force and average power

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.3w
//calculation of the power delivered by the pulling
   force and average power

//given data
m=2//mass(in kg)
theta=30//angle(in degree)
a=1//acceleration(in m/s^2) of the block
t=4//time(in s)
g=9.8//gravitational acceleration(in m/s^2) of the
   earth

//calculation
F=(m*g*sind(theta))+(m*a)//resolving the forces
   parallel to the incline
v=a*t
P=F*v//equation of power
d=a*t*t/2//from equation of motion
W=F*d
pavg=W/t//average power delivered

printf('the power delivered by the pulling force at
   t=4 s is %d W',P)
printf('\nthe average power delivered by the pulling
    force between t=0 s to t=4 s is %3.1f W',pavg)
```

Scilab code Exa 8.4w calculation of the work done by the given force

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.4w
```

```
//calculation of the work done by the given force

//given data
function F=f(x)
    F=(10+(.50*x))
endfunction
x1=0//initial position(in m) of the particle
x2=2//final position(in m) of the particle

//calculation
W=integrate('f','x',x1,x2)//work done

printf('the work done by the given force for the given displacement is %d J',W)
```

Scilab code Exa 8.5 calculation of the speed of the pendulum of bob when it makes an angle of 60 degree with the vertical

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.5
//calculation of the speed of the pendulum of bob when it makes an angle of 60 degree with the vertical

//given data
v0=3//speed(in m/s)of the bob in its lowest position
theta=60//angle(in degree)made by the pendulum with vertical
l=.5//length(in m) of the pendulum
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
```

```
//from the law of conservation of energy
//(m*v0*v0/2) - (m*v1*v1/2) = m*g*l*(1-cosd(theta))
v1=sqrt((v0*v0)-(2*g*l*(1-cosd(theta))))

printf('the speed of the pendulum of bob when it makes an angle of 60 degree with the vertical is %d m/s',v1)
```

Scilab code Exa 8.11w calculation of the speed of the particle at a given point

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.11w
//calculation of the speed of the particle at a given point

//given data
hA=1//height(in m) of point A
hB=.5//height(in m) of point B
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
//potential energies at point A and B are
//UA = M*g*hA
//UB = M*g*hB..............(1)
//principle of conservation of energy
//UA + KA = UB + KB........(2)
vB=sqrt(2*g*(hA-hB))

printf('the speed of the particle at a B point is %3.2f m/s',vB)
```

Scilab code Exa 8.12w calculation of the maximum compression of the spring

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 8.12w
//calculation of the maximum compression of the
    spring

//given data
k=400//spring constant(in N/m)
m=40*10^-3//mass(in kg)
h=4.9//height(in m)
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
//m*g*h = (k*x*x/2)
x=sqrt((2*m*g*h)/k)

printf('the maximum compression of the spring is %3
    .3f m or %3.1f cm',x,x*10^2)
```

Chapter 9

Centre of Mass Linear momentum Collision

Scilab code Exa 9.1w Locating the centre of maass of the system

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.1w
//Locating the centre of maass of the system

//given data
m1=.50//mass(in kg) at point1
m2=1//mass(in kg) at point2
m3=1.5//mass(in kg) at point3
x1=0//x coodinate (in cm) of point1
x2=4//x coodinate (in cm) of point2
x3=0//x coodinate (in cm) of point3
y1=0//y coodinate (in cm) of point1
y2=0//y coodinate (in cm) of point2
y3=3//y coodinate (in cm) of point3

//calculation
X=((m1*x1)+(m2*x2)+(m3*x3))/(m1+m2+m3)
```

```
Y=((m1*y1)+(m2*y2)+(m3*y3))/(m1+m2+m3)

printf('The centre of mass is %3.1f cm right and %3.1f cm left above the .5 kg paticle.',X,Y)
```

Scilab code Exa 9.4 calculation of the maximum compression of the string

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.4
//calculation of the maximum compression of the string

//given data
m=1//mass(in kg)
v=2//speed of the block(in m/s)
k=50//spring constant(in N/m)

//calculation
V=(m*v)/(m+m)//principle of conservation of linear momentum
ke1=(m*v*v/2)//initial kinetic energy
ke2=(m*V*V/2)+(m*V*V/2)//final kinetic energy
x=sqrt(2*(ke1-ke2)/k)//kinetic energy lost = elastic energy stored

printf('the maximum compression of the string is %3.1f m',x)
```

Scilab code Exa 9.5 calculation of the speed of combined mass

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 9.5
//calculation of the speed of combined mass

//given data
ma=50//mass(in kg) of cart A
mb=20//mass(in kg) of cart B
va=20//velocity(in km/hr) of cart A
vb=10//velocity(in km/hr) of cart B

//calculation
V=((ma*va)-(mb*vb))/(ma+mb)//principle of
    conservation of linear momentum

printf('the speed of combined mass after collision
    is %3.2f km/hr',V)
```

Scilab code Exa 9.6w calculation of the acceleration of the centre of mass

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.6w
//calculation of the acceleration of the centre of
    mass

//given data
M=2.5//mass(in kg) of the body
F1=6//force(in N) acting at point 1
F2=5//force(in N) acting at point 2
F3=6//force(in N) acting at point 3
F4=4//force(in N) acting at point 4
theta1=0//angle(in degree)
theta2=37//angle(in degree)
```

```
theta3=53//angle(in degree)
theta4=60//angle(in degree)

//calculation
Fx=(-F1*cosd(theta1))+(F2*cosd(theta2))+(F3*cosd(theta3))+(F4*cosd(theta4))//X component of resultant force
Fy=(F1*sind(theta1))+(F2*sind(theta2))+(-F3*sind(theta3))+(F4*sind(theta4))//X component of resultant force
F=sqrt((Fx*Fx)+(Fy*Fy))
theta=atand(Fy/Fx)
acm=F/M//acceleration of centre of mass

printf('the acceleration of the centre of mass is %3.1f m/s^2 and is in the direction of the resultant force',acm)
```

Scilab code Exa 9.8w calculation of the distance from launching point

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.8w
//calculation of the distance from launching point

//given data
u=100//speed(in m/s) of the projectile
theta=37//angle(in degree) of the projectile above horizontal
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
xcm=(2*u*u*sind(theta)*cosd(theta))/g//range of
```

```
    original projectile
//also   xcm=((m1*x1)+(m2*x2))/(m1+m2)
//here m1=M/4   and   m2=3*M/4
x1=xcm/2//since small part falls from heighest point
    i.e half of range
x2=(4/3)*((xcm*((1/4)+(3/4)))-(x1/4))

printf('the distance of landing of heavier piece
    from launching point is %d m',x2)
```

Scilab code Exa 9.9w calculation of the distance moved by the bigger block

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.9w
//calculation of the distance moved by the bigger
    block

//given data
L=2.2//length(in m) of the base
n=10// mass of bigger block is 'n' number of times
    the mass of smaller block

//calculation
//centre of mass at rest initially will remain in
    horizontal position thus
//M*(L-X)=10*M*X
X=L/(n+1)

printf('distance moved by the bigger block at the
    instant the smaller block reaches the ground is
    %3.1f m',X)
```

Scilab code Exa 9.10w calculation of the average force exerted by the hero on the machine gun

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.10w
//calculation of the average force exerted by the
   hero on the machine gun

//given data
m=50*10^-3//mass(in kg) of the bullet
v=1*10^3//velocity(in m/s) of the bullet
n=20//number of bullets fired
t=4//time(in s) required in firing the bullets

//calculation
me=m*v//momentumof each bullet
f=me*n/t//force=rate of change of momentum

printf('the average force exerted by the hero on the
    machine gun is %d N',f)
```

Scilab code Exa 9.11w calculation of the fractional change in kinetic energy

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.11w
//calculation of the fractional change in kinetic
   energy
```

```

//given data
vb=20//speed(in m/s) of the block
v1=30//velocity(in m/s) of one of the part

//calculation
M=1//taking mass M=1 kg  for solving the equation
v=(1/M)*((M*vb*2)-(M*v1))//principle of conservation of linear momentum
deltake=(M*v1*v1/(2*2))+(M*v*v/(2*2))-(M*vb*vb/2)//change in the kinetic energy
fdeltake=deltake/(M*vb*vb/2)//fractional change in the kinetic energy

printf('the fractional change in the kinetic energy is %3.2f',fdeltake)
```

Scilab code Exa 9.13w calculation of the final velocity of the shuttle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.13w
//calculation of the final velocity of the shuttle

//given data
v1=4000//speed(in km/hr) of shuttle with respect to the earth
v2=100//speed(in km/hr) of the module with respect to the shuttle

//calculation
M=1//taking mass M=1 kg  for solving the equation
vdash=v1-v2//speed of module with respect to the earth
```

```
V=(1/5)*((1*v1*6)-(vdash*1))//principle of conservation of linear momentum

printf('the final velocity of the shuttle is %d km/h',V)
```

Scilab code Exa 9.14w calculation of the velocity with which the board recoils

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.14w
//calculation of the velocity with which the board recoils

//given data
m1=25//mass(in kg) of the boy
m2=10//mass(in kg) of the board
v1=5//velocity(in m/s) of the boy

//calculation
v=(m1*v1)/m2//principle of conservation of linear momentum
vsep=v1+v//velocity of separation

printf('the velocity with which the board recoils is %3.1f m/s',v)
printf('\nthe velocity of separation of the boy and the board is %3.1f m/s',vsep)
```

Scilab code Exa 9.17w calculation of the speed of the bullet

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.17w
//calculation of the speed of the bullet

//given data
mb=50*10^-3//mass(in kg) of the bullet
mp=450*10^-3//mass(in kg) of the bob
h=1.8//height(in m) attained by the bob
g=10//gravitational acceleration(in m/s^2) of the
    earth

//calculation
//using principle of conservation of linear momentum
      and equation of motion (v*v) = (u*u) + (2*a*x)
v=((mb+mp)*(sqrt(h*2*g)))/mb

printf('the speed of the bullet is %d m/s',v)
```

Scilab code Exa 9.22w calculation of the loss of kinetic energy due to the collision

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 9.22w
//calculation of the loss of kinetic energy due to
    the collision

//given data
m=1.2//mass(in kg) of the block1
v=20*10^-2//velocity(in m/s) of the approach
e=3/5//value of coefficient of restitution
vdash=e*v//velocity (in m/s) of the separation
```

```
//calculation
//by principle of conservation of linear momentum
    ....v1 + v2 = v m/s.....(1)
//as the coefficient of restitution is
    3/5............v2 − v1 = vdash m/s.....(2)
//from equation (1),we get.......v2=v−v1
//substituting v2 in equation (2),we get
v1=(v-vdash)/2
v2=v-v1//from equation (1)
lke=(m/2)*((v*v)-(v1*v1)-(v2*v2))

printf('the loss of kinetic energy during the
    collision is %3.1e J',lke)
```

Chapter 10

Rotational Mechanics

Scilab code Exa 10.1 calculation of the number of revolutions made

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.1
//calculation of the number of revolutions made

//given data
wzero=100*2*%pi/60//initial angular velocity(in rad/s) of the motor
w=0//final angular velocity(in rad/s) of the motor
t=15//time interval(in s)

//calculation
alpha=(w-wzero)/t//equation of angular motion
theta=(wzero*t)+(alpha*t*t/2)//equation of angular motion

printf('the number of revolutions the motor makes before coming to rest is %3.1f',theta/(2*%pi))
```

Scilab code Exa 10.1w calculation of the number of revolutions made by the wheel

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.1w
//calculation of the number of revolutions made by
   the wheel

//given data
wzero=0//initial angular velocity(in rad/s) of the
   wheel
alpha=2//angular acceleration(in rad/s^2)
t=10//time(in s) interval

//calculation
theta=(wzero*t)+(alpha*t*t/2)//equation of angular
   motion
n=round(theta/(2*%pi))//number of revolutions

printf('the number of revolutions made by the wheel
   is %d',n)
```

Scilab code Exa 10.2 calculation of the time taken by the fan to attain half of the maximum speed

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.2
```

```
//calculation of the time taken by the fan to attain
    half of the maximum speed

//given data
wzero=0//initial angular velocity(in rad/s) of the
   fan
w=400*(2*%pi/60)//final angular velocity(in rad/s)
   of the fan
t=5//tiem(in s) taken

//calculation
alpha=(w-wzero)/t//equation of angular motion
wdash=w/2//half of maximum speed
t1=(wdash-wzero)/alpha//equation of angular motion

printf('the time taken by the fan to attain half of
   the maximum speed is %3.1f s',t1)
```

Scilab code Exa 10.2w calculation of the angle rotated during the next second

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.2w
//calculation of the angle rotated during the next
   second

//given data
theta=2.5//angular displacement(in rad) of the wheel
t=1//time(in s) required

//calculation
alpha=(theta*2)/(t*t)//equation of angular motion
theta1=(alpha*(t+1)*(t+1)/2)//angle rotated during
```

```
first two seconds
thetar=theta1-theta//angle rotated during next second

printf('the angle rotated during the next second is %3.1f rad',thetar)
```

Scilab code Exa 10.3 calculation of the angular velocity and angular acceleration of the pulley

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.3
//calculation of the angular velocity and angular acceleration of the pulley

//given data
v=20//linear speed(in cm/s) of the bucket
r=10//radius(in cm) of the pulley
a=4*10^2//linear acceleration(in cm/s^2) of the pulley

//calculation
w=v/r//formula of angular velocity
alpha=a/r//formula of angular acceleration

printf('the angular velocity of the pulley is %d rad/s and angular acceleration of the pulley is %d rad/s^2',w,alpha)
```

Scilab code Exa 10.3w calculation of the torque required to stop the wheel in one minute

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.3w
//calculation of the torque required to stop the
   wheel in one minute

//given data
wzero=50*(2*%pi/60)//initial angular velocity(in rad
   /s) of the wheel
w=0//final angular velocity(in rad/s) of the wheel
t=60//time(in s) taken to stop the wheel
I=2//moment of inertia(in kg-m^2) of the wheel

//calculation
alpha=(w-wzero)/t//equation of angular motion
tau=I*abs(alpha)//torque

printf('the torque required to stop the wheel in one
    minute is %3.2f N-m',tau)
```

Scilab code Exa 10.4w calculation of the angular velocity of the wheel

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.4w
//calculation of the angular velocity of the wheel

//given data
F=20//force(in N) of pull applied
I=.2//moment of inertia(in kg-m^2)
r=20*10^-2//radius(in m) of the wheel
t=5//time(in s) interval
wzero=0//initial angular velocity(in rad/s) of the
```

```
    wheel

//calculation
tau=F*r//torque applied to the wheel
alpha=tau/I//angular acceleration
w=wzero+(alpha*t)//equation of angular motion

printf('the angular velocity of the wheel after 5 s
    is %d rad/s',w)
```

Scilab code Exa 10.5 calculation of the moment of inertia of the wheel

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.5
//calculation of the moment of inertia of the wheel

//given data
r=10*10^-2//radius(in m) of the wheel
F=5//force(in N) of pulling
aplha=2//angular acceleration(in rad/s^2) of the
    wheel

//calculation
tau=F*r//net torque
I=tau/aplha//moment of inertia

printf('the moment of inertia of the wheel is %3.2f
    kg-m^2',I)
```

Scilab code Exa 10.7w calculation of the position of second kid on a balanced seesaw

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.7w
//calculation of the position of second kid on a
   balanced seesaw

//given data
ma=10//mass(in kg) of kid A
mb=15//mass(in kg) of kid B
l=5//length(in m) of the seesaw
la=(l/2)//distance of A kid from fulcrum as he is
   sitting at an end

//calculation
//taking torque about fulcrum...........(mb*g*x) = (
   ma*g*)
x=(ma*la)/mb

printf('the second kid should sit at a distance of
   %3.1f m from the centre',x)
```

Scilab code Exa 10.8w calculation of the normal force and the frictional force that the floor exerts on the ladder

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.8w
//calculation of the normal force and the frictional
    force that the floor exerts on the ladder

//given data
m=10//mass(in kg) of the ladder
theta=53//angle(in degree) made by the ladder
```

```
against the vertical wall
g=9.8//gravitational acceleration(in m/s^2) of the
earth

//calculation
//taking horizontal and vertical components
//N1 = f........................(1)
//N2 = W........................(2)
//taking torque about B
W=m*g
N2=W//from equation (2)
f=(W*sind(theta)/2)/(cosd(theta))//from equation (1)

printf('the normal force that the floor exerts on
the ladder is %d N',N2)
printf('\nthe frictional force that the floor exerts
on the ladder is %d N',f)
```

Scilab code Exa 10.9w calculation of the contact force exerted by the floor on each leg of ladder

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.9w
//calculation of the contact force exerted by the
floor on each leg of ladder

//given data
theta=60//angle(in degree) between the two legs
m=80//mass(in kg) of the person
g=9.8//gravitational acceleration(in m/s^2) of the
earth

//calculation
```

```
N=m*g/2
T=(N*2*tand(90-theta))/1

printf('the contact force exerted by the floor on each leg of ladder %d N',N)
printf('\nthe tension in the crossbar is %d N',T)
```

Scilab code Exa 10.12 calculation of the kinetic energy of the sphere

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.12
//calculation of the kinetic energy of the sphere

//given data
M=200*10^-3//mass(in kg) of the sphere
vcm=2*10^-2//speed(in m/s) of the sphere

//calculation
//kinetic energy is  K = (Icm*w*w/2) + (M*vcm*vcm/2)
//taking Icm = (2*M*r*r*w*w/5) and w=vcm/r
K=(M*vcm*vcm/5)+(M*vcm*vcm/2)//kinetic energy

printf('the kinetic energy of the sphere is %3.1e J',K)
```

Scilab code Exa 10.13w calculation of the kinetic energy and angular momentum of the disc

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 10.13w
//calculation of the kinetic energy and angular
   momentum of the disc

//given data
M=200*10^-3//mass(in kg) of the disc
r=4*10^-2//radius(in m) of the disc
w=10//angular velocity(in rad/s)

//calculation
I=(M*r*r)/4//moment of inertia
K=(I*w*w/2)//kinetic energy
L=I*w//angular momentum

printf('the kinetic energy of the disc is %3.1e J',K
   )
printf('\nthe angular momentum of the disc is %3.1e
   J-s',L)
```

Scilab code Exa 10.14w calculation of the work done by the torque in first two seconds

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.14w
//calculation of the work done by the torque in
   first two seconds

//given data
wzero=20//initial angular velocity(in rad/s) of the
   motor
w=0//final angular velocity(in rad/s) of the motor
t=4//time(in s) taken to attain rest position
I=.20//moment of inertia(in kg-m^2) of the disc
```

```
    about axis of rotation
t1=2//time(in s)

