# Scilab Textbook Companion for Engineering Physics by G. S. Raghuvanshi[1]

Created by
Dakshee Lahkar
B.Tech
Electronics and Telecommunication Engineering
Assam Engineering College
College Teacher
None
Cross-Checked by
None

May 2, 2019

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

# Book Description

**Title:** Engineering Physics

**Author:** G. S. Raghuvanshi

**Publisher:** Prentice Hall Of India, New Delhi

**Edition:** 1

**Year:** 2008

**ISBN:** 9788120334304

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

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

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

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

# Contents

# List of Scilab Codes

# Chapter 1

# Interference of Light

**Scilab code Exa 1.1** calculation of position of zero order and twentieth order frin

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.1
//calculation of position of zero order and twentieth order fringe
//given data
lambda=6000; //wavelength (in Armstrong) of light
lambda1=5000; //new wavelength (in Armstrong) of light
n=10; //order of maxima
n1=20; //order of maxima
D=1; //assuming D=1 for simplicity of calculation
d=1; //assuming d=1 for simplicity of calculation
y0=12.34; //position of zero order maxima
y10=14.73; //position of tenth order maxima
//calculation
y_bn=y10-y0; //difference between position of tenth order maxima and zero order maxima (in mm)
y_bn_dash=y_bn*(n1*lambda1*(D/d))/(n*lambda*(D/d)); //position of central bright fringe (in mm)
```

```
y0_dash=y0; //position (in mm) of zero order fringe
y_b20_dash=y0_dash+y_bn_dash; //position (in mm) of twentieth order fringe
printf('\nposition of zero order fringe is %0.2f mm',y0_dash)
printf('\nposition of twentieth order fringe is %0.2f mm',y_b20_dash)
//the answers vary due to round off error
```

**Scilab code Exa 1.2** calculation of wavelength of light used in Fresnel biprism exp

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.2
//calculation of wavelength of light used in Fresnel's biprism experiment
//given data
W=0.196; //fringe width (in mm)
D=1.00*10^3; //distance from the slit(in mm)
//in the codes below, Dn+m is considered as D_n1
d1=6.00; //separation between images when convex lens was placed at one place between biprism and eye piece (in mm)
d2=1.5; //separation between images when convex lens was placed at another place between biprism and eye piece (in mm)
//calculation
d=sqrt(d1*d2); //actual separation (in mm)
format (16)
lambda=(W*d)/D; //wavelength (in mm) of light
disp(lambda*10^7,'wavelength (in Armstrong) of light is')
```

**Scilab code Exa 1.3** calculation of separation of the coherent slit images and frin

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.3
//calculation of (i)separation of the coherent slit
   images (ii) fringe width
//given data
lambda=5893*10^-10; //wavelength (in m)
mu=1.50; //refractive index of biprism
alpha1=1.04; //refracting angle (in degrees)
alpha2=1.23; //refracting angle (in degrees)
b=56.1; //distance (in cm) of focal planes of
   eyepiece from the biprism
a=12.4; //distance (in cm) from slit to the biprism
//calculation
d=a*(mu-1)*(alpha1+alpha2)/180*3.14; //separation of
    coherent slit images(in cm)
D=(a+b); //distance from slit(in cm)
W=(lambda*D*10^-2)/(d*10^-2); //fringe width (in m)
printf("\nseparation of coherent slit images is %0.4
   f cm",d)
printf("\nfringe width is %0.4f mm",W*10^3)
```

**Scilab code Exa 1.5** calculation of refractive index of the film material

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.5
```

```
//calculation of refractive index of the film
    material
//given data
t=6.3*10^-4*10^-2; //thickness of thin sheet of mica
     (in m)
lambda= 5460*10^-10; //wavelength of light (in m)
n=6; //sixth bright fringe
//calculation
mu=1+(n*lambda)/t; //refractive index of the film
    material
disp(mu,'refractive index of the film material is')
```