//calculation
alpha=(wzero-w)/t//equation of angular motion in case of deceleration
tau=I*alpha//torque
theta=(wzero*t1)-(alpha*t1*t1/2)//equation of angular motion
W=tau*theta//work done by the torque

printf('the work done by the torque in first two seconds is %d J',W)
```

Scilab code Exa 10.19w calculation of the moment of inertia of the system about the axis perpendicular to the rod passing through its middle point

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.19w
//calculation of the moment of inertia of the system about the axis perpendicular to the rod passing through its middle point

//given data
m=1.2//mass(in kg) of the sphere
R=10*10^-2//radius(in cm) of the sphere
sep=50*10^-2//separation(in m) between the two spheres

//calculation
d=sep/2//distance of each sphere from centre
Icm=(2*m*R*R)/5//moment of inertia about diameter
I=Icm+(m*d*d)//by parallel axis theorem,moment of
```

```
    inertia about given axis
//since second sphere has same moment of inertia
Isys=2*I//moment of inertia of the system

printf('the moment of inertia of the system about
    the axis perpendicular to the rod passing through
     its middle point is %3.3f kg-m^2',Isys)
```

Scilab code Exa 10.22w calculation of the number of revolutions made by the wheel per second

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 10.22w
//calculation of the number of revolutions made by
    the wheel per second

//given data
p=220*10^-2//perimeter(in cm) of the wheel
v=9*10^3/(60*60)//linear speed(in m/s) of wheel on
    the road

//calculation
r=p/(2*%pi)//radius of the wheel
w=v/r//angular speed
n=w/(2*%pi)//number of revolutions

printf('the number of revolutions made by the wheel
    per second is %3.2f rev/s',n)
```

Chapter 11

Gravitation

Scilab code Exa 11.1 calculation of the initial acceleration of the particles

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.1
//calculation of the initial acceleration of the
   particles

//given data
m1=1//masss(in kg) of particle1
m2=2//masss(in kg) of particle2
r=50*10^-2//separation(in m) between the two
   particles
G=6.67*10^-11//universal constant of gravitation(in
   N-m^2/kg^2)

//calculation
F=G*m1*m2/(r*r)//force of gravitation
a1=F/m1//initial acceleration of the particle1
a2=F/m2//initial acceleration of the particle2

printf('the initial acceleration of the particle1
```

```
towards particle2 is %3.1e m/s^2',a1)
printf('\nthe initial acceleration of the particle2 towards particle1 is %3.1e m/s^2',a2)
```

Scilab code Exa 11.2 calculation of the work done in bringing three particles together

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.2
//calculation of the work done in bringing three particles together

//given data
m1=100*10^-3//masss(in kg) of particle1
r=20*10^-2//separation(in m) between the two particles
G=6.67*10^-11//universal constant of gravitation(in N-m^2/kg^2)

//calculation
//since the work done by the gravitational force is equal to change in the potential energy
U=3*(-G*m1*m1/r)

printf('the work done in bringing three particles is %3.1e J',U)
```

Scilab code Exa 11.2w calculation of the distance from the earth surface where resultant gravitational field due to the earth and the moon is zero

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 11.2w
//calculation of the distance from the earth's
   surface where resultant gravitational field due
   to the earth and the moon is zero

//given data
Me=6*10^24//mass(in kg) of the earth
Mm=7.4*10^22//mass(in kg) of the moon
d=4*10^5*10^3//distance(in m) between the earth and
   the moon

//calculation
//gravitational field due to the earth at that point
//E1 = G*Me/x^2.........................(1)
//gravitational field due to the moon at that point
//E2 = G*Mm/(d-x)^2.....................(2)
//E1 = E2.....given
x=(d*sqrt(Me/Mm))/(1+sqrt(Me/Mm))

printf('the distance from the earth surface where
   resultant gravitational field due to the earth
   and the moon is zero is %3.1e km',x*10^-3)
```

Scilab code Exa 11.4 calculation of the gravitational field

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.4
//calculation of the gravitational field

//given data
F=2//gravitational force(in N)
```

```
m=50*10^-3//mass(in kg) of the particle

//calculation
E=F/m//gravitational field

printf('the gravitational field along the direction of force is %d N/kg',E)
```

Scilab code Exa 11.4w calculation of the separation between the particles under mutual attraction

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.4w
//calculation of the separation between the particles under mutual attraction

//given data
mA=1//mass(in kg) of particle A
mB=2//mass(in kg) of particle B
R=1//initial distance(in m) between the two particles
vB=3.6*10^-2/(60*60)//speed(in m/s) of the particle B
G=6.67*10^-11//universal constant of gravitation(in N-m^2/kg^2)

//calculation
v=(mB*vB)/mA//principle of conservation of linear momentum
U1=-G*mA*mB/R//initial potential energy of the pair
d=U1/(U1-(mB*vB*vB/2)-(mA*v*v/2))//principle of conservation of energy

```

```
printf('the speed of particle A is %3.1e m/s',v)
printf('\nthe separation between the particles under mutual attraction is %3.2f m',d)
```

Scilab code Exa 11.5w calculation of the work done by an external agent

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.5w
//calculation of the work done by an external agent

//given data
//E = (10 N/kg)(i + j).....given gravitational field
Ex=10//value of X-component of gravitational field(in N/kg)
Ey=10//value of Y-component of gravitational field(in N/kg)
m=2//mass(in kg) of the gravitational field
x0=0//value of X component of initial location(in m)
x1=5//value of X component of final location(in m)
y0=0//value of Y component of initial location(in m)
y1=4//value of Y component of final location(in m)

//calculation
function Fx=fx(x)
    Fx=m*Ex//value of X component of force
endfunction

function Fy=fy(x)
    Fy=m*Ey//value of Y component of force
endfunction

//calculation
W1=integrate('fx','x',x0,x1)//work done by X
```

```
component of external force
W2=integrate('fy','x',y0,y1)//work done by Y component of external force
W=W1+W2

printf('the work done by the external agent is %d J',-W)
```

Scilab code Exa 11.7 calculation of the gravitational field due to the moon at its surface

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.7
//calculation of the gravitational field due to the moon at its surface

//given data
M=7.36*10^22//mass(in kg) of the moon
G=6.67*10^-11//universal constant of gravitation(in N-m^2/kg^2)
a=1.74*10^6//radius(in m) of the moon

//calculation
E=G*M/(a*a)//formula of gravitational field

printf('the gravitational field due to the moon at its surface is %3.2f N/kg',E)
```

Scilab code Exa 11.8 calculation of the value of acceleration due to gavity

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 11.8
//calculation of the value of acceleration due to
    gavity

//given data
h=5*10^3//height(in m) above the earth's surface
R=6400*10^3//radius(in m) of the earth
g0=9.8//gravitational acceleration(in m/s^2) of the
    earth
d=5*10^3//depth(in m) below the earth's surface

//calculation
gh=g0*(1-(2*h/R))//formula of gravitational
    acceleration at height h above the earth's
    surface
gd=g0*(1-(d/R))//formula of gravitational
    acceleration at depth d below the earth's surface

printf('the value of gravitational acceleration at
    height 5 km above the earth surface is %3.2f m/s
    ^2',gh)
printf('\nthe value of gravitational acceleration at
     depth 5 km below the earth surface is %3.2f m/s
    ^2',gd)
```

Scilab code Exa 11.9 calculation of the speed and time period of the satellite

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.9
//calculation of the speed and time period of the
```

```
   satellite

//given data
h=600*10^3//height(in m) of the satellite
M=6*10^24//mass(in kg) of the earth
R=6400*10^3//radius(in m) of the earth
G=6.67*10^-11//universal constant of gravitation(in
   N-m^2/kg^2)

//calculation
a=h+R//distance of satellite from centre of the
   earth
v=sqrt(G*M/a)//speed of satellite
T=(2*%pi*a)/v//time period of satellite

printf('the speed of the satellite is %3.1e m/s or
   %3.1f km/s',v,v*10^-3)
printf('\nthe time period of the satellite is %3.1e
   s',T)
```

Scilab code Exa 11.9w calculation of the maximum height attained by the particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.9w
//calculation of the maximum height attained by the
   particle

//given data
v0=9.8*10^3//speed(in m/s) the particle is fired
R=6400*10^3//radius(in m) of the earth
g=9.8//gravitational acceleration(in m/s^2) of the
   earth
```

```
//calculation
//by the principle of conservation of energy
//(-G*M*m/R) + (m*v0*v0/2) = -(G*M*m/(R+H))
H=(R*R/(R-(v0*v0/(2*g))))-R

printf('the maximum height attained by the particle
    is %d km',H*10^-3)
```

Scilab code Exa 11.10 calculation of the escape velocity from the moon

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.10
//calculation of the escape velocity from the moon

//given data
M=7.4*10^22//mass(in kg) of the moon
R=1740*10^3//radius(in m) of the moon
G=6.67*10^-11//universal constant of gravitation(in
    N-m^2/kg^2)

//calculation
v=sqrt(2*G*M/R)//formula of the escape velocity

printf('the escape velocity from the moon is %3.1f
    km/s',v*10^-3)
```

Scilab code Exa 11.10w calculation of the stretch produced in the spring

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 11.10w
//calculation of the stretch produced in the spring

//given data
d=1*10^-2//stretch(in m) of the spring
R=6400*10^3//radius(in m) of the earth
h=800*10^3//height(in m) above the earth's surface

//calculation
//The extension in the spring on the surface is
//1*10^-2 = (G*M*m)/(k*R^2)...........(1)
//The extension in the spring at height h above the surface
//x = (G*M*m)/(k*(R+h)^2).............(2)
//from above equations,we get
x=d*((R^2)/(R+h)^2)

printf('the stretch produced in the spring is %3.2f cm',x*10^2)
```

Scilab code Exa 11.11w calculation of time period of the pendulum if used at the equator

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.11w
//calculation of time period of the pendulum if used at the equator

//given data
t=2//time period (in s) of the pendulum at North pole
g=9.8//gravitational acceleration(in m/s^2) of the
```

```
earth
G=6.67*10^-11//universal constant of gravitation(in N-m^2/kg^2)
w=(2*%pi)/(24*60*60)//angular velocity(in rad/s) of the earth
R=6400*10^3//radius(in m) of the earth

//calculation
//By equilibrium conditions ,we get
//t = 2*%pi*sqrt(l/g) ...............................(1)
//tdash = 2*%pi*sqrt(l/(g-(w*w*R)) .................(2)
//from equations (1) and (2),we get
tdash=t*(1+(w*w*R/(2*g)))

printf('the value of time period of the pendulum if used at the equator is %3.4f s',tdash)
```

Scilab code Exa 11.12w calculation of the speed of projection of the satellite into an orbit

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.12w
//calculation of the speed of projection of the satellite into an orbit

//given data
r=8000*10^3//radius(in m) of the orbit of the satellite
R=6400*10^3//radius(in m) of the earth
g=9.8//gravitational acceleration(in m/s^2) of the earth
```

```

//calculation
//using Newton's second law
//(G*M*m/(r*r)) = m*v*v/r
v=sqrt(g*R*R/r)
t=(2*%pi*r/v)//time period of the satellite

printf('the speed of projection of the satellite into the orbit is %3.2f km/s',v*10^-3)
printf('\nthe time period of the satellite in the orbit is %d minutes',t*(1/(60)))
```

Scilab code Exa 11.13w calculation of the speed and the angular speed of the satellite S2 relative to the satellite S1

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 11.13w
//calculation of the speed and the angular speed of the satellite S2 relative to the satellite S1

//given data
T1=1//period of revolution(in h) of satellite S1
T2=8//period of revolution(in h) of satellite S2
R1=10^4//radius(in km) of the orbit of satellite S1

//calculation
//by Kelpler's third law
//(R2/R1)^3 = (T2/T1)^2
R2=R1*(((T2/T1)^2)^(1/3))
v1=(2*%pi*R1/T1)//speed(in km/h) of satellite S1
v2=(2*%pi*R2/T2)//speed(in km/h) of satellite S2
v=abs(v2-v1)//speed of satellite S2 with respect to satellite S1
```

```
w=v/(R2-R1)//angular speed of satellite S2 as
    observed by an astronaut in satellite S1

printf('the speed of the satellite S2 with respect
    to the satellite S1 is %3.1e km/h',v)
printf('\nthe angular speed of the satellite S2 as
    observed by an astronaut in the satellite S1 is
    %3.2f rad/h',w)
```

Chapter 12

Simple Harmonic Motion

Scilab code Exa 12.1 calculation of the spring constant

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.1
//calculation of the spring constant

//given data
F=4//force(in N) acting
x=5*10^-2//distance(in m) from the centre

//calculation
k=F/x//value of spring constant

printf('the value of spring constant is %d N/m',k)
```

Scilab code Exa 12.1w calculation of the amplitude time period maximum speed and velocity at time t

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.1w
//calculation of the amplitude,time period,maximum speed and velocity at time t

//given data
//x = (5 m)*sind((%pi s^-1)t + (180/3))...... equation of simple harmonic motion

//calculation
A=5//amplitude(in m)
w=%pi
T=(2*%pi)/w//time period(in s)
vmax=A*w//maximum speed
v=A*w*cosd(180+(180/3))

printf('the amplitude is %d m',A)
printf('\nthe time period is %d s',T)
printf('\nthe maximum speed is %3.2f m/s',vmax)
printf('\nthe velocity at time t=1 s is %3.2f m/s',v)
```

Scilab code Exa 12.2 calculation of the amplitude of the motion

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.2
//calculation of the amplitude of the motion

//given data
m=0.5//mass(in kg) of the particle
//F = -50*x ......force(in N/m)
```

```
v=10//speed(in m/s) of the oscillation

//calculation
E=(m*v*v/2)//kinetic energy of the particle at
    centre of oscillation
//from principle of conservation of energy......E =
    (k*A*A/2)
A=sqrt(E*2/50)

printf('the amplitude of the motion is %d m',A)
```

Scilab code Exa 12.2w calculation of the maximum force exerted by the spring on the block

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.2w
//calculation of the maximum force exerted by the
    spring on the block

//given data
m=5//masss(in kg) of the block
A=0.1//amplitude(in m) of the motion
T=3.14//time period(in s) of the motion

//calculation
w=2*%pi/T//angular frequency
k=m*w*w//spring constant
F=k*A//maximum force

printf('the maximum force exerted by the spring on
    the block is %d N',F)
```

Scilab code Exa 12.3 calculation of the time period of oscillation of the particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.3
//calculation of the time period of oscillation of
   the particle

//given data
m=200*10^-3//mass(in kg) of the particle
k=80//spring constant(in N/m)

//calculation
T=2*%pi*sqrt(m/k)//formula of time period

printf('the time period of oscillation of the
   particle is %3.2f s',T)
```

Scilab code Exa 12.3w calculation of the maximum time period maximum speed maximum acceleration speed for a given displacement speed at a given time

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.3w
//calculation of the maximum time period ,maximum
   speed ,maximum acceleration ,speed for a given
   displacement ,speed at a given time

```

```
//given data
w=6.28//angular frequency(in s^-1) of simple harmonic motion
A=10*10^-2//amplitude(in m) of simple harmonic motion
x=6*10^-2//displacement(in m) from the mean position
t=1/6//time(in s)

//calculation
T=2*%pi/w//time period
vmax=A*w//maximum speed
amax=A*w^2//maximum acceleration
vx=w*sqrt(A^2-x^2)//speed for displacement x from mean position
vt=-A*w*sind((w*t)*(180/%pi))//speed at time t

printf('the time period is %d s',T)
printf('\nthe maximum speed is %3.3f m/s',vmax)
printf('\nthe maximum acceleration is %d m/s^2',round(amax))
printf('\nthe speed for displacement x=6 cm from mean position is %3.1f cm/s',vx*10^2)
printf('\nthe speed at time t= 1/6 s is %3.1f cm/s',vt*10^2)
```

Scilab code Exa 12.4 calculation of the value of phase constant

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.4
//calculation of the value of phase constant

//given data
//x = A/2
```

```
//x = A *sind((w*t) + delta).......equation

//calculation
//at t=0   delta=asind((A/2)/A)
delta=asind(1/2)
delta1=180-delta//another value of delta
//v = dx/dt = A*w*cosd((w*t) + delta)
//at t=0 , v = A*w*cosd(delta)
m1=cosd(delta)
m2=cosd(delta1)
if(m1>0)
    deltaf=delta//value of v positive at t=0
else
    deltaf=delta1
end

printf('the value of phase constant is %d degree',
   deltaf)
```

Scilab code Exa 12.5 calculation of the total mechanical energy of the system

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.5
//calculation of the total mechanical energy of the
   system

//given data
m=40*10^-3//mass(in kg) of the particle
A=2*10^-2//amplitude(in cm) of motion
T=0.2//time period(in s) of oscillation

//calculation
```

```
E=(2*%pi*%pi*m*A*A)/(T*T)//total mechanical energy of the system

printf('the total mechanical energy of the system is %3.1e J',E)
```

Scilab code Exa 12.6 writing the equation giving angular displacement as a function of time

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.6
//writing the equation giving angular displacement as a function of time

//given data
theta0=%pi/10//amplitude(in rad) of motion
theta=%pi/10//displacement(in rad) at t=0 s
T=.05//time period(in s)

//calculation
//required equation is ......theta = theta0*sind((w*t) + delta)
w=(2*%pi)/T//value of w in above equation
delta=asind(theta/theta0)//value of delta in above equation...i.e at t=0

printf('equation giving angular displacement as a function of time is \n theta = (%3.2f rad)*sin[(%3.2f s^-1)t + %d]',theta0,w,delta)
```

Scilab code Exa 12.6w calculation of the maximum speed of the block and the speed when the spring is stretched

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.6w
//calculation of the maximum speed of the block and the speed when the spring is stretched

//given data
nu=10//frequency(in s^-1) of oscillation
l=.20*10^-2//stretch(in m) of the spring
g=%pi^2//gravitational acceleration(in m/s^2) of the earth

//calculation
//Amplitude.................A = m*g/k ...............(1)
//angular frequency.......w=sqrt(k/m) ..............(2)
//from above equations,we get
w=2*%pi*nu//angular frequency
A=((1/w)^2)*g
vmax=A*w//maximum speed
x=A-l//displacement(in m) from mean position
v=w*(sqrt(A^2-x^2))

printf('the maximum speed of the block is %3.2f cm/s',vmax*10^2)
printf('\nthe speed when the spring is stretched by 0.20 cm is %3.1f cm/s',v*10^2)
```

Scilab code Exa 12.7 calculation of the time period of a pendulum

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.7
//calculation of the time period of a pendulum

//given data
g=%pi^2//gravitational acceleration(in m/s^2) of the earth
l=1//length(in m) of the pendulum

//calculation
T=2*%pi*sqrt(l*g^-1)//formula of time period

printf('the time period of the pendulum is %3.1f s',T)
```

Scilab code Exa 12.8 calculation of the value of the acceleration due to gravity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.8
//calculation of the value of the acceleration due to gravity

//given data
t=36//time(in s) taken
n=20//number of oscillations
l=80*10^-2//effective length(in m)

//calculation
T=t/n//time period
g=(4*%pi^2*l)/(T^2)//formula of time period
```

```
..........T=2*%pi*sqrt(l*g^-1)

printf('the value of the acceleration due to gravity is %3.2f m/s^2',g)
```

Scilab code Exa 12.9 calculation of the time period of oscillation

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.9
//calculation of the time period of oscillation

//given data
L=1//length(in m) of the rod
g=9.8//gravitational acceleration(in m/s^2) of the earth

//calculation
//from formula of time period......T = 2*%pi*sqrt(I/(m*g*l))
//for uniform rod ....I = (m*L*L/3) and l=L/2
T=2*%pi*sqrt((2*L)/(3*g))

printf('the time period of oscillation is %3.2f s',T)
```

Scilab code Exa 12.10 calculation of the value of torsional constant of the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 12.10
//calculation of the value of torsional constant of
   the wire

//given data
m=200*10^-3//mass(in kg) of the disc
r=5*10^-2//radius(in m) of the disc
T=0.2//time period(in s) of oscillation

//calculation
I=m*r*r/2//moment of inertia of the disc about the
   wire
k=4*%pi^2*I/T^2//from formula of time period......T
   = 2*%pi*sqrt(I/k)

printf('the value of torsional constant of the wire
   is %3.2f kg-m^2/s^2',k)
```

Scilab code Exa 12.11 calculation of the amplitude of the simple harmonic motion

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.11
//calculation of the amplitude of the simple
   harmonic motion

//given data
//x1 = (2.0 cm)*sind(w*t)
//x2 = (2.0 cm)*sind((w*t) + (180/3))
A1=2//amplitude(in cm) of the wave 1
A2=2//amplitude(in cm) of the wave 2
delta=180/3//phase difference(in degree) between the
    two waves
```

```
//calculation
A=sqrt(A1^2+A2^2+(2*A1*A2*cosd(delta)))//amplitude of the resultant wave

printf('the amplitude of the simple harmonic motion is %3.1f cm',A )
```

Scilab code Exa 12.14w calculation of the time period linear amplitudde speed and angular acceleration

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.14w
//calculation of the time period,linear amplitudde, speed and angular acceleration

//given data
l=40*10^-2//length(in m) of the pendulum
theta=.04//angular amplitude(in radian)
theta1=.02//angle(in radian) with the vertical
g=10//gravitational acceleration(in m/s^2) of the earth
t=5//time(in s) taken

//calculation
w=sqrt(g/l)//angular frequency
T=2*%pi/w//time period
A=l*theta//linear amplitude
ohm=t*sqrt(theta^2-theta1^2)//angular speed at disp[lacement theta1
v=l*ohm//linear speed
alpha=theta*w^2//angular acceleration

```

```
printf('the time period of the pendululum is %3.2f s',T)
printf('\nthe linear amplitude of the pendulum is %3.1f cm',A*10^2)
printf('\nthe linear speed of the pendulum at displacement of 0.02 rad is %3.1f cm/s',v*10^2)
printf('\nthe angular acceleration of the pendulum is %d rad s^-2',alpha)
```

Scilab code Exa 12.16w calculation of the time period of small oscillations

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.16w
//calculation of the time period of small oscillations

//given data
//h=R.....height equal to radius of the circle
g=%pi^2//gravitational acceleration(in m/s^2) of the earth
l=1//length(in m) of the string

//calculation
//at height R
//gdash = G*M/(R+R)^2 = g/4
gdash=g/4
T=2*%pi*sqrt(l/gdash)//time period

printf('The time period of small oscillations is %d s',T)
```

Scilab code Exa 12.18w calculation of the time period of small oscillation about the point of suspension

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.18w
//calculation of the time period of small oscillation about the point of suspension

//given data
l=1//length(in m) of the stick
d=40*10^-2//distance(in m) of the centre from point of suspension
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
//moment of inertia.....I = (m*l*l/12) + (m*d*d)
//time period...........T=2*%pi*sqrt(I/m*g*d)
//solving the above equations,we get
T=2*%pi*sqrt(((l*l/12)+(d*d))/(g*d))

printf('the time period of small oscillation about the point of suspension is %3.2f s',T)
```

Scilab code Exa 12.19w calculation of the moment of inertia of the second disc about the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 12.19w
//calculation of the moment of inertia of the second disc about the wire

//given data
I=0.2//moment of inertia(in kg-m^2) of the original disc
T=2//time period(in s) of the oscillation of the original disc
T1=2.5//time period(in s) of the oscillation of the system of two discs

//calculation
//from equation of time period......T = 2*%pi*sqrt(I/K)
I1=((T1^2/T^2)*(I))-I//moment of inertia of the second disc

printf('the moment of inertia of the second disc about the wire is %3.2f kg-m^2',I1)
```

Scilab code Exa 12.22w calculation of the phase difference between the individual motions

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 12.22w
//calculation of the phase difference between the individual motions

//given data
//amplitudes of both the waves are same
//resultant amplitude is equal to individual amplitudes
```

```
//calculation
//the resultant amplitude is.......A = sqrt(A^2 + A^2 + 2*A*A*cosd(delta))
//on further solving..........A = 2*A*cos(delta/2)
delta=2*(acosd(1/2))

printf('the phase difference between the individual motions is %d degree',delta)
```

Chapter 13

Fluid Mechanics

Scilab code Exa 13.1 calculation of the force exerted by the water on the bottom

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.1
//calculation of the force exerted by the water on
   the bottom

//given data
h=20*10^-2//height(in m) of the flask
r=10*10^-2//radius(in m) of the bottom of the flask
P0=1.01*10^5//atmospheric pressure(in Pa)
rho=1000//density of water(in kg/m^3)
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
P=P0+(h*rho*g)//pressure at the bottom
A=%pi*r^2//area of the bottom
F=P*A//force on the bottom

```

```
printf('the force exerted by the water on the bottom
    is %d N',F)
```

Scilab code Exa 13.1w calculation of the force exerted by the mercury on the bottom of the beaker

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.1w
//calculation of the force exerted by the mercury on
    the bottom of the beaker

//given data
h=10*10^-2//height(in m) of the mercury
r=4*10^-2//radius(in m) of the beaker
g=10//gravitational acceleration(in m/s^2) of the
    earth
P0=1*10^5//atmospheric pressure(in Pa)
rho=13600//density of mercury(in kg/m^3)

//calculation
P=P0+(h*rho*g)//pressure at the bottom
A=%pi*r^2//area of the bottom
F=P*A//force on the bottom

printf('the force exerted by the mercury on the
    bottom of the beaker is %d N',F)
```

Scilab code Exa 13.2 calculation of the volume of the cube outside the water

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 13.2
//calculation of the volume of the cube outside the
   water

//given data
m=700*10^-3//mass(in kg) of the cube
l=10*10^-2//length(in m) of the cube
rho=1000//density of water(in kg/m^3)

//calculation
V=m/rho//weight of displaced water = V*rho*g
Vtotal=l^3//total volume of the cube
Vout=Vtotal-V//volume of the cube outside the water

printf('the volume of the cube outside the water is
   %d cm^3',Vout*10^6)
```

Scilab code Exa 13.2w calculation of the height of the atmosphere to exert the same pressure as at the surface of the earth

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.2w
//calculation of the height of the atmosphere to
   exert the same pressure as at the surface of the
   earth

//given data
P0=1*10^5//atmospheric pressure(in Pa)
rho=1.3//density of air(in kg/m^3)
g=9.8//gravitational acceleration(in m/s^2) of the
   earth
```

```
//calculation
h=P0/(g*rho)

printf('the height of the atmosphere to exert the
   same pressure as at the surface of the earth is
   %d m',round(h))
```

Scilab code Exa 13.3 calculation of the speed of the outgoing liquid

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.3
//calculation of the speed of the outgoing liquid

//given data
A1=1*10^-4//area(in m^2) of the inlet of the tube
A2=20*10^-6//area(in m^2) of the outlet of the tube
v1=2//speed(in cm/s) of the ingoing liquid

//calculation
v2=A1*v1/A2//equation of continuity

printf('the speed of the outgoing liquid is %d cm/s'
   ,v2)
```

Scilab code Exa 13.3w calculation of the height of the water coloumn

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.3w
```

```
//calculation of the height of the water coloumn

//given data
h1=2*10^-2//difference in the height(in m)
s=13.6//specific gravity of mercury

//calculation
//P = P0 + (h*rho*g)........using this equation
h=h1*s//height of the water coloumn

printf('the height of the water coloumn is %d cm',h
    *10^2)
```

Scilab code Exa 13.4 calculation of the difference in the pressures at A and B point

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.4
//calculation of the difference in the pressures at
    A and B point

//given data
A1=1*10^-4//area(in m^2) at point A of the tube
A2=20*10^-6//area(in m^2) at point B of the tube
v1=10*10^-2//speed(in m/s) of the ingoing liquid
rho=1200//density of the liquid(in kg/m^3)

//calculation
v2=A1*v1/A2//equation of continuity
//by Bernoulli equtation.....P1 + (rho*g*h1) + (rho*
    v1^2/2) = P2 + (rho*g*h2) + (rho*v2^2/2)
deltaP=(1/2)*rho*(v2^2-v1^2)

```

```
printf('the difference in the pressures at A and B
    point is %d Pa',deltaP)
```

Scilab code Exa 13.5 calculation of the speed of the water coming out of the tap

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.5
//calculation of the speed of the water coming out
    of the tap

//given data
h=6//depth(in m) of the tap
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
v=sqrt(2*g*h)//torricelli's theorem

printf('the speed of the water coming out of the tap
     is %d m/s',round(v))
```

Scilab code Exa 13.5w calculation of the force applied on the water in the thicker arm

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.5w
//calculation of the force applied on the water in
    the thicker arm
```

```
//given data
A1=1*10^-4//area(in m^2) of arm 1
A2=10*10^-4//area(in m^2) of arm 2
f=5//force(in N) applied on the water in the thinner arm

//calculation
//P = P0 + (h*rho*g)........using this equation
F=f*A2/A1//force applied on the water in the thicker arm

printf('the force applied on the water in the thicker arm is %d N',F)
```

Scilab code Exa 13.6w calculation of the elongation of the spring

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.6w
//calculation of the elongation of the spring

//given data
m=10*10^-3//mass(in kg) of the copper piece
l=1*10^-2//elongation(in m) in the spring
g=10//gravitational acceleration(in m/s^2) of the earth
rho=9000//density of copper(in kg/m^3)
rho0=1000//density of water(in kg/m^3)

//calculation
k=m*g/l//spring constant
V=m/rho//volume of copper
Fb=V*rho0*g//force of buoyancy
```

```
x=((k*l)-Fb)/k//elongation of the spring

printf('the elongation of the spring is %3.2f cm',x
   *10^2)
```

Scilab code Exa 13.7w calculation of the maximum weight that can be put on the block without wetting it

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.7w
//calculation of the maximum weight that can be put
   on the block without wetting it

//given data
l=3*10^-2//length(in m) of the edge of the cubical
   block
rho=800//density of wood(in kg/m^3)
k=50//spring constant(in N/m)
g=10//gravitational acceleration(in m/s^2) of the
   earth
rho0=1000//density of water(in kg/m^3)

//calculation
s=rho/rho0//specific gravity
hin=l*s//height inside water
hout=l-hin//height outside water
V=l^3//volume of the block
Fb=V*rho0*g//force of buoyancy
Fs=k*hout//force exerted by the spring
Wdash=V*rho*g//weight of the block
W=Fb+Fs-Wdash//maximum weight

printf('the maximum weight that can be put on the
```

```
block without wetting it is %3.2f N',W)
```

Scilab code Exa 13.8w calculation of the angle that the plank makes with the vertical in equilibrium

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.8w
//calculation of the angle that the plank makes with the vertical in equilibrium

//given data
l=1//length(in m) of the planck
h=0.5//height(in m) of the water level in the tank
s=0.5//specific gravity of the planck

//calculation
//A = OC/2 = l/(2*cosd(theta)
// mg = 2*l*rho*g
//buoyant force  Fb=(2*l*rho*g)/cosd(theta)
//m*g*(OB)*sind(theta) = F(OA)*sind(theta)
theta=acosd(sqrt(1/2))

printf('the angle that the plank makes with the vertical in equilibrium is %d degree',theta)
```

Scilab code Exa 13.10w calculation of the rate of water flow through the tube

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 13.10w
//calculation of the rate of water flow through the tube

//given data
A1=30//area(in cm^2) of the tube at point A
A2=15//area(in cm^2) of the tube at point B
deltaP=600//change in pressure(in N/m^2)
rho0=1000//density of the water(in kg/m^3)

//calculation
r=A1/A2//ratio of area
//from equation of continuity  vB/vA = A1/A2 = r = 2
//by Bernoulli equtation.....P1 + (rho*g*h1) + (rho*v1^2/2) = P2 + (rho*g*h2) + (rho*v2^2/2)
//take vB = vA*2
vA=sqrt(deltaP*(r/(r+1))*(1/rho0))
Rflow=vA*A1//rate of water flow

printf('the rate of water flow through the tube is %d cm^3/s',Rflow*10^2)
```

Scilab code Exa 13.11w calculation of the velocity of the water coming out of the opening

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 13.11w
//calculation of the velocity of the water coming out of the opening

//given data
AA=.5//area(in m^2) of the tank
AB=1*10^-4//area(in m^2) of the cross section at the
```

```
bottom
m=20//mass(in kg) of the load
h=50*10^-2//height(in m)of the water level
g=10//gravitational acceleration(in m/s^2) of the earth
rho=1000//density of the water(in kg/m^3)

//calculation
//from the equation............P = P0 + (h*rho*g)//pressure at the bottom
r=m*g/AA//in above equation it is the value of (h*rho*g)
//on solving,we get............PA = P0 + (400 N/m^2)
//from Bernoulli equtation.....P1 + (rho*g*h1) + (rho*v1^2/2) = P2 + (rho*g*h2) + (rho*v2^2/2)
//we get
vB=sqrt((2*(r+(rho*g*h)))/rho)

printf('the velocity of the water coming out of the opening is %3.1f m/s',vB)
```

Chapter 14

Some Mechanical Properties of Matter

Scilab code Exa 14.1 calculation of the tensile stress developed in the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.1
//calculation of the tensile stress developed in the
    wire

//given data
m=4//mass(in kg) of the load
r=2*10^-3//radius(in m) of the wire
g=3.1*%pi//gravitational acceleration(in m/s^2) of
   the earth

//calculation
F=m*g//gravitational force
A=%pi*r^2//area
St=F/A//tensile stress

printf('the tensile stress developed in the wire is
```

```
%3.1e N/m^2',St)
```

Scilab code Exa 14.1w calculation of the extension of the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.1w
//calculation of the extension of the wire

//given data
L=2//lengh(in m)of the wire
A=.2*10^-4//area(in m^2)
m=4.8//mass(in kg)
Y=2*10^11//Young modulus of steel
g=10//gravitational acceleration(in m/s^2) of the
    earth

//calculation
T=m*g//weight
l=(T*L)/(A*Y)//exension

printf('the extension of the wire is %3.1e m',l)
```

Scilab code Exa 14.2 calculation of the value of Young modulus

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.2
//calculation of the value of Young modulus

//given data
```

```
m=4//mass(in kg) of the load
l=20//length(in m) of the steel wire
r=2*10^-3//radius(in m) of the steel wire
dl=.031*10^-3//increase in the length(in m)
g=3.1*%pi//gravitational acceleration(in m/s^2) of the earth

//calculation
Ssl=(m*g)/(%pi*r^2)//longitudinal stress
Stl=dl/l//longitudinal strain
Y=Ssl/Stl//Young modulus

printf('the value of Young modulus is %3.1e N/m^2',Y)
```

Scilab code Exa 14.2w calculation of the elongation of the rope and corresponding change in the diameter

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.2w
//calculation of the elongation of the rope and corresponding change in the diameter

//given data
L=4.5//length(in m) of the nylon rope
d=6*10^-3//diameter(in m) of the nylon rope
T=100//weight(in N) of the monkey
Y=4.8*10^11//Young modulus(in N/m^2) of the rope
Pr=.2//Poission ratio of nylon

//calculation
A=%pi*(d/2)^2//area of cross section
l=(T*L)/(A*Y)//elongation
```

```
deltad=(Pr*l*d)/(L)//change in diameter

printf('the elongation of the rope is %3.2e m',l)
printf('\nthe corresponding change in the diameter
   is %3.1e m',deltad)
```

Scilab code Exa 14.3 calculation of the elastic potential energy stored in the stretched steel wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.3
//calculation of the elastic potential energy stored
    in the stretched steel wire

//given data
l=2//length(in m) of the steel wire
A=4*10^-6//cross sectional area(in m^2) of the steel
    wire
dl=2*10^-3//increase in the length(in m)
Y=2*10^11//Young modulus(in N/m^2)

//calculation
St=dl/l//strain in the wire
Ss=Y*St//stress in the wire
V=A*l//volume of the steel wire
U=Ss*St*V/2

printf('the elastic potential energy stored in the
   stretched steel wire is %3.1f J',U)
```

Scilab code Exa 14.3w calculation of the minimum radius of the wire used if it is not to break

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.3w
//calculation of the minimum radius of the wire used
    if it is not to break

//given data
m1=1//mass(in kg) of block1
m2=2//mass(in kg) of block2
Ss=2*10^9//breaking stress(in N/m^2)  of the metal
g=10//gravitational acceleration(in m/s^2) of the
    earth

//calculation
//using equation ....stress = tension / Area of
    cross secion
//T - (m1*g) = m1 * a...........(1)
//(m2*g) - T = m2*a.............(2)
//Adding equation (1) and equation (2),we get
a=((m2*g)-(m1*g))/(m1+m2)
T=(m1*g)+(m1*a)//tension in the string from equation
    (1)
r=sqrt(T/(Ss*%pi))//radius

printf('the minimum radius of the wire used if it is
    not to break is %3.1e m',r)
```

Scilab code Exa 14.4 calculation of the force by which the surface on one side of the diameter pulls the suface on the other side

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 14.4
//calculation of the force by which the surface on
    one side of the diameter pulls the suface on the
    other side

//given data
r=5*10^-2//radius(in m) of the beaker
S=.075//surface tension(in N/m) of the water

//calculation
l=2*r//length of diameter of the surface
F=S*l//force

printf('the force by which the surface on one side
    of the diameter pulls the suface on the other
    side is %3.1e N',F)
```

Scilab code Exa 14.4w calculation of the ratio of the lengths of the two wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.4w
//calculation of the ratio of the lengths of the two
     wire

//given data
Ys=2*10^11//Young modulus(in N/m^2) of the steel
    wire
Yc=1.1*10^11//Young modulus(in N/m^2) of the copper
    wire

```

```
//calculation
//r = Ls/Lc......required ratio
r=Ys/Yc//required ratio

printf('the ratio of the lengths of the two wire(Ls/
   Lc) is %f:1',r)
```

Scilab code Exa 14.5 calculation of the gain in the surface energy

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.5
//calculation of the gain in the surface energy

//given data
R=10^-2//radius(in m) of the drop
n=1000//number of droplets formed
S=.075//surface tension(in N/m) of the water