**Scilab code Exa 1.6** calculation of shift of fringe system and whether shift is tow

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.6
//calculation of shift of fringe system
//given data
lambda=5893; //wavelength of sodium light
W=0.347; //fringe width (in mm)
tA=0.016; //thickness of sheet A (in mm)
tB=0.02; //thickness of sheet B (in mm)
muA=1.65; //refractive index of sheet A
muB=1.45; //refractive index of sheet B
//calculation
xA=(muA-1)*tA; //path difference introduced by film
    A (in mm)
xB=(muB-1)*tB; //path difference introduced by film
    B (in mm)
n=(xA-xB)/(lambda*10^-7); //number of fringe
s=n*W; //shift in fringe system (in mm)
printf("\nshift in fringe system is %0.3f mm",s)
if xA>xB then
```

```
    disp("pattern will shift towards A")
else
    disp("pattern will shift towards B")
end
```

**Scilab code Exa 1.7** `calculation of thickness of a plate`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.7
//calculation of thickness of a plate
//given data
lambda=6000*10^-10; //wavelength (in m) of light
mu=1.50; //refractive index of glss plate
n=6; //sixth bright fringe
//calculation
t=(n*lambda)/(mu-1); //thickness (in m) of the plate
printf("\nthickness of the plate is %1.4f mm",t*10^3)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 1.8** `calculation of fringe width`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.8
//calculation of fringe width
//given data
lambda= 6000*10^-10; //wavelength (in m) of monochromatic light
L=0.15; //distance (in m) from the edge of the wedge
```

```
h=0.03*10^-3; //diameter of wire (in m)
mu=1; //refractive index of air
//calculation
W=(lambda*L)/h; //fringe width (in m)
printf('\nfringe width  is %1.0e m',W)
```

**Scilab code Exa 1.9** calculation of wavelength of light used

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.9
//calculation of wavelength of light used
//given data
mu=1.4; //refractive index of wedge
theta=40; //angle of wedge (in seconds)
W=1.25*10^-3; //distance between successive fringes
    (in m)
//calculation
lambda=2*mu*(theta*3.14)/(3600*180)*W; //wavelength
    of light used (in m)
printf('\nwavelength of light used is %1.0e m',
    lambda)
```

**Scilab code Exa 1.10** calculation of depth of scratches

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.10
//calculation of depth of scratches
//given data
lambda=5350*10^-10; //wavelength (in m) of light
```

```
//calculation
t=4/10*lambda/2; //depth (in m) of scratches
printf("\ndepth of scratches is %0.1f micrometer",t*10^6)
```

**Scilab code Exa 1.11** calculation of thickness of film

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.11
//calculation of thickness of film
//given data
lambda1=6.1*10^-5; //wavelength (in cm) of light
lambda2=6*10^-5; //wavelength (in cm) of light
mu=4/3; //refractive index for film
i=asin(4/5); //angle of incidence
//calculation
t=(lambda1*lambda2)/(lambda1-lambda2)*1/(2*sqrt(mu^2-sin(i)^2))
printf('\nthickness of film is %0.4f cm',t)
```

**Scilab code Exa 1.12** calculation of wavelengths which will be in visible region

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.12
//calculation of wavelengths which will be in visible region (4000 Armstrong-8000 Armstrong)
//given data
t=5*10^-7; //thickness (in m) of film of soapy water
mu=1.33; //refractive index of soapy water
```

```
r=0; //angle (in degrees) at which white light falls
    on a film of soapy water
n1=1; //1st order
n2=2; //2nd order
n3=3; //3rd order
n4=4; //4th order
//calculation
lambda1=(2*mu*t*cosd(r))/(n1-1/2); //wavelength (in
   m) of light in 1st order
lambda2=(2*mu*t*cosd(r))/(n2-1/2); //wavelength (in
   m) of light in 2nd order
lambda3=(2*mu*t*cosd(r))/(n3-1/2); //wavelength (in
   m) of light in 3rd order
lambda4=(2*mu*t*cosd(r))/(n4-1/2); //wavelength (in
   m) of light in 4th order
printf("\nwavelength of light in 1st order is %d
    Armstrong",lambda1*10^10)
printf("\nwavelength of light in 2nd order is %d
    Armstrong",lambda2*10^10)
printf("\nwavelength of light in 3rd order is %d
    Armstrong",lambda3*10^10)
printf("\nwavelength of light in 4th order is %d
    Armstrong",lambda4*10^10)
printf("\namongst these, the wavelength which falls
    in the visible region (4000-8000)Armstrong is
    5319 Armstrong")
//the second and third wavelength vary due to round
    off error
```