//calculation
//volume of original drop = total volume of all
   droplets formed
r=R/n^(1/3)//radius of each droplet
A1=4*%pi*R^2//surface area of drop
A2=n*(4*%pi*r^2)//surface area of each droplet
deltaA=A2-A1//change in suface area
deltaU=deltaA*S//change in surface energy

printf('the gain in the surface energy is %3.1e J',
   deltaU)
```

Scilab code Exa 14.5w calculation of the decrease in the volume of the sample of water

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.5w
//calculation of the decrease in the volume of the
   sample of water

//given data
V1=1000*10^-6//initial volume(in m^3)
P1=10^5//initial pressure(in N/m^2)
P2=10^6//final pressure(in N/m^2)
C=50*10^-11//compressibility(in m^2/N)of the water

//calculation
deltap=P2-P1//change in pressure
//compressibility = 1/Bulk modulus = -(deltaV/V)/
   deltaP
deltaV=-(C*deltap*V1)

printf('the decrease in the volume of the sample of
   water is %3.2f cm^3',-deltaV*10^6)
```

Scilab code Exa 14.6 calculation of the excess pressure inside a mercury drop

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.6
//calculation of the excess pressure inside a
   mercury drop

```

```
//given data
R=2*10^-3//radius(in m) of the drop
S=.464//surface tension(in N/m) of the drop

//calculation
deltaP=2*S/R//excess pressure

printf('the excess pressure inside a mercury drop is
    %d N/m^2',deltaP)
```

Scilab code Exa 14.6w calculation of the longitudinal strain in two wires

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.6w
//calculation of the longitudinal strain in two
    wires

//given data
m1=1//mass(in kg) of load 1
m2=2//mass(in kg) of load 2
A=.005*10^-4//area(in m^2) of the cross section
Y=2*10^11///Young modulus(in N/m^2) of the wire
g=10//gravitational acceleration(in m/s^2) of the
    earth

//calculation
T1=m1*g//tension in wire 1
Ss1=T1/A//longitudinal stress
St1=Ss1/Y//longitudinal strain
T2=(m2*g)+T1//tension in wire 2
Ss2=T2/A//longitudinal stress
St2=Ss2/Y//longitudinal strain

```

```
printf('the longitudinal strain in wire 1 is %3.1e',
    St1)
printf('\nthe longitudinal strain in wire 2 is %3.1e
    ',St2)
```

Scilab code Exa 14.7 calculation of the density of the liquid

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.7
//calculation of the density of the liquid

//given data
h=.02*10^-2//height(in m) of the column of liquid
R=7.5*10^-3//radius(in m) of the soap bubble
S=.03//surface tension(in N/m) of the soap solution
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
deltaP=4*S/R//excess pressure inside the soap bubble
rho=deltaP/(h*g)//densiy

printf('the density of the liquid is %3.1e kg/m^3',
    rho)
```

Scilab code Exa 14.7w calculation of the longitudinal strain developed in each wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 14.7w
//calculation of the longitudinal strain developed
   in each wire

//given data
m=3//mass(in kg) of each block
A=.005*10^-4//area(in m^2) of the cross section
Y=2*10^11///Young modulus(in N/m^2) of the wire
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
//using equation of motion,
//TA = m*a...............(1)
//TB - TA = m*a.........(2)
//m*g - TB = m*a........(3)
//adding equation (2) and equation (3) and
   substituting TA from equation (1),we get
a=(m*g)/(3*m)//acceleration
TA=m*a//Tension(in N) in wire A
TB=(m*a)+TA//Tension(in N) in wire B..from equation
   (2)
StA=(TA)/(A*Y)//longitudinal strain in wire A
StB=(TB)/(A*Y)//longitudinal strain in wire B

printf('the longitudinal strain developed in wire A
   is %3.1e',StA)
printf('\nthe longitudinal strain developed in wire
   B is %3.1e',StB)
```

Scilab code Exa 14.8 calculation of the height of the water in the column

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 14.8
//calculation of the height of the water in the
   column

//given data
r=.2*10^-3//radius(in m) of the tube
S=.075//surface tension(in N/m) of the water
g=10//gravitational acceleration(in m/s^2) of the
   earth
rho=1000//density of the water(in kg/m^3)
theta=0//tube dipped vertically

//calculation
h=(2*S*cosd(theta))/(r*rho*g)//height in column

printf('the height of the water in the column is %3
   .1f cm',h*10^2)
```

Scilab code Exa 14.8w calculation of the elastic potential energy stored in the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.8w
//calculation of the elastic potential energy stored
    in the wire

//given data
A=3*10^-6//area(in m^2) of the cross section
l=50*10^-2//natural length(in m)
m=2.1//mass(in kg) hanged
Y=1.9*10^11///Young modulus(in N/m^2) of the wire
g=10//gravitational acceleration(in m/s^2) of the
   earth
```

```

//calculation
V=A*l//volume of the wire
T=m*g//tension in the wire
Ss=T/A//stress
St=Ss/Y//strain
U=(Ss*St*V/2)//elastic potential energy

printf('the elastic potential energy stored in the
    wire is %3.1e J',U)
```

Scilab code Exa 14.9 calculation of the value of the coefficient of viscosity of the solution

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.9
//calculation of the value of the coefficient of
    viscosity of the solution

//given data
d=2*10^-3//diameter(in m) of the air bubble
sigma=1750//density(in kg/m^3) of the solution
v=.35*10^-2//rate of flow(in m/s)
g=9.8//gravitational acceleration(in m/s^2) of the
    earth

//calculation
r=d/2//radius of the air bubble
//force of buoyancy is........B = (4/3)*%pi*r^3*
    sigma*g
//viscous force is............F = 6*%pi*eta*r*v
//above two forces are equal,thus we get
eta=(2*r^2*sigma*g)/(9*v)//coefficient of viscosity
```

```

printf('the value of the coefficient of viscosity of the solution is %d poise',round(eta*10))//0 1 poise = .1 N-s/m^2
```

Scilab code Exa 14.9w calculation of the elongation of the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.9w
//calculation of the elongation of the wire

//given data
W=10//weight(in N) of the block
A=3*10^-6//area(in m^2) of the cross section
r=20*10^-2//radius(in m) of the circle of rotation
v=2//speed(in m/s) of the block
Y=2*10^11///Young modulus(in N/m^2) of the wire
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
m=W/g//mass of the block
T=W+(m*v*v/r)//tension
L=r
l=(T*L)/(A*Y)//elongation

printf('the elongation of the wire is %3.1e cm',l*10^2)
```

Scilab code Exa 14.11w calculation of the amount by which the pressure inside the bubble is greater than the atmospheric pressure

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.11w
//calculation of the amount by which the pressure
   inside the bubble is greater than the atmospheric
    pressure

//given data
r=1*10^-3//radius(in m) of the air bubble
S=.075//suface tension(in N/m)
rho=1000//density(in kg/m^3) of the liquid
h=10*10^-2//depth(in m) of the bubble
g=9.8//gravitational acceleration(in m/s^2) of the
   earth

//calculation
//P = P0 +(h*rho*g)........(1)
//Pdash = P + (2*S/r)......(2)
//deltaP = Pdash - P0
deltaP=(h*rho*g)+(2*S/r)//difference in the pressure

printf('the pressure inside the bubble is greater
   than the atmospheric pressure by %d Pa',deltaP)
```

Scilab code Exa 14.12w calculation of the load W suspended from wire to keep it in equilibrium

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.12w
//calculation of the load W suspended from wire to
   keep it in equilibrium

```

```
//given data
l=10*10^-2//length(in m) of the wire
//1 dyne = 10^-5 N
S=25*10^-5*10^2//suface tension(in N/m) of the soap solution
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
F=2*l*S//force exerted by the film on the wire
m=F/g//mass of the load

printf('the load W suspended from wire to keep it in equilibrium should be %3.1e N',F)
printf('\nthe mass of the load suspended from wire to keep it in equilibrium should be %3.1e kg or %3.1f g',m,m*10^3)
```

Scilab code Exa 14.13w calculation of the radius of the capillary tube

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.13w
//calculation of the radius of the capillary tube

//given data
h=7.5*10^-2//height(in m) by which the capillary rises
S=7.5*10^-2//suface tension(in N/m) of water
theta=0//contact angle(in degree) between water and glass
g=10//gravitational acceleration(in m/s^2) of the earth
rho=1000//density(in kg/m^3) of water
```

```
//calculation
r=(2*S*cosd(theta))/(h*rho*g)//from formula of height in capillary tube

printf('the radius of the capillary tube is %3.1f mm',r*10^3)
```

Scilab code Exa 14.15w calculation of the tangential force needed to keep the plate moving

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.15w
//calculation of the tangential force needed to keep the plate moving

//given data
A=10//area(in m^2) of the plate
v=2//speed(in m/s) of the plate
d=1//depth(in m) of the river
// 1 poise = .1 N-s/m^2... unit of viscosity
eta=10^-2*10^-1//coefficient of viscosity(in N-s/m^2)

//calculation
dvbydx=v/d//velocity gradient
F=eta*dvbydx*A//force exerted

printf('the tangential force needed to keep the plate moving is %3.2f N',F)
```

Scilab code Exa 14.16w calculation of the shearing stress between the horizontal layers of water

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.16w
//calculation of the shearing stress between the
   horizontal layers of water

//given data
v=18*10^3/(60*60)//velocity(in m/s) of the water in
   river
d=5//depth(in m) of the river
// 1 poise = 0.1 N-s/m^2
eta=10^-2*10^-1//coefficient of viscosity(in N-s/m
   ^2) of the water

//calculation
dvbydx=v/d//velocity gradient
//force of viscosity ......F=eta*A*(dvbydx)
//shearing stress..........Ss=F/A
Ss=eta*(dvbydx)

printf('the shearing stress between the horizontal
   layers of water is %3.1e N/m^2',Ss)
```

Scilab code Exa 14.17w calculation of the terminal velocity of the rain drop

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 14.17w
//calculation of the terminal velocity of the rain
```

```
drop

//given data
r=.01*10^-3//radius(in m) of the drop
eta=1.8*10^-5//coefficient of viscosity(in N-s/m^2) of the air
rho=1.2//density(in kg/m^3) of the air
rho0=1000//density(in kg/m^3) of the water
g=10//gravitational acceleration(in m/s^2) of the earth

//calculation
//at terminal velocity.........6*%pi*eta*r*v = (4/3)*%pi*r^3*rho*g
v=(2*r^2*rho0*g)/(9*eta)//terminal velocity

printf('the terminal velocity of the rain drop is %3.1f cm/s^2',v*10^2)
```

Chapter 15

Wave Motion and Waves on a String

Scilab code Exa 15.1 calculation of the velocity function ft giving displacement function gx giving shape

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.1
//calculation of the velocity,function f(t) giving displacement,function g(x) giving shape

//given data
//y = y0*exp-(((t/T) - (x/lambda))^2)
y0=4*10^-3//value of y0(in m)
T=1//value of T(in s)
lambda=4*10^-2//value of lambda(in m)

//calculation
v=lambda/T//velocity of the wave
//by putting x=0 in equation (1)......f(t) = y0*exp-((t/T)^2)
//by putting t=0 in equation (1)......g(x) = y0*exp
```

```
    -((x/lambda)^2)

printf('the velocity of the wave is %d cm/s',v*10^2)
printf('\nthe function f(t) giving displacement is
    ...................f(t) = y0*exp-((t/T)^2)')
printf('\nthe function g(x) giving shape of the
    string at t=0 is.....g(x) = y0*exp-((x/lambda)^2)
    ')
```

Scilab code Exa 15.1w calculation of the amplitude wavelength frequency speed of the wave

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.1w
//calculation of the amplitude,wavelength,frequency,
    speed of the wave

//given data
//given wave equation is.....y = (3.0cm)*sin
    (6.28(.50*x - 50*t))

//calculation
//comparing with standard equation of wave....y = A*
    sin*2*%pi*((x/lambda) - (t/T)),we get
A=3//amplitude(in cm)
lambda=(1/0.50)//wavelength(in cm)
T=1/50//time period(in s)
nu=1/T//frequency(in Hz)
v=nu*lambda//wave velocity(in cm s^-1)

printf('the amplitude is %d cm',A)
printf('\nthe wavelength is %d cm',lambda)
printf('\nthe frequency is %d Hz',nu)
```

```
printf('\nthe wave velocity is %d cm/s',v)
```

Scilab code Exa 15.2 calculation of the amplitude wave number wavelength frequency time period wave velocity

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.2
//calculation of the amplitude,wave number,wavelength,frequency,time period,wave velocity

//given data
//given equation......y = (5mm)*sin((1cm^-1)*x - (60 s^-1)*t)
w=60//angular frequency

//calculation
A=5//amplitude(in cm)
k=1//wave number(in cm^-1)
lambda=(2*%pi)/k//wavelength(in cm)
nu=w/(2*%pi)//frequency(in Hz)
T=1/nu//Time period(in s)
v=nu*lambda//wave velocity(in cm/s)

printf('the amplitude is %d mm',A)
printf('\nthe wave number is %d cm^-1',k)
printf('\nthe wavelength is %3.2f cm',lambda)
printf('\nthe frequency is %3.2f Hz',nu)
printf('\nthe time period is %3.2f s',T)
printf('\nthe wave velocity is %d cm/s',v)
```

Scilab code Exa 15.2w calculation of the maximum velocity and acceleraion of the particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.2w
//calculation of the maximum velocity and acceleraion of the particle

//given data
//given wave equation is.....y = (3.0cm)*sind((3.14 cm^-1)x - (3.14 s^-1)*t))
t=0//time taken(in s)
t1=.11//time(in s) for acceleration
function yv=f(t)
    yv = (3.0)*sin(-(3.14)*t)//take x=0 (after derivative )..for maximum velocity
endfunction

//calculation
//V = dy/dt
vmax=derivative(f,t)
//vn=(-9.4)*(314)*(sin((3.14*x)+(314*t)))......take x=6(after derivative)...for acceleration at x=6 cm
a=-(2952)*sin(6*%pi-11*%pi)

printf('the maximum velocity is %3.2f m/s',vmax)
printf('\nthe acceleration at t=0.11 s and x= 6 cm is %d cm^2/s',a)
```

Scilab code Exa 15.3 calculation of the time taken by the pulse in travelling through a distance

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.3
//calculation of the time taken by the pulse in
   travelling through a distance

//given data
m=1//mass(in kg) of the block
mu=1*10^-3*10^2//mass density(in kg/m)
l=50*10^-2//disatnce(in m) travelled
g=10//gravitational acceleration(in m/s^2) of the
   earth

//calculation
F=m*g//tension in the string
v=sqrt(F/mu)//wave velocity
T=l/v//time taken

printf('the time taken by the pulse in travelling
   through a distance of 50 cm is %3.2f s',T)
```

Scilab code Exa 15.3w calculation of the speed and displacement of the particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.3w
//calculation of the speed and displacement of the
   particle

//given data
A=.80*10^-6//area(im m^2) of the string
rho=12.5*10^-3*10^6//density(in kg/m^3)
```

```
nu=20//transverse frequency(in Hz)
F=64//tension(in N)

//calculation
mu=A*1*rho//mass of 1 m of the string = linear mass density
v=sqrt(F/mu)//wave speed
w=2*%pi*nu//angular velocity
//substituting above values equation becomes.....y = (1.0cm)*cos(125*(t-(x/v)))

function y=f(t,x)
    y=1*cos(2*%pi*nu*(t-(x/v)))
endfunction
t=0.05//time taken(in s)
x=50*10^-2 //displacement(in m)
yn=f(t,x)

function yfv=ffv(t)
    yfv=1*cos(2*%pi*nu*(t-((50*10^-2)/v)))//putting value of x ..to be substituted after derivation
endfunction
vn=derivative(ffv,t)

printf('the wave speed is %d m/s',v)
printf('\nthe wave equation is .......y = (1.0cm)*cos(%d*(t-(x/%d)))',w,v)
printf('\nthe displacement of the particle at x=50 cm at time t=0.05 s is %3.2f cm',yn)
printf('\nthe velocity of the particle at that position is %d cm/s',round(vn))
```

Scilab code Exa 15.4 calculation of the power transmitted through a given point

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.4
//calculation of the power transmitted through a
   given point

//given data
P1=.20//average power(in W)
A1=2//amplitude(in mm) at this point
A2=3//amplitude(in mm)

//calculation
//transmitted power is proportional to the square of
    the amplitude
P2=P1*(A2/A1)^2

printf('the power transmitted through the given
   point is %3.2f W',P2)
```

Scilab code Exa 15.4w calculation of the extension of the wire over its natural length

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.4w
//calculation of the extension of the wire over its
   natural length

//given data
m=5*10^-3//mass(in kg) of the wire
L=50*10^-2//length(in cm) of the wire
v=80//speed(in m/s) of the wave
Y=16*10^11//Young modulus(in N/m^2)
```

```
A=1*10^-6//area(in m^2) of cross section of the wire

//calculation
mu=m/L//linear mass density
F=mu*v^2//tension in the wire
deltaL=(F*L)/(A*Y)//extension in the length of wire

printf('the extension of the wire over its natural
    length is %3.2f mm',deltaL*10^3)
```

Scilab code Exa 15.5 calculation of the phase difference between the waves and amplitude of the resultant wave

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.5
//calculation of the phase difference between the
    waves and amplitude of the resultant wave

//given data
//equations of the wave are
//y1 = A1*sin(k(x-v*t))...........(1)
//y2 = A2*sin(k(x-v*t+x0))........(2)
k=6.28*10^2//wave number(in m^-1)
x0=1.50*10^-2//value of x0(in m)
A1=5*10^-3//amplitude(in m) of wave 1
A2=4*10^-3//amplitude(in m) of wave 2

//calculation
deltaP=k*x0//phase difference
deltaA=abs(A1-A2)//amplitude of the wave

printf('the phase difference between the waves is %3
    .2f rad',deltaP)
```

```
printf('\nthe amplitude of the resultant wave is %3
    .1f mm',deltaA*10^3)
```

Scilab code Exa 15.5w calculation of the wavelength of the pulse when it reaches the top of the rope

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.5w
//calculation of the wavelength of the pulse when it
     reaches the top of the rope

//given data
lr=12//length(in m) of the rope
mr=6//mass(in kg) of the rope
mb=2//mass(in kg) of the block
lambda=.06//wavelength(in m) of the wave produced at
     the lower end

//calculation
//from equation .......v = nu*lambda
//putting v = sqrt(F/lambda)....we get
//sqrt(F/lambda) = nu*sqrt(mu)....using this
    equation,we get
lambda1=lambda*sqrt((mr+mb)/mb)

printf('the wavelength of the pulse when it reaches
    the top of the rope is %3.2f m',lambda1)
```

Scilab code Exa 15.6 calculation of the velocity node closest to origin antinode closest to origin amplitude at x

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.6
//calculation of the velocity ,node closest to origin
   ,antinode closest to origin ,amplitude at x

//given data
//equation of the wave is......y = A*cosd(k*x)*sind(
   w*t)
A=1//amplitude(in mm)
k=1.57//value of k(in cm^-1)
w=78.5//angular velocity(in s^-1)
x=2.33//value of x(in cm)

//calculation
v=w/k//wave velocity
xn=%pi/(2*k)//for a node ...cosd(kx) = 0
xa=%pi/k//for a antinode ...|cosd(kx)| = 1
Ar=A*abs(cos(k*x))

printf('the velocity of the wave is %d cm/s',v)
printf('\nthe node closest to the origin is located
   at x=%d cm',xn)
printf('\nthe antinode closest to the origin is
   located at x=%d cm',xa)
printf('\nthe amplitude at x=2.33 is %3.2f mm',Ar)
```

Scilab code Exa 15.6w calculation of the displacement of the particle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.6w
//calculation of the displacement of the particle
```

```

//given data
//given equations are
//y1 = (1.0 cm)*sin((3.14 cm-1)*x - (157 s^-1)*t)...........(1)
//y2 = (1.5 cm)*sin((1.57 cm-1)*x - (314 s^-1)*t)...........(2)

//calculation
function y1=f1(t,x)
    y1=1*sin((3.14*x)-(157*t))
endfunction

function y2=f2(t,x)
    y2=1.5*sin((1.57*x)-(314*t))
endfunction

x=4.5//given value of x(in cm)
t=5*10^-3//given value of t(in s)
//y = y1 + y2.......net displacement
y=f1(t,x)+f2(t,x)

printf('the displacement of the particle at x=4.5 cm and t=5.0 ms is %3.2f cm',y)
```

Scilab code Exa 15.7 calculation of the fundamental frequency of the portion of the string between the wall and the pulley

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.7
//calculation of the fundamental frequency of the portion of the string between the wall and the pulley
```

```
//given data
m=1.6//mass(in kg) of the load
mw=20*10^-3//mass(in kg) of the wire
l=50*10^-2//length(in kg/m) of wire
g=10//gravitational acceleration(in m/s^2) of the earth
L=40*10^-2//length(in m) of the string between the wall and the pulley

//calculation
F=m*g//tension in the string
mu=mw/l//linear mass density
nu0=(1/(2*L))*sqrt(F/mu)//fundamental frequency

printf('the fundamental frequency of the portion of the string between the wall and the pulley is %d Hz',nu0)
```

Scilab code Exa 15.7w calculation of the maximum displacement wavelengths and wave speed velocity nodes and antinodes number of loops

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.7w
//calculation of the maximum displacement, wavelengths and wave speed,velocity,nodes and antinodes,number of loops

//given data
//given equation is ....y = (5.00 mm)*sin(1.57 cm^-1)*sin((314 s^-1)*t)

//calculation
```

```
//at x=5.66 cm
A=(5*10^-3)*sin(1.57*5.66)//amplitude
k=1.57//value of k(in cm^-1)
w=314//angular frequency(in s^-1)
lambda=(2*%pi)/k//wavelength
nu=(w)/(2*%pi)//frequency
//v = dy/dt = (157 cm/s)*sin(1.57 cm^-1*x)*cos((314 s^-1)*t)
function v=f(t,x)
    v=157*sin(1.57*x)*cos((314)*t)
endfunction
x=5.66//value of x (in cm)
t=2//value of t (in s)
vn=f(t,x)//velocity of the particle

//for nodes......sin(1.57 cm^-1)*x = 0..........gives x=2*n
//since l=10 cm..nodes occur at 0 cm,2 cm,4 cm,6 cm,8 cm,10 cm
//antinodes occur in between at 1 cm,3 cm,5 cm,7 cm,9 cm
nloops=10*(1/2)

printf('the amplitude is %3.2f mm',10^3*A)
printf('\nthe wavelength is %3.2f cm',lambda)
printf('\nthe velocity is %3.2f cm/s',vn)// Textbook Correction : correct answer is 76.48 cm/s
printf('\nnodes occur at 0 cm,2 cm,4 cm,6 cm,8 cm,10 cm')
printf('\nantinodes occur in between at 1 cm,3 cm,5 cm,7 cm,9 cm')
printf('\nthe number of loops is %d',nloops)
```

Scilab code Exa 15.8 calculation of the length of the experimental wire to get the resonance

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.8
//calculation of the length of the experimental wire to get the resonance

//given data
nu1=256//frequency(in Hz) of the tunning fork 1
nu2=384//frequency(in Hz) of the tunning fork 2
l1=21//length(in cm) of the wire for tunning fork 1

//calculation
l2=(nu1/nu2)*l1//law of length

printf('the length of the experimental wire to get the resonance is %d cm',l2)
```

Scilab code Exa 15.8w calculation of the pressing in the guitar to produce required fundamental frequency

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.8w
//calculation of the pressing in the guitar to produce required fundamental frequency

//given data
L1=90//length(in cm) of the guitar string
nu1=124//fundamental frequency(in Hz) for L1
nu2=186//required fundamental frequency(in Hz)

//calculation
//from equation of fundamental frequency....nu =
```

```
(1/(2*L))*sqrt(F/mu)
L2=L1*(nu1/nu2)

printf('the pressing in the guitar to produce the fundamental frequency of 186 Hz is %d cm',L2)
```

Scilab code Exa 15.9w calculation of the position of bridges in sonometer wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.9w
//calculation of the position of bridges in sonometer wire

//given data
//nu1 : nu2 : nu3 = 1 : 2 : 3
L=1//length(in m) of the sonometer wire
m1=1//taking value from ratio
m2=2//taking value from ratio
m3=3//taking value from ratio

//calculation
//from formula of fundamental frequency.....nu = (1/(2*L))*sqrt(F/mu)
L1=L/((1/m1)+(1/m2)+(1/m3))//position of bridge 1 from one end
L2=L1/2
L3=L1/3//position of bridge 2 from the other end

printf('the position of bridge 1 from one end is %3.2f m',L1)
printf('\nthe position of bridge 2 from the other end is %3.2f m',L3)
```

Scilab code Exa 15.10w calculation of the length of the wire

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 15.10w
//calculation of the length of the wire

//given data
mu=5*10^-3//mass density(in kg/m) of the wire
F=450//tension(in N) produced in the wire
nu1=420//frequency(in Hz) of nth harmonic
nu2=490//frequency(in Hz) of (n+1)th harmonic

//calculation
//from formula of fundamental frequency.....nu =
    (1/(2*L))*sqrt(F/mu)......(1)
n=nu1/(nu2-nu1)//value of n
L=(n/(2*nu1))*sqrt(F/mu)//erom equation (1)

printf('the length of the wire is %3.1f m',L)
```

Chapter 16

Sound Waves

Scilab code Exa 16.1 calculation of the audibility of a wave

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.1
//calculation of the audibility of a wave

//given data
v=300//velocity(in m/s) of the wave
lambda=.60*10^-2//wavelength(in m) of the wave

//calculation
nu=v/lambda//frequency of the wave
if(nu<20)
    printf('the wave is not audible')
elseif(nu>20000)
    printf('the wave is not audible')
else
    printf('the wave is audible')
end
```

Scilab code Exa 16.1w calculation of the depth of the sea and wavelength of the signal in the water

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.1w
//calculation of the depth of the sea and wavelength
    of the signal in the water

//given data
nu=50*10^3//frequency(in Hz) of the given signal
t=0.8//time(in s)requires for reflected wave to
   return
v=1500//speed(in m/s) of the sound in water

//calculation
d=v*t/2//depth of the sea
lambda=v/nu//wavelength in water

printf('the depth of the sea is %d m',d)
printf('\nthe wavelength of the signal in the water
   is %3.1f cm',lambda*10^2)
```

Scilab code Exa 16.2 calculation of the amplitude of vibration of the particles of the medium

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.2
```

```
//calculation of the amplitude of vibration of the
   particles of the medium

//given data
lambda=40*10^-2//wavelength(in m) of the wave
deltap=1*10^-3//difference between the minimum and
   the maximum pressure(in N/m^2)
B=1.4*10^5//Bulk modulus(in N/m^2)

//calculation
p0=deltap/2//pressure amplitude
s0=(p0*lambda)/(2*%pi*B)//from equation of Bulk
   modulus

printf('the amplitude of vibration of the particles
   of the medium is %3.2e m',s0)
```

Scilab code Exa 16.2w calculation of the location of the plane

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.2w
//calculation of the location of the plane

//given data
v=510*10^3/(60*60)//speed(in m/s) of the plane
h=2000//height(in m) of the plane
vs=340//speed(in m.s) of the sound in air

//calculation
t=h/vs//time taken by the sound to reach the
   observer
d=v*t//location of the plane

```

```
printf('the plane will be %d m ahead of the observer
    on its line of motion',d)
```

Scilab code Exa 16.3 calculation of the intensity of the sound wave

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.3
//calculation of the intensity of the sound wave

//given data
p0=2*10^-2//pressure amplitue(in N/m^2)
p0dash=2.5*10^-2//new pressure amplitue(in N/m^2)
I=5.0*10^-7//intensity(in W/m^2) of the wave

//calculation
//intensity of the wave is proportional to square of
    the pressure amplituide
Idash=I*((p0dash/p0)^2)

printf('the intensity of the sound wave is %3.1e W/m
   ^2',Idash)
```

Scilab code Exa 16.3w calculation of the frequency wavelength speed maximum and minimum pressures of the sound wave

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.3w
//calculation of the frequency,wavelength,speed,
   maximum and minimum pressures of the sound wave
```

```
//given data
//equation of the sound wave is
//p = (0.01 N/m^2)*sin((1000 s^-1)*t - (3.0 m^-1)*x)............(1)
peq=1.0*10^5//equilibrium pressure(in N/m^2) of the air

//calculation
//comparing equation (1) with standard equation p = p0*sin(w*(t-(x/v)))...we get
w=1000//value of w(in s^-1)
nu=w/(2*%pi)//frequency
v=w/3//velocity
lambda=v/nu//wavelength
p0=0.01//pressure amplitude(in N/m^2)

printf('the frequency is %d Hz',nu)
printf('\nthe wavelength is %3.1f m',lambda)
printf('\nthe speed of the sound wave is %d m/s',v)
printf('\nthe maximum pressure amplitude is (%3.2e + %3.2f) N/m^2',peq,p0)
printf('\nthe minimum pressure amplitude is (%3.2e - %3.2f) N/m^2',peq,p0)
```

Scilab code Exa 16.4 calculation of the increase in the sound level in decibels

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.4
//calculation of the increase in the sound level in decibels

```

```
//given data
r=20//intensity is increase by r factor

//calculation
//using the equation.....beta = 10*log(I/I0)...we get
deltabeta=10*log10(r)//increase in sound level

printf('the increase in the sound level in decibels is %d dB',deltabeta)
```

Scilab code Exa 16.4w calculation of the minimum separation between the two points for a given phase difference

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.4w
//calculation of the minimum separation between the two points for a given phase difference

//given data
nu=10*10^3//frequency(in Hz) of the sound wave
v=340//speed(in m/s) of the wave
delta=60//phase difference(in degree)

//calculation
lambda=v/nu//wavelength
k=2*%pi/lambda//wave number
d=(delta*%pi/180)/k

printf('the minimum separation between the two points for phase difference of 60 degree is %3.2f cm',d*10^2)
```

Scilab code Exa 16.5 calculation of the nature of interference

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.5
//calculation of the nature of interference

//given data
nu=1*10^3//frequency(in Hz) of the source
deltax=83*10^-2//difference in the length(in m) of
    paths
v=332//speed(in m/s) of the sound in air

//calculation
lambda=v/nu//wavelength
delta=(2*%pi/lambda)*deltax
n=delta/%pi//phase difference is 'n' multiple of pi
if(modulo(n,2)==0)
    printf('the waves will interfere constructively.
        ')//for even values of 'n'
else
    printf('the waves will interfere destructively.'
        )//for odd values of 'n'
end
```

Scilab code Exa 16.5w calculation of the atmospheric temperature

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.5w
```

```
//calculation of the atmospheric temperature

//given data
v1=336//speed(in m/s) travelled by the sound
v0=332//speed(in m/s) of the sound at O degreecelsius
T0=0+273//temperature(in kelvin)

//calculation
T=((v1/v0)^2)*T0//temperature (in kelvin)
t=T-273//temperature(in degreecelsius)

printf('the atmospheric temperature is %d degreecelsius',round(t))
```

Scilab code Exa 16.6 calculation of the distance of the piston from the open end for tube to vibrate in its first overtone

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.6
//calculation of the distance of the piston from the open end,for tube to vibrate in its first overtone

//given data
nu=416//frequency(in Hz) of the tunning fork
v=333//speed(in m/s) of the sound in air

//calculation
lambda=v/nu//wavelength
L=3*lambda/4//length of the tube

printf('the distance of the piston from the open end
```

```
    , for tube to vibrate in its first overtone is %3
    .1 f cm',L*10^2)
```

Scilab code Exa 16.6w calculation of the speed of sound wave in hydrogen

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.6w
//calculation of the speed of sound wave in hydrogen

//given data
gama=1.4//value of constant gama for hydrogen
voxygen=470//speed(in m/s) of the sound wave in
    oxygen

//calculation
//speed of sound wave in a gas is ........v = sqrt(
    gama*P/rho)
//at STP ,density of oxygen is 16 times density of
    hydrogen
vhydrogen=voxygen*sqrt(16)//speed of sound in
    hydrogen

printf('the speed of sound wave in hydrogen is %d m/
    s',vhydrogen)
```

Scilab code Exa 16.7 calculation of the tunning frequency of fork B

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.7
```

```
//calculation of the tunning frequency of fork B

//given data
nu1=384//tunning frequency(in Hz) of fork A
n=6//number of beats
t=2//time(in s) taken by the beats

//calculation
deltanu=n/t//frequency of beats
nu2=nu1+deltanu//frequency of fork B
nu2dash=nu1-deltanu//another frequency of fork B

printf('the tunning frequency of fork B is %d Hz or
    %d Hz',nu2dash,nu2)
```

Scilab code Exa 16.7w calculation of the energy delivered to the microphone

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.7w
//calculation of the energy delivered to the
    microphone

//given data
A=.80*10^-4//area(in m^2) of the cross section
U=3//power(in W0 output of the speaker
d=2//distance(in m) between the microphone and the
    speaker
t=5//time(in s) taken

//calculation
U0=A*U/(4*%pi*d^2)//energy falling on the microphone
     in 1 s
```

```
Udash=U0*t//energy falling on the microphone in t s

printf('the energy delivered to the microphone in t=5 s is %d microJ',round(Udash*10^6))
```

Scilab code Exa 16.8 calculation of the most dominant frequency

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.8
//calculation of the most dominant frequency

//given data
us=36*10^3/(60*60)//speed(in m/s) of the train
nudash=12*10^3//frequency(in Hz) detected by the detector
v=340//velocity(in m/s) of the sound in air

//calculation
//frequency detected is ......nudash = (v*nu0)/(v-us)
nu0=(1-(us/v))*nudash//required frequency

printf('the most dominant frequency is %3.1f kHz',nu0*10^-3)
```

Scilab code Exa 16.8w calculation of the amplitude of vibration of the particles of the air

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 16.8w
//calculation of the amplitude of vibration of the particles of the air

//given data
I=2*10^-6//intensity(in W/m^2) of the sound wave
nu=1*10^3//frequency(in Hz) of the sound wave
rho0=1.2//density(in kg/m^3) of the air
v=330//speed(in m/s) of the sound in the air

//calculation
s0=sqrt(I/(2*%pi^2*nu^2*rho0*v))//equation of displacement amplitide

printf('the amplitude of vibration of the particles of the air is %3.1e m',s0)
```

Scilab code Exa 16.9w calculation of the factor by which the pressure amplituide increases

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.9w
//calculation of the factor by which the pressure amplituide increases

//given data
n=30//increase(in dB) of the sound level

//calculation
//m = I2/I1 = intensity ratio
m=10^(n/10)
//since p2/p1 = sqrt(I2/I1)
f=sqrt(m)//require factor
```

```

printf('the factor by which the pressure amplituide
    increases is %d',round(f))
```

Scilab code Exa 16.10w calculation of the frequency at which the maxima of intensity are detected

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.10w
//calculation of the frequency at which the maxima
    of intensity are detected

//given data
r=20*10^-2//radius(in m) of the semicircular part
v=340//speed(in m/s) of the sound in air

//calculation
l1=2*r//straight distance
l2=%pi*r//curve distance
deltal=l2-l1
nu=v/deltal

printf('the frequency at which the maxima of
    intensity are detected are %d Hz and %d Hz',nu,2*
    nu)
```

Scilab code Exa 16.11w calculation of the minimum distance between the source and the detector for maximum sound detection

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 16.11w
//calculation of the minimum distance between the
    source and the detector for maximum sound
    detection

//given data
nu=180//frequency(in Hz)
d=2//distance(in m)
v=360//speed(in m/s) of the sound wave in air

//calculation
//path difference.....delta = (2*((2^2) + (x^2/4))
    ^(1/2)) - (x)
lambda=v/nu//wavelength
delta=lambda
//solving the above equation,we get
x=4-1

printf('the minimum distance between the source and
    the detector for maximum sound detection is %d m'
    ,x)
```

Scilab code Exa 16.12w calculation of the length of the shortest closed organ pipe that will resonate with the tunning fork

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.12w
//calculation of the length of the shortest closed
    organ pipe that will resonate with the tunning
    fork

//given data
```

```
nu=264//frequency(in Hz)of the tunning fork
v=350//speed(in m/s) of the sound in air

//calculation
//from the equation of the resonate frequency of the
    closed organ pipe.... l = (n*v)/(4*nu)
n=1//for l to be minimum
lmin=(v)/(4*nu)//equation of the resonate frequency
   of the closed organ pipe

printf('the length of the shortest closed organ pipe
    that will resonate with the tunning fork is %d
   cm',lmin*10^2)
```

Scilab code Exa 16.13w calculation of the length of the closed pipe

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.13w
//calculation of the length of the closed pipe

//given data
l0=60*10^-2//length(in m) of the open pipe

//calculation
//from the equation of the resonate frequency of the
    closed organ pipe.... l=(n*v)/(4*nu)
l1=l0/4

printf('the length of the closed pipe is %d cm',l1
   *10^2)
```

Scilab code Exa 16.14w calculation of the speed of the sound in air

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.14w
//calculation of the speed of the sound in air

//given data
nu=800//frequency(in Hz) of the tunning fork
l1=9.75*10^-2//distance(in m) where resonance is observed
l2=31.25*10^-2//distance(in m) where resonance is observed
l3=52.75*10^-2//distance(in m) where resonance is observed

//calculation
//from the equation of the resonate frequency ....l = (n*v)/(4*nu)
//(n*v)/(4*l1) = nu...................(1)
//((n+2)*v)/(4*l2) = nu...............(2)
//((n+4)*v)/(4*l3) = nu...............(3)
//form above equations ,we get
v=2*nu*(l2-l1)

printf('the speed of the sound in air is %d m/s',v)
```

Scilab code Exa 16.15w calculation of the fundamental frequency if the air is replaced by hydrogen

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.15w
```

```
//calculation of the fundamental frequency if the
   air is replaced by hydrogen

//given data
nu0=500//fundamental frequency(in Hz)
rhoa=1.20//density(in kg/m^3) of air
rhoh=0.089//density(in kg/m^3) of hydrogen