---

**Scilab code Exa 1.15** calculation of wavelength of light

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.15
```

```
//calculation of wavelength of light
//given data
del_h=2.2*10^-6; //distance (in m) raised by the
    lens
n_dash=15; //fifteenth bright ring
n=7; //seventh bright ring
//calculation
lambda=(2*del_h)/(n_dash-n); //wavelength (in m) of
    light
printf("\nwavelength of light used is %d Armstrong",
    lambda*10^10)
```

**Scilab code Exa 1.16** calculation of new radius of ring when lens is raised

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.16
//calculation of new radius of ring when lens is
    raised
//given data
R=40; //radius of curvature of lens (in cm)
r_dn=2.5; //radius of ring (in mm)
del_h=5.0; //distance of lens raised from plate (in
    micrometer)
//calculation
r_d1n=sqrt(r_dn^2-2*del_h*10^-3*R*10^1); //new
    radius of ring (in mm)
disp(r_d1n,'new radius (in mm) of ring when lens is
    raised is')
```

**Scilab code Exa 1.17** calculation of radius of curvature of lens and thickness of a

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.17
//calculation of radius of curvature of lens and thickness of air film at the ring
//given data
lambda=5.9*10^-5; //wavelength of light (in cm)
D_d10=0.50; //diameter of tenth ring (in cm)
n=10; //number of dark ting
//calculation
R=(D_d10^2)/(4*n*lambda); //radius of curvature of lens (in cm)
t_dn=(n*lambda)/2; //thickness of the air film at the lens (in cm)
printf('\nradius of curvature of lens is %0.1f cm',R)
printf('\nthickness of film at the ring is %1.2e cm',t_dn)
```

---

**Scilab code Exa 1.18** calculation of the order of the dark ring which will have dou

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.18
//calculation of the order of the dark ring which will have double the diameter of that of the 20th dark ring
D_nsq=1; //assuming the square of the diameter of nth ring to be 1 for simplicity of calculation
D_20sq=(D_nsq)/4; //given that nth dark ring which will have double the diameter of that of the 20th dark ring (D_20sq is square of the diameter of 20th ring and D_nsq is square of the diameter of
```

```
    nth ring)
n_20=20; //order of the 20th ring
//from the formula D_n=sqrt(4*n*lambda*R), D_nsq is
    directly proportional to n
n=n_20*(D_nsq)/(D_20sq); //order of the dark ring
    which will have double the diameter of that of
    the 20th dark ring
printf("\norder of the dark ring which will have
    double the diameter of that of the 20th dark ring
     is %d",n)
```

**Scilab code Exa 1.19** calculation of n and diameter of the ring

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.19
//calculation of n and diameter of the ring
//given data
lambda1=6560; //first wavelength of light (in
    Armstrong)
lambda2=5248; //second wavelength of light (in
    Armstrong)
R=0.8; //radius of curvature (in m)
//calculation
n=lambda2/(lambda1-lambda2); //number of dark ring
D_n=sqrt(4*n*lambda1*10^-10*R); //diameter of the
    ring (in m)
disp(n,'n is')
printf("\ndiameter of ring is %0.3f mm",D_n*10^3)
```

**Scilab code Exa 1.20** calculation of diameter of 25th and 37th bright rings and wav

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.20
//calculation of diameter of 25th and 37th bright
    rings and wavelength of light
//given data
D_n=0.314; //diameter of 5th bright ring (in cm)
D_n1=0.584; //diameter of 16th bright ring (in cm)
R=120; //radius of curvature (in cm)
n=5; //number of bright ring
n1=16; //number of bright ring
n2=25; //number of bright ring
n3=37; //number of bright ring
//calculation
m=n1-n; //difference between the number of bright
    rings
lambda=(D_n1^2-D_n^2)/(4*m*R); //wavelength of light
     used (in cm)
r_b25=sqrt((n2-1/2)*lambda*R); //diameter of 25th
    ring (in cm)
r_b37=sqrt((n3-1/2)*lambda*R); //diameter of 37th
    ring (in cm)
printf('\nwavelength of light used is %0.2f
    Armstrong',lambda*10^8)
printf('\ndiameter of 25th ring is %0.2f mm',r_b25
    *10)
printf('\ndiameter of 37th ring is %0.2f mm',r_b37
    *10)
//answer provided in the textbook is wrong
```