//calculation
//fundamental frequency of an organ pipe is
   proportional to the speed of the sound
nu=nu0*sqrt(rhoa/rhoh)

printf('the fundamental frequency if the air is
   replaced by hydrogen is %d Hz',nu)
```

Scilab code Exa 16.16w calculation of the speed wavelength in the rod frequency wavelength in the air

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.16w
//calculation of the speed,wavelength in the rod,
   frequency,wavelength in the air

//given data
l=90*10^-2//length(in m) of the rod
rho=2600//density(in kg/m^3) of the aluminium
Y=7.80*10^10//Young modulus(in N/m^2)
vai=340//speed(in m/s) of the sound in the air

//calculation
v=sqrt(Y/rho)//speed of the sound in aluminium
lambda=2*l//wavelength....since rod vibrates with
```

```
   fundamental frequency
nu=v/lambda//frequency
lambdaai=vai/nu//wavelength in the air

printf('the speed of the sound in aluminium is %d m/
   s',v)//Textbook Correction : correct answer is
   5477 m/s
printf('\nthe wavelength of the sound in aluminium
   rod is %d cm',lambda*10^2)
printf('\nthe frequency of the sound produced is %d
   Hz',nu)//Textbook Correction : correct answer is
   3042 Hz
printf('\nthe wavelength of the sound in air is %3.1
   f cm',lambdaai*10^2)
```

Scilab code Exa 16.17w calculation of the frequency of the note emitted by the taut string

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.17w
//calculation of the frequency of the note emitted
   by the taut string

//given data
nu1=440//frequency(in Hz) of the string
n=4//number of beats per second
nuf=440//tunning frequency(in Hz) of the fork

//calculation
fre=nuf+n//required frequncy

printf('the frequency of the note emitted by the
   taut string is %d Hz',fre)
```

Scilab code Exa 16.18w calculation of the apparent frequency

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.18w
//calculation of the apparent frequency

//given data
us=36*10^3/(60*60)//speed(in m/s)of the car
v=340//speed(in m/s) of the sound in the air
nu=500//frequency(in Hz)

//calculation
nudash=(v/(v+us))*nu//apparent frequency  heard by
    the observer
nudashdash=(v/(v-us))*nu//frequency received by the
    wall

printf('the apparent frequency heard by the ground
    observer is %d Hz',round(nudash))
printf('\nthe frequency of the reflected wave as
    heard by the ground observer is %d Hz',nudashdash
    )
```

Scilab code Exa 16.19w calculation of the frequency of the whistle of the train

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 16.19w
//calculation of the frequency of the whistle of the
    train

//given data
us=72*10^3/(60*60)//speed(in m/s) of the train 1
u0=54*10^3/(60*60)//speed(in m/s) of the train 2
nu=600//frequency(in Hz) of the whistle
v=340//speed(in m/s)of sound in the air

//calculation
nudash=((v+u0)/(v-us))*nu//frequency heard by the
    observer before the meeting of the trains
nudashdash=((v-u0)/(v+us))*nu//frequency heard by
    the observer after the crossing of the trains

printf('the frequency heard by the observer before
    the meeting of the trains is %d Hz',round(nudash)
    )
printf('\nthe frequency heard by the observer after
    the crossing of the trains is %d Hz',round(
    nudashdash))
```

Scilab code Exa 16.20w calculation of the main frequency heard by the person

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.20w
//calculation of the main frequency heard by the
    person

//given data
us=36*10^3/(60*60)//speed(in m/s) of the person on
```

```
the scooter
v=340//speed(in m/s) of sound in the air
nu=600//frequency(in Hz) of the siren

//calculation
nudash=(v/(v+us))*nu//main frequency

printf('the main frequency heard by the person is %d Hz',round(nudash))
```

Scilab code Exa 16.21w calculation of the original frequency of the source

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.21w
//calculation of the original frequency of the source

//given data
u0=10//speed(in m/s) of the observer going away from the source
us=10//speed(in m/s) of the source going away from observer
nudash=1950//frequency(in Hz) of the sound detected by the detector
v=340//speed(in m/s) of the sound in the air

//calculation
nu=((v+us)/(v-u0))*nudash//original frequency

printf('the original frequency of the source is %d Hz',round(nu))
```

Scilab code Exa 16.22w calculation of the speed of the car

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 16.22w
//calculation of the speed of the car

//given data
nudash=440//frequency(in Hz) emitted by the wall
nudashdash=480//frequency(in Hz) heard by the car driver
v=330//speed(in m/s) of the sound in the air

//calculation
//frequency received by the wall..............nudash = (v/(v-u))*nu............(1)
//frequency(in Hz) heard by the car driver....nudashdash = ((v+u)/v)*nudash....(2)
//from above two equations,we get
u=((nudashdash-nudash)/(nudashdash+nudash))*v//speed of the car

printf('the speed of the car is %3.1f m/s or %d km/h',u,round(u*10^-3*60*60))
```

Scilab code Exa 16.23w calculation of the frequency of train whistle heard by the person standing on the road perpendicular to the track

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
```

```
//example 16.23w
//calculation of the frequency of train whistle
   heard by the person standing on the road
   perpendicular to the track

//given data
v=340//speed(in m/s) of the sound in the air
d1=300//distance(in m) of the train from the
   crossing
u=120*10^3/(60*60)//speed(in m/s) of the train
nu=640//frequency(in Hz) of the whistle
d2=400//distance(in m) of the person from the
   crossing ,perpendicular to the track

//calculation
theta=acosd(d1/sqrt(d1^2+d2^2))//pythagoras theorem
nudash=(v/(v-(u*cosd(theta))))*nu//frequency of the
   whistle heard

printf('the frequency of train whistle heard by the
   person standing on the road perpendicular to the
   track is %d Hz',nudash)
```

Chapter 17

Light Waves

Scilab code Exa 17.1 calculation of the speed of light in glass

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.1
//calculation of the speed of light in glass

//given data
mu=1.5//refractive index of glass
v0=3*10^8//speed(in m/s) of light in vacuum

//calculation
v=v0/mu//definition of the refractive index

printf('the speed of light in glass is %3.1e m/s',v)
```

Scilab code Exa 17.1w calculation of the limits of wavelengths in the water

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.1w
//calculation of the limits of wavelengths in the
    water

//given data
lambda01=400//mimimum wavelength(in nm) of the light
     used
lambda02=700//maximum wavelength(in nm) of the light
     used
mu=1.33//refractive index of water

//calculation
lambda1=lambda01/mu//definition of the refractive
    index
lambda2=lambda02/mu//definition of the refractive
    index

printf('the limits of wavelengths in the water are
    %d nm and %d nm',lambda1,lambda2)
```

Scilab code Exa 17.2 calculation of the separation between successive bright fringes

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.2
//calculation of the separation between successive
    bright fringes

//given data
d=0.10*10^-3//separation(in m) between the slits
```

```
lambda=600*10^-9//wavelength(in m) of the light used
D=1//separation(in m) between the slits and the
    screen

//calculation
w=D*lambda/d//separation between successive bright
    fringes

printf('the separation between successive bright
    fringes is %3.1e m or %3.1f mm',w,w*10^3)
```

Scilab code Exa 17.2w calculation of the refractive index of the glass

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.2w
//calculation of the refractive index of the glass

//given data
x1=2//distance(in cm)travelled through the glass
x2=2.25//distance(in cm)travelled through the water
muw=1.33//refractive index of water

//calculation
//for 'x' distance travelled through a medium of
    refractive index 'mu',the optical path is 'mu*x'
mug=muw*x2/x1//refractive index of glass

printf('the refractive index of the glass is %3.2f',
    mug)
```

Scilab code Exa 17.3 calculation of the wavelength of light in the water

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.3
//calculation of the wavelength of light in the
   water

//given data
lambdan=589//wavelength(in nm) of light in vacuum
mu=1.33//refractive index of water

//calculation
lambda=lambdan/mu//definition of the refractive
   index

printf('the wavelength of light in the water is %d
   nm',round(lambda))
```

Scilab code Exa 17.3w calculation of the wavelengths of the violet and the red lightcalculation of the wavelengths of the violet and the red light

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.3w
//calculation of the wavelengths of the violet and
   the red light

//given data
D=2.5//separation(in m) between the slit and the
   screen
d=0.5*10^-3//separation(in m) between the slits
yv=2*10^-3//distance(in m) between the central white
    fringe and the first violet fringe
yr=3.5*10^-3//distance(in m) between the central
```

```
    white fringe and the first red fringe

//calculation
lambdav=yv*d/D//wavelength of the violet light
lambdar=yr*d/D//wavelength of the red light

printf('the wavelength of the violet light is %d nm'
    ,lambdav*10^9)
printf('\nthe wavelength of the red light is %d nm',
    lambdar*10^9)
```

Scilab code Exa 17.4 calculation of the minimum thickness of the film

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.4
//calculation of the minimum thickness of the film

//given data
lambda=589//wavelength(in nm) of the light used
mu=1.25//refractive index of the material

//calculation
//for strong reflection......2*mu*d = lambda/2
d=lambda/(4*mu)//minimum thickness

printf('the minimum thickness of the film is %d nm',
    round(d))
```

Scilab code Exa 17.4w calculation of the separation between the slits

```
//developed in windows XP operating system 32bit
```

```
//platform Scilab 5.4.1
clc;clear;
//example 17.4w
//calculation of the separation between the slits

//given data
lambda=589.3*10^-9//wavelength(in m) of the sodium light
D=100*10^-2//separation(in m) between the slit and the screen
n=10//number of the bright fringe
x=12*10^-3//distance(in m) between the central maximum and the tenth bright fringe

//calculation
d=n*lambda*D/x//separation between the slits

printf('the separation between the slits is %3.1e m or %3.2f mm',d,d*10^3)
```

Scilab code Exa 17.5 calculation of the angular divergence for most of the light getting diffracted

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.5
//calculation of the angular divergence for most of the light getting diffracted

//given data
lambda=450*10^-9//wavelength(in m) of the light used
b=0.2*10^-3//width(in m) of the slit

//calculation
```

```
//for theta tends to zero......sin(theta) = theta
theta1=lambda/b//angle of minima
theta2=-lambda/b//angle of minima
theta=theta1-theta2//angular divergence

printf('the angular divergence for most of the light getting diffracted is %3.1e radian',theta)
```

Scilab code Exa 17.5w calculation of the ratio of maximum intensity to the minimum intensity in the interference fringe pattern

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.5w
//calculation of the ratio of maximum intensity to the minimum intensity in the interference fringe pattern

//given data
//intensity of the light coming from one slit in Young's double slit experiment is double the intensity of the light coming from the other slit
n=2

//calculation
r=((sqrt(n)+1)^2)/((sqrt(n)-1)^2)//required ratio

printf('the ratio of maximum intensity to the minimum intensity in the interference fringe pattern is %d',round(r))
```

Scilab code Exa 17.6 calculation of the diameter of the disc image

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.6
//calculation of the diameter of the disc image

//given data
lambda=590*10^-9//wavelength(in m) of the light used
b=10*10^-2//diameter(in m) of the converging lens used
d=20//distance(in m) between the lens and the point of focus

//calculation
sintheta=1.22*lambda/b//angular radius
r=d*sintheta//radius of the disc image
d=2*r//diameter of the disc image

printf('the diameter of the disc image is %3.1e cm',d)
```

Scilab code Exa 17.6w calculation of the ratio of maximum intensity to the minimum intensity in the interference pattern

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.6w
//calculation of the ratio of maximum intensity to the minimum intensity in the interference pattern

//given data
//width of one slit in Young's double slit experiment is double that of the other
n=2
```

```

//calculation
r=((n+1)^2)/((n-1)^2)//required ratio

printf('the ratio of maximum intensity to the minimum intensity in the interference pattern is %d',r)
```

Scilab code Exa 17.7w calculation of the maximum and the minimum path difference at the detector

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.7w
//calculation of the maximum and the minimum path difference at the detector

//given data
lambda=600*10^-9//wavelength(in m) of the light
d=1*10^-2*10^-2//distance(in m) between the sources

//calculation
pdmax=d//path diffrence maximum
pdmin=0//path diffrence minimum
//farthest minima occurs for path difference lambda/2
//sqrt(D^2 + d^2) - D = lambda/2
D=(d^2/lambda)-(lambda/4)//distance of the farthest minima

printf('the maximum path difference on moving the detector along S1P line is %3.1e cm',pdmax*10^2)
printf('\nthe minimum path difference on moving the detector along S1P line is %3.1f cm',pdmin*10^2)
```

```
printf('\nthe farthest minimum is located at a distance of %3.1f cm from the point S1',D*10^2)
```

Scilab code Exa 17.8w calculation of the distance of bright fringe from the central maximum

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.8w
//calculation of the distance of bright fringe from the central maximum

//given data
lambda1=6500*10^-10//wavelength(in m) of the light beam1
lambda2=5200*10^-10//wavelength(in m) of the light beam2
d=2.0*10^-3//separation(in m) between the slits
D=120*10^-2//separation(in m) between the slits and the screen
n=3//number of the bright fringe

//calculation
y=n*lambda1*D/d//the distance of bright fringe from the central maximum
//from the equation of the distance of bright fringe from the central maximum.....y=n*lambda*D/d
//let m th bright fringe of beam 1 coincides with n th bright fringe of beam 2
//ym = yn
//m : n = 4 : 5.....is their minimum integral ratio
m=4
ym=m*lambda1*D/d//least distance from the central maximum where both wavelengths coincides
```

```
printf('the distance of the third bright fringe from
    the central maximum is %3.2f cm',y*10^2)
printf('\nthe least distance from the central
   maximum where both the wavelengths coincides is
   %3.2f cm',ym*10^2)
```

Scilab code Exa 17.9w calculation of the number of fringes that will shift due to introduction of the sheet

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.9w
//calculation of the number of fringes that will
   shift due to introduction of the sheet

//given data
lambda=600*10^-9//wavelength(in m) of the light used
t=1.8*10^-5//thickness(in m) of the transparent
   sheet
mu=1.6//refractive index of the material

//calculation
n=((mu-1)*t)/lambda//number of fringes shifted

printf('the number of fringes that will shift due to
    introduction of the sheet is %d',n)
```

Scilab code Exa 17.10w calculation of the wavelengths in the visible region that are strongly reflected

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 17.10w
//calculation of the wavelengths in the visible region that are strongly reflected

//given data
d=.5*10^-6//thickness(in m) of the glass plate
mu=1.5//refractive index of the medium
lambda1=400*10^-9//minimum wavelength(in m) of the visible region
lambda2=700*10^-9//maximum wavelength(in m) of the visible region

//calculation
//condition for strong reflection of light of wavelength lambda is
//2*mu*d = (n + (1/2))*lambda............(1)
n1=round((2*mu*d/lambda1)-(1/2))//integral value of n for lambda1
n2=round((2*mu*d/lambda2)-(1/2))//integral value of n for lambda2
lambda1n=(2*mu*d)/(n1+(1/2))//from equation (1)
lambda2n=(2*mu*d)/(n2+(1/2))//from equation (1)

printf('the wavelengths in the visible region that are strongly reflected are %d nm and %d nm',round(lambda1n*10^9),round(lambda2n*10^9))
```

Scilab code Exa 17.11w calculation of the distance between the two first order minima

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 17.11w
```

```
//calculation of the distance between the two first order minima

//given data
b=.40*10^-3//width(in m) of the slit
D=40*10^-2//separation(in m) between the slit and the screen
lambda=546*10^-9//wavelength(in m) of the light used

//calculation
//linear distances from the central maxima are given by..x = D*tan(theta) = D*sin(theta) = +-lambda*D/b
sep=2*lambda*D/b//separation between the minima

printf('the distance between the two first order minima is %3.1f mm',sep*10^3)
```

Chapter 18

Geometrical Optics

Scilab code Exa 18.1 calculation of position of the image of an object placed at a distance from the mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.1
//calculation of position of the image of an object
   placed at a distance from the mirror.

//given data
u=-12; //object distance(in cm)
R=20; //radius of curvature of the mirror (in cm)

//calculation
v=1/((2/R)-(1/u)); //mirror formula

if(v>0)
    disp(v,'virtual image is formed on right side of
        mirror at a distance(in cm)');
else
    disp(v,'real image is formed on left side of
       mirror at a distance(in cm)');
```

```
end
```

Scilab code Exa 18.1w calculation of position and nature of the image of an object placed at a distance from a concave mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.1w
//calculation of position and nature of the image of an object placed at a distance from a concave mirror

//given data
u=-8; //object distance(in cm)
f=-10; //focal length of the concave mirror(in cm)

//calculation
v=1/((1/f)-(1/u)); //mirror formula

if(v>0)
    disp(v,'virtual image is formed on right side of mirror at a distance(in cm)');
else
    disp(v,'real image is formed on left side of mirror at a distance(in cm)');
end
```

Scilab code Exa 18.2 calculation of length of the image of an object placed at a distance from a concave mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 18.2
//calculation of length of the image of an object placed at a distance from a concave mirror.

//given data
//F=-f    focal length(in cm)
//u=-1.5f    object distance(in cm)
h1=2.5; //object height(in cm)

//calculation
//v=1/((1/F)-(1/u))    mirror formula
//v=-3f
//also m=-v/u    lateral magnification formula for mirror
//m=-2    lateral magnification ratio

m=-2; //lateral magnification ratio
h2=m*h1; //lateral magnification formula

if(h2>0)
    disp(h2,'image is erect and is of length(in cm)');
else
    disp(h2,'image is inverted and is of length(in cm)');
end
```

Scilab code Exa 18.2w calculation of length of the image of an object placed horizontal at a distance from the mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.2w
```

```
//calculation of length of the image of an object
   placed horizontal at a distance from the mirror.

//given data
u=-30; //object distance of point A(in cm)
f=-10; //focal length of the mirror(in cm)
//r=2f=20 cm
//image of B is formed at centre of curvature since
   it is located at the centre of curvature.