**Scilab code Exa 1.21** calculation of wavelength of light used

```
clc;clear;
//OS windows 7
```

```
//scilab 6.0.1
//example 1.21
//calculation of wavelength of light used
//given data
D15=0.59; //diameter of 15th ring (in cm)
D5=0.336; //diameter of 5th ring (in cm)
R=100; //radius of curvature (in cm)
n=15; //fifteenth ring
n1=5; //fifth ring
//calculation
m=n-n1; //difference between the number of rings
lambda=(D15^2-D5^2)/(4*R*m); //wavelength of light used (in cm)
printf('\nwavelength of light is %d Armstrong', lambda*10^8)
//answer varies due to round off error
```

**Scilab code Exa 1.22** calculation of diameter of 20th ring and order of dark ring w

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.22
//calculation of diameter of 20th ring and order of dark ring when thickness is equal to wavelength
//given data
n=4; //fourth ring
n1=12; //seventh ring
D_n=0.4; //diameter (in cm) of 4th ring
D_n1=0.7; //diameter (in cm) of 12th ring
t_n=1; //assuming thickness to be 1 for simplicity of calculation
lambda=1; //assuming wavelength to be 1 for simplicity of calculation
//calculation
```

```
m=n1-n; //difference between the number of rings
lambdaR=(D_n1^2-D_n^2)/(4*m); // for simplicity of calculation lamnbda*R is taken as one variable
n2=20; //twentieth ring
D_20d=sqrt(4*n2*lambdaR); //diameter (in cm) of twentieth ring
n=(2*t_n)/lambda; //order of dark ring when thickness is equal to wavelength
printf('\ndiameter of twentieth ring is %0.2f cm',D_20d)
printf('\norder of dark ring is %d',n)
//answer provided in the textbook is wrong
```

**Scilab code Exa 1.23** calculation of wavelength of light used

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.23
//calculation of wavelength of light used
//given data
D_n=4.2*10^-3; //diameter of nth dark ring (in m)
D_n_plus_10=7.0*10^-3; //diameter of (n+10)th dark ring (in m)
R=2; //radius of curvature (in m)
//calculation
m=10;
lambda=((D_n_plus_10^2)-(D_n^2))/(4*m*R); //wavelength of light used (in m)
disp(lambda*10^10,'Wavelength of light (in Armstrong) used is')
```

**Scilab code Exa 1.24** calculation of radius of curvature of lens

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.24
//calculation of radius of curvature of lens
//given data
lambda=5*10^-7; //wavelength of light (in m)
D_10d=0.5*10^-2; //diameter of 10th dark ring (in m)
n=10; //number of dark ring
//calculation
R=D_10d^2/(4*n*lambda); //radius of curvature (in m)
disp(R,'radius of curvature (in m) is')
```

**Scilab code Exa 1.25** calculation of refractive index of a liquid

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.25
//calculation of refractive index of a liquid
//given data
lambda=5890; //wavelength of reflected light (in
    Armstrong)
n=5; //number of dark ring
D_n=0.32; //diameter of 5th ring (in cm)
R=1.20; //radius of curvature (in m)
//calculation
mu=(4*n*lambda*10^-10*R)/(D_n*10^-2)^2; //refractive
     index of the liquid
printf("\nrefractive index of the liquid is %0.2f",
    mu)
```

**Scilab code Exa 1.26** calculation of diameter of nth ring

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.26
//calculation of diameter of nth ring
//given data
lambda1=6000*10^-10; //wavelength (in m)
lambda2=4500*10^-10; //another wavelength (in m)
R=0.9; //radius of curvature (in m)
//calculation
n=4*lambda2*R/(4*lambda1*R-4*lambda2*R); //order of
    ring
D=sqrt(4*n*lambda1*R); //diameter (in m) of nth ring
printf('\ndiameter of nth ring is %0.4f cm',D*10^2)
//The answers vary due to round off error
```

**Scilab code Exa 1.27** calculation of distance between 10th and 20th rings

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.27
//calculation of distance between  10th and 20th
    rings
//given data
R1=1.0; //radius of curvature (in m)
R2=1.0; //another radius of curvature (in m)
lambda=600*10^-9; //wavelength of light (in m)
n=10; //order of ring
n_dash=20; //order of ring
//calculation
r10=sqrt((n*lambda*R1*R2)/(R1+R2)); //radius (in m)
    of 10th ring
r20=sqrt((n_dash*lambda*R1*R2)/(R1+R2)); //radius (
    in m) of 20th ring
```

```
d=r20-r10; //difference between 10th and 20th rings (in m)
printf('\ndifference between 10th and 20th rings is %0.3f mm',d*10^3)
```