//calculation
v=1/((1/f)-(1/u)); //mirror formula

disp(v+(2*-f),'length(in cm) of the image is');
```

Scilab code Exa 18.3 calculation of shift in the position of printed letters by a glass cube

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.3
//calculation of shift in the position of printed
   letters by a glass cube

//given data
t=6; //thickness of the cube(in cm)
mu=1.5; //refractive index of glass cube

//calculation
deltat=(1-1/mu)*t; //vertical shift formula derived
   from snell's law

disp(deltat,'shift(in cm) in the position of printed
    letters is');
```

Scilab code Exa 18.3w calculation of object distance for half image height as compared to original height in case of reflection by convex mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.3w
//calculation of object distance for half image
   height as compared to original height in case of
   reflection by convex mirror

//given data
m=.5; //magnification ratio
f=2.5; //focal length of the convex mirror(in m)

//calculation
//(1/u)+(1/v)=(1/f); //mirror formula
//now m=-v/u=0.5
u=-f; //from formula taking v=-u/2 mirror formula
   gives this relation

disp(abs(u),'the boy should stand at a distance(in m
   ) from the convex mirror');
```

Scilab code Exa 18.4 calculation of refractive index of material from known critical angle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.4
```

```
//calculation of refractive index of material from known critical angle

//given data
thetac=48.2; //critical angle for water(in degree)

//calculation
//snell's law with respect to total internal reflection
mu=1/sind(thetac); //sind represents that the argument is in degree

disp(mu,'refractive index of material is ');
```

Scilab code Exa 18.4w calculation of image distance and focal length of concave mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.4w
//calculation of image distance and focal length of concave mirror

//given data
h1=2; //height of object(in cm)
h2=-5; //height of image(in cm)
u=-12; //object distance in cm

//calculation
v=-(h2/h1)*u //image distance(in cm) using formula of lateral magnification

if(v<0)
    disp(-v,'image is formed on same side of object
```

```
        at a distance(in cm)');
else
    disp(v,'image is formed on opposite side of
        mirror at a distance(in cm)');
end

f=(u*v)/(u+v); //mirror formula

disp(abs(f),'focal length(in cm) of the given
    concave mirror is');
```

Scilab code Exa 18.5 calculation of refractive index of material from known value of angle of minimum deviation by prism

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.5
//calculation of refractive index of material from
    known value of angle of minimum deviation by
    prism

//given data
deltam=37; // angle of minimum deviation by prism of
     the material(in degree)
A=53; //angle of prism(in degree)

//calculation
mu=sind((A+deltam)/2)/sind(A/2); //relation between
    refractive index and angle of minimum deviation
    by prism

disp(mu,'refractive index of material of the prism
    is');
```

Scilab code Exa 18.5w calculation of maximum angle of reflection for a surface

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.5w
//calculation of maximum angle of reflection for a
    surface

//given data
mu=1.25; //refractive index of medium

//calculation
thetadashdash=asind(1/mu); //critical angle for
    total internal reflection(in degree)
thetadash=90-thetadashdash;
theta=asind(mu*sind(thetadash)); //snell's law
    sin(theta)/sin(thetadash)=mu

disp(theta,'maximum value of theta(in degree) for
    total internal reflection at vertical surface');
```

Scilab code Exa 18.6 calculation of position of the image of an object placed at a distance from spherical convex surface

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.6
//calculation of position of the image of an object
    placed at a distance from spherical convex
```

```
   surface

//given data
u=-15; //object distance(in cm)
R=30; //radius of curvature of the spherical convex
   surface(in cm)
mu1=1; //refractive index of the medium in which
   object is kept
mu2=1.5; //refractive index of the medium of
   spherical convex surface

//calculation
v=mu2/((mu2-mu1)/R+(mu1/u)); //formula for
   refraction at spherical surface

if(v>0)
    disp(v,'real image is formed on right side of
       spherical surface at a distance(in cm)');
else
    disp(v,'virtual image is formed on left side of
       spherical surface at a distance(in cm)');
end
```

Scilab code Exa 18.6aw calculation of minimum refractive index for parallel emergence for given condition in right prism

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.6aw
//calculation of minimum refractive index for
   parallel emergence for given condition in right
   prism

//given data
```

```
thetac=45; //critical angle(in degree) for given conditions

//calculation
mu=1/(sind(thetac)); //anell's law

disp(mu,'for total internal reflection refractive index of material of given right prism should be greater than or equal to');
```

Scilab code Exa 18.6bw verification of total internal reflection for given conditions of right prism

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.6bw
//verification of total internal reflection for given conditions of right prism

//given data
mu=5/3; //refracive index of the material of the right prism

//calculation
thetac=asind(1/mu) //snell's law

if(thetac<60)
    disp('total internal reflection does not take place for given conditions of right prism');
else
    disp('total internal reflection do take place for given conditions of right prism');
end
```

Scilab code Exa 18.7 calculation of the size of the image of an object placed at a distance from the spherical concave surface

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.7
//calculation of the size of the image of an object
   placed at a distance from the spherical concave
   surface

//given data
u=-40; //object distance(in cm)
R=-20; //radius of curvature of the spherical
   concave surface(in cm)
mu1=1; //refractive index of the medium in which
   object is kept
mu2=1.33; //refractive index of the medium of
   spherical concave surface
h1=1; //size of the object(in cm)

//calculation
v=mu2/((mu2-mu1)/R+(mu1/u)); //formula for
   refraction at spherical surface
h2=(mu1*v*h1)/(mu2*u);  //formula for lateral
   magnification

if(h2>0)
    disp(h2,'image is erect and is of size(in cm)');
else
    disp(h2,'image is inverted and is of size(in cm)
       ');
end
```

Scilab code Exa 18.8 calculation of focal length of a biconvex lens from known value of radii of curvature of refracting surfaces

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.8
//calculation of focal length of a biconvex lens
   from known value of radii of curvature of
   refracting surfaces

//given data
R1=20; //radius of curvature(in cm) of first surface
    of biconvex lens
R2=-20; //radius of curvature(in cm) of second
   surface of biconvex lens
mu=1.5; //refractive index of the material of lens

//calculation
f=1/((mu-1)*(1/R1-1/R2)); //lens maker's formula

disp(f,'focal length(in cm) of the given biconvex
   lens is');
```

Scilab code Exa 18.9 calculation of size of the image of an object placed at a distance from a convex lens

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.9
```

```
//calculation of size of the image of an object
   placed at a distance from a convex lens

//given data
f=12 //focal length(in cm)
u=-8 //object distance(in cm)
h1=2; //object height(in cm)

//calculation
v=1/((1/f)+(1/u)); //lens formula
m=v/u; //lateral magnification formula for lens
h2=m*h1; //lateral magnification formula for lens

if(h2>0)
    disp(h2,'image is erect and is of length(in cm)'
       );
else
    disp(h2,'image is inverted and is of length(in
       cm)');
end
```

Scilab code Exa 18.11w locating image of a dust particle on the surface of water filled in a concave mirror as observed from top

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.11w
//locating image of a dust particle on the surface
   of water filled in a concave mirror as observed
   from top

//given data
R=-40; //radius of curvature(in cm) of the concave
   mirror
```

```
u=-5; //object distance(in cm) from the concave
    mirror
mu=1.33; //refractive index of water

//calculation
v=1/((2/R)-(1/u))//mirror formula

if(v>0)
    disp(v,'virtual image is formed due to
        reflection through concave mirror below
        surface of mirror at a depth(in cm) of ');
else
    disp(v,'real image is formed due to reflection
        through concave mirror above surface of
        mirror at a height(in cm) of ');
end

total_distance=v+(-u);  //water is filled upto
    height equal to object distance of dust particle
    P
vfinal=total_distance*(1-1/mu); //snell's law

disp(total_distance-vfinal,'final image is formed
    below water surface at a distance(in cm)');
```

Scilab code Exa 18.12w calculation of position of final image formed due to a system of glass slab and a concave mirror

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.12w
//calculation of position of final image formed due
    to a system of glass slab and a concave mirror

```

```
//given data
u=-21; //object distance(in cm) from concave mirror
R=20; //radius of curvature(in cm) of the concave mirror
mu=1.5; //refractive index of the glass'
t=3; //thickness of glass slab(in cm)

//calculation
tshift=t*(1-1/mu); //snell's law
img_pos=-u-tshift; //image position with respect to glass slab,i.e object distance(in cm) of concave mirror

if(img_pos==R)
    disp('here img_pos is same as radius of curvature of concave mirror and thus final image is formed at P itself');
```

Scilab code Exa 18.13w calculation angle of minimum deviation for equilateral prism of silicate flint glass from known value of wavelength

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.13w
//calculation angle of minimum deviation for equilateral prism of silicate flint glass from known vlue of wavelength

//given data
A=60; //angle of prism(in degree)
mu1=1.66; //refractive index of silicate flint glass for 400nm wavelength
mu2=1.61; //refractive index of silicate flint glass for 700nm wavelength
```

```

//calculation
//mu=sind((A+deltam)/2)/sind(A/2)    relation between refractive index and angle of minimum deviation by prism
deltam1=2*((asind(mu1*sind(A/2)))-30);
deltam2=2*((asind(mu2*sind(A/2)))-30);

disp(deltam1,'minimum angle of deviation(in degree) for 400nm wavelength in equilateral prism of silicate flint glass is');
disp(deltam2,'minimum angle of deviation(in degree) for 700nm wavelength in equilateral prism of silicate flint glass is');
```

Scilab code Exa 18.14w calculation of angle of rotation of the mirror in given setup

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.14w
//calculation of angle of rotation of the mirror in given setup

//given data
mu=1.5; //refractive index of convex lens
A=4; //angle of prism (in degree)

//calculation
delta=(mu-1)*A

disp(delta,'the mirror should be rotated by angle(in degree) of');
```

Scilab code Exa 18.15w calculation of location of the image of an object placed at a distance from the spherical convex surface

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.15w
//calculation of location of the image of an object
   placed at a distance from the spherical convex
   surface

//given data
u=-25; //object distance(in cm)
R=20; //radius of curvature of the spherical convex
   surface(in cm)
mu1=1; //refractive index of the medium in which
   object is kept
mu2=1.5; //refractive index of the medium of
   spherical convex surface

//calculation
v=mu2/((mu2-mu1)/R+(mu1/u)) //formula for refraction
    at spherical surface

if(v>0)
    disp(v,'image is formed on the right of the
       separating surface at a distance(in cm) of');
else
    disp(-v,'image is formed on the left of the
       separating surface at a distance(in cm) of');
end
```

Scilab code Exa 18.16w calculation of height of the image of an object placed along axis at a distance from a horizontal cylindrical glass rod

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.16w
//calculation of height of the image of an object
   placed along axis ,at a distance from a
   horizontal cylindrical glass rod

//given data
u=-20; //object distance (in cm)
R=5; //radius of curvature of the spherical convex
   surface (in cm)
mu1=1; //refractive index of the medium in which
   object is kept
mu2=1.5; //refractive index of the medium of
   spherical concave surface
h1=.5; //height of the object in mm

//calculation
v=mu2/((mu2-mu1)/R+(mu1/u)) //formula for refraction
    at spherical surface
m=(mu1*v)/(mu2*u); //lateral magnification ratio

if(v>0)
    disp(v,'image is formed inside the rod at a
       distance(in cm) of');
if(m==-1)
    disp('the image will be of same height as the
       object and is inverted');
if(m==1)
    disp('the image will be of same height as the
       object and is erect');
end
```

Scilab code Exa 18.17w calculation of apparent depth of a air bubble inside a glass sphere

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.17w
//calculation of apparent depth of a air bubble
    inside a glass sphere

//given data
u=-4; //object distance (in cm)
R=-10; //radius of curvature of the spherical glass
    sphere(in cm)
mu1=1.5; //refractive index of the glass sphere
mu2=1; //refractive index of air bubble

//calculation
v=mu2/((mu2-mu1)/R+(mu1/u)) //formula for refraction
    at spherical surface

if(v<0)
    disp(-v,'below the surface the bubble will
        appear at a distance(in cm) of');
 else
     disp(v,'above the surface the bubble will
         appear at a distance(in cm) of');
end
```

Scilab code Exa 18.18w calculation of position of image due to refraction at the first surface and position of final image

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.18w
//calculation of position of image due to refraction at the first surface and position of final image

//given data1
//u=infinite    object distance (in cm)
R=2; //radius of curvature of the spherical convex surface (in mm)
mu1=1.33; //refractive index of the medium from which light beam is entering
mu2=1; //refractive index of the medium of spherical air bubble
nR=-2; //radius of curvature of the spherical convex surface (in mm)
nmu1=1; //refractive index of the medium in which previous image is formed
nmu2=1.33; //refractive index of the medium from which light beam is entering

//calculation
v=R/(mu2-mu1) //formula for refraction at spherical surface for object at infinite distance
nu=-(-v+-(2*nR))

if(v<0)
    disp(-v,'virtual image is formed on the same side of water at a distance(in mm) of');
 else
     disp(v,'real image is formed on the other side of water at a distance(in mm) of');
end

nv=nmu2/((nmu2-nmu1)/nR+(nmu1/nu)) //formula for refraction at spherical surface

```

```
if(nv<0)
    disp(-nv,'final image is formed on the same side
        of air at a distance(in mm) of');
 else
     disp(nv,'final image is formed on the other
        side of air at a distance(in mm) of');
end

disp(-v+R,'from the centre first image is formed on
   the side from which incident rays are coming at a
    distance(in mm) of ');
disp(-nv+nR,'from the centre second image is formed
   on the side from which incident rays are coming
   at a distance(in mm) of ');
```

Scilab code Exa 18.19w calculation of focal length of thin lens

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.19w
//calculation of focal length of thin lens

//given data
R1=10; //radius of curvature(in cm) of first surface
    of given lens
R2=20; //radius of curvature(in cm) of second
   surface of given lens
mu=1.5; //refractive index of the material of lens

//calculation
f=1/((mu-1)*(1/R1-1/R2)); //lens maker's formula

disp(f,'focal length(in cm) of the given lens is');
```

Scilab code Exa 18.20w calculation of position of diverging mirror to obtain real image at the source itself for given system

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.20w
//calculation of position of diverging mirror to
   obtain real image at the source itself for given
   system

//given data
u=-15; //object distance(in cm)
f=10; //focal length(in cm) of converging lens
fm=12; //focal length(in cm) of convex mirror

//calculation
v=1/((1/f)+(1/u)); //lens formula
LI1=2*abs(u);
MI1=2*abs(fm);
LM=LI1-MI1;

disp(LM,'on the right of the lens mirror should be
   placed at a distance(in cm) of');
```

Scilab code Exa 18.21w calculation of separation between mirror and the lens for parallel emergence of the final beam

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.21w
```

```
//calculation of separation between mirror and the
   lens for parallel emergence of the final beam

//given data
u=-12; //object distance(in cm)
f=15; //focal length(in cm) of the converging lens

//calculation
v=1/((1/f)+(1/u)); //lens formula

if(v<0)
    disp(-v,'image due to lens is formed on the left
        side of the lens at a distance(in cm) of');
else
    disp(v,'image due to lens is formed on the right
        side of the lens at a distance(in cm) of');
end

I1L=2*abs(v);
LI2=abs(f);
I1I2=I1L+LI2;

//let distance of mirror from I2 is x
//I1I2=75 cm
//u=-(75+x) cm
//v=-x cm
//f_mirror=-20 cm
//(1/v)+(1/u)=(1/f);    mirror formula
//substituting u,v,f we get  equation       x^2+35*X
   -1500=0

a=1; // for above equation coefficient of x^2
b=35; // for above equation coefficient of x^1
c=-1500; // for above equation coefficient of x^0 or
    the constant

x1=(-b+sqrt((b*b)-(4*a*c)))/(2*a); //first solution
x2=(-b-sqrt((b*b)-(4*a*c)))/(2*a); //second solution
```

```
//considering only the positive value of the
   solution ,as negative value has no physical
   meaning
if(x1>0)
    disp(f+x1,'the separation(in cm) between the
       lens and the mirror is');
if(x2>0)
    disp(f+x2,'the separation(in cm) between the
       lens and the mirror is');
end
```

Scilab code Exa 18.22w calculation of object distance from the lens with one side silvered

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.22w
//calculation of object distance from the lens with
   one side silvered

//given data
v=-25; //image distance (in cm)
R=25; //radius of curvature of the spherical convex
   surface (in cm)
mu1=1; //refractive index of the medium in which
   object is kept
mu2=1.5; //refractive index of the medium of lens

//calculation
u=mu1/((mu2/v)-((mu2-mu1)/R)); //formula for
   refraction at spherical surface

disp(abs(u),'object should be placed at a distance(
   in cm) of');
```

Scilab code Exa 18.23w calculation of location of image of an object placed in front of a concavo convex lens made of glass

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.23w
//calculation of location of image of an object
   placed in front of a concavo-convex lens made of
   glass

//given data
R1=20; //radius of curvature(in cm) of first surface
    of concavo-convex lens
R2=60; //radius of curvature(in cm) of second
   surface of concavo-convex lens
mu=1.5; //refractive index of the material of lens
u=-80; //object distance(in cm)
C1C2=160; //coaxial distance(in cm) between both the
    lenses

//calculation
f=1/((mu-1)*(1/R1-1/R2)); //lens maker's formula

disp(f,'focal length(in cm) of the given concavo-
   convex lens is');

v=1/((1/u)+(1/f)); //lens formula

if(v>0)
    disp(v,'first image is formed on right side of
       first lens at a distance(in cm) of');
else
    disp(-v,'first image is formed on left side of
```

```
        first lens at a distance(in cm) of');
end

ff=f; //focal length(in cm) of the second lens same
    as first lens
uf=v-C1C2 //object distance(in cm) for second lens
    since image by first lens acts as object of the
    second lens
vf=1/((1/uf)+(1/ff)); //lens formula

if(vf>0)
    disp(vf,'final image is formed on right side of
        second lens at a distance(in cm) of');
else
    disp(-vf,'final image is formed on left side of
        second lens at a distance(in cm) of');
end
```

Scilab code Exa 18.24w calculation of new focal length of a lens on immersing in water

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.24w
//calculation of new focal length of a lens on
    immersing in water

//given data
f=12; //focal length(in cm) of the lens
mu1=1; //refractive index of air
mu2=1.5; //refractive index of glass
mu3=1.33; //refractive index of water
//let (1/R1)-(1/R2)=a   variable

```

```
//calculation
a=1/((mu2/mu1-1)*(f)) //refractive mediums in cascading
f_new=1/((mu2/mu3-1)*a)//refractive mediums in cascading

disp(f_new,'new focal length(in cm) of a lens on immersing it in water is');
```

Scilab code Exa 18.25w calculation of location of final image for an object on the axis of a cylindrical tube containing water closed by an equiconvex lens

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.25w
//calculation of location of final image for an object on the axis of a cylindrical tube containing water closed by an equiconvex lens

//given data
u=-21; //object distance(in cm)
f=10; //focal length(in cm) of the lens
mu1=1; //refractive index of air
mu2=1.5; //refractive index of lens
mu3=1.33; //refractive index of water
//v1  image due to refraction at the first surface

//calculation
//from formula of refraction at the spherical surface
//(mu2/v1)-(1/u)=(mu2-mu1)/R                (1)
//(mu3/v)-(mu2/v1)=(mu3-mu2)/-R             (2)
//adding (1) and (2)
```

```
//(1/v)=(1/(2*R))-(1/28)                    (3)
//f=1/((mu2-1)*(1/R+1/R)) refractive surfaces in
   cascade

R=2*f*(mu2-1) //refractive surfaces in cascade
v=1/((1/(2*R))-(1/28)) //from equation (3)

disp(v,'the image is formed inside the cylindrical
   tube at distance(in cm) of');
```

Scilab code Exa 18.26w calculation of new position of the slide of projector if the position of the screen is changed

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.26w
//calculation of new position of the slide of
   projector if the position of the screen is
   changed

//given data
v=10; //image distance(in m)
m=500; //lateral magnification ratio
d=2; //distance(in m) the screen is moved

//calculation
u=-v/m; //lateral magnification formula
f=1/((1/v)-(1/u)) //lens formula
vdash=v-d //effect of moving screen d m closer
udash=1/((1/vdash)-(1/f)) //lens formula

if(udash<0)
    disp(-udash,'away from the lens,the slide should
        be moved by a distance(in m) of');
```

```
else
    disp(udash,'towards the lens,the slide should be moved by a distance(in m) of');
end
```

Scilab code Exa 18.27w calculation of position of the object to get a focused image

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.27w
//calculation of position of the object to get a focused image

//given data
v=10; //image distance(in cm) from the convex lens
u=-10; //object distance(in cm) from the convex lens
mu=1.5; //refractive index of glass
t=1.5; //thickness(in cm) of the glass plate inserted

//calculation
f=1/((1/v)-(1/u)) //lens formula
I1I=t*(1-1/mu) //shift in position(in cm) of image due to glass plate
v_new=v-I1I //lens forms image at this distance(in cm) from itself
u_new=1/((1/v_new)-(1/f)) //lens formula

disp(abs(u_new),'from the lens,the object should be placed at a distance(in cm) of');
```

Scilab code Exa 18.28aw finding the image of a distant object formed by combination of two convex lens by using thin lens formula

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.28aw
//finding the image of a distant object formed by
   combination of two convex lens by using thin lens
    formula

//given data
f=20; //focal length(in cm) of the given convex lens
d=60; //coaxial separation(in cm) between the two
   convex lenses
u=-(d-f); //object distance(in cm) for the second
   lens since first image is formed at focus of
   first lens

//calculation
v=1/((1/u)+(1/f)); //lens formula

disp(v,'final image is formed on the right of the
   second lens at a distance(in cm) of');
```

Scilab code Exa 18.28bw finding the image of a distant object formed by combination of two convex lens by using equivalent lens method

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 18.28bw
//finding the image of a distant object formed by
   combination of two convex lens by using
   equivalent lens method
```

```

//given data
f1=20; //focal length(in cm) of the first convex
   lens
f2=20; //focal length(in cm) of the first convex
   lens
d=60; //coaxial separation(in cm) between the two
   convex lenses

//calculation
F=1/((1/f1)+(1/f2)-(d/(f1*f2))); //equivalent focal
   length formula for equivalent lens method
D=d*F/f1; //distance(in cm) from the second lens at
   which equivalent lens is to be placed
//image of distant object is formed at focus of
   equivalent lens

disp(abs(D)-abs(F),'on right side of the second lens
   ,the final image is formed at a distance(in cm)
   of ');
```

Chapter 19

Optical Instruments

Scilab code Exa 19.1 determining which boy appears taller

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.1
//determining which boy appears taller

//given data
d1=4//distance(in m) of boy1 from the eye
d2=5//distance(in m) of boy2 from the eye
h1=52//height(in inch) of boy1
h2=55//height(in inch) of boy2

//calculation
alpha1=h1/d1//angle subtended by the first boy on
    the eye
alpha2=h2/d2//angle subtended by the second boy on
    the eye
if(alpha1>alpha2)
    printf('the first boy will look taller to the
        eye')
elseif(alpha1<alpha2)
```

```
    printf('the second boy will look taller to the eye')
else
    printf('Both boys will appear same in height to the eye')
end
```

Scilab code Exa 19.1w calculation of the angular magnification

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.1w
//calculation of the angular magnification

//given data
f=12*10^-2//focal length(in m) of the simple microscope
D=25*10^-2//distance(in m) at which the image is formed away from the eye

//calculation
m=1+(D/f)//angular magnification

printf('the angular magnification is %3.2f',m)
```

Scilab code Exa 19.2 calculation of the angular magnification and the length of the microscope tube

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.2
```

```
//calculation of the angular magnification and the length of the microscope tube

//given data
fo=1*10^-2//focal length(in m) of the objective lens
fe=2.5*10^-2//focal length(in m) of the eyepiece
u=-1.2*10^-2//object distance(in m)
D=25*10^-2//least distance(in m) for the clear vision

//calculation
v=1/((1/fo)+(1/u))//distance where the first image is formed ....by the lens formula
m=(v*D)/(u*fe)//angular magnification
L=v+fe//length of the tube

printf('the angular magnification is %d',round(m))
printf('\nthe length of the microscope tube is %3.1f cm',L*10^2)
```

Scilab code Exa 19.2w calculation of the object distance to obtain maximum angular magnification for a normal eye

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.2w
//calculation of the object distance to obtain maximum angular magnification for a normal eye

//given data
D=10//power(in D) of the lens
v=-25*10^-2//image distance(in m) i.e at the near point

```

```
//calculation
f=1/D//focal length
u=1/((1/v)-(1/f))//lens formula

printf('the object distance to obtain maximum
   angular magnification for a normal eye is %3.1f
   cm',u*10^2)
```

Scilab code Exa 19.3 calculation of the power of lens for the spectacles

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.3
//calculation of the power of lens for the
   spectacles

//given data
d=1.5//distance(in m) upto which the man can clearly
    see objects

//calculation
f=-d//focal length of the lens
P=1/f//definition of power of the lens

printf('the power of lens for the spectacles is %3.2
   f D',P)
```

Scilab code Exa 19.3w calculation of the position of the image linear magnification and the angular magnification

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 19.3w
//calculation of the position of the image ,linear
   magnification and the angular magnification

//given data
u=-3.6*10^-2//object distance(in m)
f=4*10^-2//focal length(in m)
D=25*10^-2//least distance for clear vision

//calculation
v=1/((1/f)+(1/u))//lens formula
m=v/u//linear magnification
alpha=D/abs(u)//angular magnification

printf('the image distance is %d cm',v*10^2)
printf('\nthe linear magnification is %d',m)
printf('\nthe angular magnification is %3.1f',round(
   alpha))
```

Scilab code Exa 19.4w calculation of the object distance and the angular magnification

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.4w
//calculation of the object distance and the angular
    magnification

//given data
fo=1*10^-2//focal length(in m) of the objective lens
fe=5*10^-2//focal length(in m) of the eyepiece
d=12.2*10^-2//separation(in m) between the objective
    lens and the eyepiece
```

```
D=25*10^-2//least distance(in m) for the clear visio

//calculation
ve=-D//image distance for the eyepiece
ue=1/((1/ve)-(1/fe))//object distance for eyepiece
    ....by the lens formula
vo=d-abs(ue)//image distance for objective lens
uo=1/((1/vo)-(1/fo))//object distance for objective
    lens....by the lens formula
m=(vo/uo)*(1+(D/fe))//angular magnification

printf('the object should be placed at a distance of
     %3.1f cm from the objective lens to focus it
    properly',abs(uo*10^2))
printf('\nthe angular magnification is %d',m)
```

Scilab code Exa 19.5w calculation of the object distance and the angular magnification for the least strain in the eyes

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.5w
//calculation of the object distance and the angular
     magnification for the least strain in the eyes

//given data
fo=.5*10^-2//focal length(in m) of the objective
    lens
fe=5*10^-2//focal length(in m) of the eyepiece
d=7*10^-2//separation(in m) between the objective
    lens and the eyepiece
D=25*10^-2//least distance(in m) for the clear
    vision

```

```
//calculation
v=d-fe//distance at which the first image should be formed
u=1/((1/v)-(1/fo))//lens formula for the objective lens
m=(v*D)/(u*fe)//angular magnification

printf('the object distance for the least strain in the eyes is %3.1f cm',abs(u*10^2))
printf('\nthe angular magnification for the least strain in the eyes is %d',m)
```

Scilab code Exa 19.6w calculation of the length of the tube and the angular magnification produced by the telescope

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.6w
//calculation of the length of the tube and the angular magnification produced by the telescope

//given data
fo=200*10^-2//focal length(in m) of the objective lens
fe=4*10^-2//focal length(in m) of the eyepiece
u=10*10^3//object distance(in m)

//calculation
L=fo+fe//length of the tube
m=-fo/fe//angular magnification

printf('the length of the tube is %d cm',L*10^2)
printf('\ngthe angular magnification is %d',m)
```

Scilab code Exa 19.7w calculation of the tube length magnifying power and angular magnification

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.7w
//calculation of the tube length,magnifying power and angular magnification

//given data
fo=50*10^-2//focal length(in m) of the objective lens
fe=-5*10^-2//focal length(in m) of the eyepiece
u=-2//object distance(in m)

//calculation
L=fo-abs(fe)//length of the tube
m=-fo/fe//magnifying power
v=1/((1/fo)+(1/u))//by lens formula for the objective lens
Ldash=v-abs(fe)//tube length
mdash=v/abs(fe)//angular magnification

printf('the tube length for large distance viewing is %d cm',L*10^2)
printf('\nthe magnifying power for the large distance viewing is %d',m)
printf('\nthe tube length for viewing object at 2 m is %3.2f cm',Ldash*10^2)
printf('\nthe angular magnification for viewing object at 2 m is %3.2f',mdash)
```

Scilab code Exa 19.8w calculation of the angular magnification due to the converging lens

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.8w
//calculation of the angular magnification due to the converging lens

//given data
f=50*10^-2//focal length(in m) of the converging lens
d=25*10^-2//distance(in m) from where the image can be seen by unaided eye

//calculation
//linear size = f*alpha
//angle formed .....abs(beta) = f*abs(alpha)/d
m=-f/d//angular magnification ...m = -abs(beta)/abs(alpha)

printf('the angular magnification due to the converging lens is %d',m)
```

Scilab code Exa 19.9w calculation of the power of lens and maximum distance that can be seen clearly

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.9w
```

```
//calculation of the power of lens and maximum
   distance that can be seen clearly

//given data
u=-25*10^-2//object distance(in m)
v=-40*10^-2//image distance(in m)....i.e equal to
   near point distance
vdash=-250*10^-2//maximum distance(in m) that an
   unaided eye can see....i.e equal to far point
   distance

//calculation
f=1/((1/v)-(1/u))//focal length ....by using the
   lens formula
P=1/f//power of the lens
d=1/((1/vdash)-(1/f))//maximum distance for clear
   vision.... by using the lens formula

printf('the power of the lens is %3.1f D',P)
printf('\nthe maximum distance upto which,the person
    will be able to see clearly is %d cm',round(abs(
   d*10^2)))
```

Scilab code Exa 19.10w calculation of the near point and the distance of the retina from the lens

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 19.10w
//calculation of the near point and the distance of
   the retina from the lens

//given data
P1=50//power1(in D) of the lens
```

```
P2=60//power2(in D) of the lens

//calculation
//for the eye in fully relaxed condition,the focal length is the largest.
//larger the focal length,smaller is the power of lens
if(P1<P2)
    P=P1
else
    P=P2
end
f=1/P//distance of the retina from lens ,equal to the focal length
//for eye focused at near point the power is maximum
if(P1>P2)
    Pdash=P1
else
    Pdash=P2
end
fdash=1/Pdash//focal length
v=abs(f)//image is formed at the retina
u=1/((1/v)-(1/fdash))//near point......using the lens formula

printf('the distance of the retina from the lens is %d cm',f*10^2)
printf('\nthe near point is at %d cm',abs(u*10^2))
```

Chapter 20

Dispersion and Spectra

Scilab code Exa 20.1 calculation of the dispersive power of the flint glass

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 20.1
//calculation of the dispersive power of the flint
   glass

//given data
mur=1.613//refractive index of flint glass for the
   red light
mu=1.620//refractive index of flint glass for the
   yellow light
muv=1.632//refractive index of flint glass for the
   violet light

//calculation
w=(muv-mur)/(mu-1)//definition of the dispersive
   power

printf('the dispersive power of the flint glass is
   %3.4f',w)
```

Scilab code Exa 20.1w calculation of the angular dispersion produced by a thin prism of the flint glass

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 20.1w
//calculation of the angular dispersion produced by a thin prism of the flint glass

//given data

mur=1.613//refractive index of flint glass for the red light
muv=1.632//refractive index of flint glass for the violet light
A=5//refracting angle(in degree)

//calculation
delta=(muv-mur)*A//angular dispersion

printf('the angular dispersion produced by the thin prism of the flint glass is %3.3f degree',delta)
```

Scilab code Exa 20.2 calculation of the dispersive power of the material of the lens

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 20.2
```

```
//calculation of the dispersive power of the material of the lens

//given data
fr=90//focal length(in cm) for the red light
fv=86.4//focal length(in cm) for the violet light

//calculation
//(1/f) = (mu-1) * ((1/R1) - (1/R2))
//muv - 1 =K/fv.....and.....mur - 1 = K/fr
//let m = muv - mur and K = 1
m=((1/fv)-(1/fr))
//muy - 1 = ((muv + mur)/2) - 1 = (K/2)*((1/fv) - (1/fr))
//let n = muy -1 and K = 1
n=(1/2)*((1/fv)+(1/fr))
//w = (muv-mur)/(mu-1).........definition of the dispersive power
w=m/n

printf('the dispersive power of the material of the lens is %3.3f',w)
```

Scilab code Exa 20.2w calculation of the angle of flint glass prism and angular dispersion produced by the combination

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 20.2w
//calculation of the angle of flint glass prism and angular dispersion produced by the combination

//given data
A=5//angle of crown glass prism(in degree)
```

```
mur=1.514//refractive index of crown glass for the red light
mu=1.517//refractive index of crown glass for the yellow light
muv=1.523//refractive index of crown glass for the violet light
murdash=1.613//refractive index of flint glass for the red light
mudash=1.620//refractive index of flint glass for the yellow light
muvdash=1.632//refractive index of flint glass for the violet light

//calculation
//delta = (mu - 1) * A.......deviation produced by the prism
//D = ((mu - 1)*A) - ((mudash - 1)*Adash)....net deviation
//net deviation for the mean ray is equal to zero
Adash=((mu-1)/(mudash-1))*A//angle of flint glass prism
//deltav - deltar = (muv - mur)*A .........................for crown glass prism
//deltavdash - deltardash = (muvdash - murdash)*Adash...for flint glass prism
delta=((muv-mur)*A)-((muvdash-murdash)*Adash)//net angular dispersion

printf('the angle of flint glass prism needed is %3.1f degree',Adash)
printf('\nthe angular dispersion produced by the combination is %3.4f degree',abs(delta))
```

Scilab code Exa 20.3w calculation of the refracting angles of the two prisms

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 20.3w
//calculation of the refracting angles of the two
   prisms

//given data
w=0.03//dispersive power of crown glass
wdash=0.05//dispersive power of flint glass
delta=1//deviation(in degree) produced
mu=1.517//refractive index for crown glass
mudash=1.621//refractive index for flint glass

//calculation
//w = (muv - mur)/(mu - 1)........dispersive power
//(muv - mur)*A = (mu-1)*w*A......angular dispersion
m=((mu-1)/(mudash-1))*(w/wdash)
//Adash = A*m........(1)
//net deviation produced is delta
A=delta/((mu-1)-((mudash-1)*m))//refracting angle of
    crown glass
Adash=A*m//refracting angle of flint glass

printf('the refracting angle of the crown prism is
   %3.1f degree',A)
printf('\nthe refracting angle of the flint prism is
    %3.1f degree',Adash)
```

Chapter 22

Photometry

Scilab code Exa 22.1 calculation of the luminous flux

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 22.1
//calculation of the luminous flux

//given data
lambda=600//wavelength(in nm) given
P=10//wattage(in W) of source
rellum=.6//relative luminosity

//calculation
//1 W source of 555 nm = 685 lumen
lumflux=P*685*rellum//luminous flux

printf('the luminous flux is %d lumen',lumflux)
```

Scilab code Exa 22.1w calculation of the total radiant flux total luminous flux and the luminous efficiency

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 22.1w
//calculation of the total radiant flux,total luminous flux and the luminous efficiency

//given data
E1=12//energy(in J) emitted by the source
lambda1=620*10^-9//wavelength(in m) of the light1
E2=8//energy(in J) emitted by the source
lambda2=580*10^-9//wavelength(in m) of the light2
rellum1=.35//relative luminosity of the light1
rellum2=.80//relative luminosity of the light2

//calculation
radflux=E1+E2//total radiant flux
lumflux1=E1*685*rellum1//luminous flux corresponding to the 12 W
lumflux2=E2*685*rellum2//luminous flux corresponding to the 8 W
lumflux=lumflux1+lumflux2//total luminous flux
lumeff=lumflux/radflux//luminous efficiency

printf('the total radiant flux is %d W',radflux)
printf('\nthe total luminous flux is %d lumen',lumflux)
printf('\nthe luminous efficiency is %d lumen W^-1',lumeff)
```

Scilab code Exa 22.2w calculation of the total luminous flux emitted by the source and the total luminous intensity of the source

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
```

```
clc;clear;
//example 22.2w
//calculation of the total luminous flux emitted by
   the source and the total luminous intensity of
   the source

//given data
r=1*10^-2//radius(in m) of the circular area
d=2//distance(in m) from the point source
lumflux=2*10^-3//luminous flux(in lumen)

//calculation
deltaw=(%pi*r*r)/(d*d)//solid angle subtended by the
    area on the point source
F=(4*%pi*lumflux)/(deltaw)//total luminous flux
lumint=lumflux/deltaw//luminous intensity

printf('the total luminous flux emitted by the
   source is %d lumen',round(F))
printf('\nthe total luminous intensity of the source
    is %d cd',lumint)
```

Scilab code Exa 22.3w calculation of the luminous flux falling on a plane

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 22.3w
//calculation of the luminous flux falling on a
   plane

//given data
P=100//power(in W) input of the bulb
lumeff=25//luminous efficiency(in lumen W^-1)
A=1*10^-4//area(in m^2)
```

```
d=50*10^-2//distance(in m) of the area from the lamp

//calculation
deltaF=lumeff*P//luminous flux emitted by the bulb
I=deltaF/(2*%pi)
deltaw=A/d^2//solid angle(in sr)subtended by the
    object on the lamp
//I = deltaF/deltaw ......luminous intensity
deltaF=I*deltaw//luminous flux emitted in the solid
    angle

printf('the luminous flux falling on the plane is %3
    .2f lumen',deltaF)
```

Scilab code Exa 22.4w calculation of the illuminance at a small surface area of the table top

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 22.4w
//calculation of the illuminance at a small surface
    area of the table-top

//given data
d=.50//distance(in m) of the point source above the
    table-top
lumflux=1570//luminous flux(in lumen) of the source
d1=.8//distance(in m)from the source

//calculation
I=lumflux/(4*%pi)//luminous intensity of the source
    in any direction

//E=I*cosd(theta)/r^2.......illuminance
```

```
r=d//for point A
theta=0//for point A
EA=I*cosd(theta)/r^2//illuminance at point A

r1=d1//for point B
theta1=acosd(d/d1)//for point B
EB=I*cosd(theta1)/r1^2//illuminance at point B

printf('the illuminance at a small surface area of
   the table-top directly below the surface is %d
   lux',round(EA))
printf('\nthe illuminance at a small surface area of
    the table-top at a distance 0.80 m from the
   source is %d lux',EB)
```

Scilab code Exa 22.5w calculation of the luminous flux emitted into a cone of given solid angle

```
//developed in windows XP operating system 32bit
//platform Scilab 5.4.1
clc;clear;
//example 22.5w
//calculation of the luminous flux emitted into a
   cone of given solid angle

//given data
I0=160//luminous intensity(in candela) of small
   plane source
deltaw=0.02//solid angle(in sr)
theta=60//angle(in degree) made by the centre line
   of the cone with the forward normal

//calculation
I=I0*cosd(theta)//by using Lambert's cosine law
deltaF=I*deltaw//luminous flux
```

```
printf('the luminous flux emitted into a cone of solid angle 0.02 sr around a line making an angle of 60 degree with the forward normal is %3.1f lumen',deltaF)
```