**Scilab code Exa 1.29** calculation of diameter of nth dark ring

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.29
//calculation of diameter of nth dark ring
//given data
lambda1=6000*10^-10; //wavelength (in m) of light
lambda2=5000*10^-10; //wavelength (in m) of light
R=0.90; //radius of curvature (in m)
//calculation
n=(lambda2*R)/((lambda1-lambda2)*R); //order of dark ring
D_n=sqrt(4*n*lambda1*R); //diameter (in m) of nth ring
printf("\nThe diameter of nth ring is %0.3f mm",D_n*10^3)
//the answers vary due to round off error
```

**Scilab code Exa 1.31** calculation of difference of squares of diameters of successi

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.31
//calculation of difference of squares of diameters of successive rings if (a) wavelength of light is
```

```
    changed to 4.5*10^-5 cm (b)liquid of refractive
    index 1.33 is introduced between the lens and the
    plate (c) the plane glass plate is replaced by a
    planocomcave lens of radius of curvature twice
    that of the planoconvex lens (d) the plane glass
    plate is replaced by planoconvex lens identical
    to one and put on the top of it
//given data
lambda1=6*10^-5; //wavelength (in cm) of light
lambda2=4.5*10^-5; //new wavelength (in cm) of light
D1sq_minus_D2sq=0.125; //difference of squares of
    diameters (in cm^2)
mu_liq=1.33; //refractive index of liquid
mu=1; //refractive index of air
m=1; //difference between the number of rings
//calculation
//in the codes, Dn+m is considered as D1 and Dn as
    D2
lambda4R=(D1sq_minus_D2sq*mu)/m; //assuming lambda4R
    as one variable for simplicity of calculation
D1sq_dash_minus_D2sq_dash_lambda2=D1sq_minus_D2sq*
    lambda2/lambda1; //difference of squares of
    diameters of successive rings if wavelength of
    light is changed (in cm^2)
D1sq_minus_D2sq_liquid=D1sq_minus_D2sq/mu_liq; //
    difference of squares of diameters of successive
    rings if liquid of refractive index is introduced
    (in cm^2)
D1sq_dash_minus_D2sq_dash_R=2*lambda4R; //difference
    of squares of diameters of successive rings if
    radius of curvature becomes twice (in cm^2)
D1sq_dash_minus_D2sq_dash_t=1/2*(lambda4R); //
    difference of squares of diameters of successive
    rings when thickness is changed (in cm^2)
printf("\n(a)difference of squares of diameters of
    successive rings if wavelength of light is
    changed is %0.3f cm^2",
    D1sq_dash_minus_D2sq_dash_lambda2)
```

```
printf("\n(b)difference of squares of diameters of successive rings if liquid of refractive index is introduced is %0.3f cm^2",D1sq_minus_D2sq_liquid)
printf("\n(c)difference of squares of diameters of successive rings if radius of curvature becomes twice is %0.3f cm^2",D1sq_dash_minus_D2sq_dash_R)
printf("\n(d)difference of squares of diameters of successive rings when thickness is changed is %0.4f cm^2",D1sq_dash_minus_D2sq_dash_t)
```

**Scilab code Exa 1.32** calculation of number of bright fringes

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.32
//calculation of number of bright fringes
//given data
lambda=589.00; //wavelength (in nm) of yellow light
x=1.0000; //distance (in cm) moved by the mirror
//calculation
n=round((2*x*10^-2)/(lambda*10^-9)); //number of bright fringes
printf("\nthe number of bright bringes are %d",n)
```

**Scilab code Exa 1.33** calculation of change in path length

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.33
//calculation of change in path length
```

```
//given data
lambda=5000*10^-10; //wavelength (in m) of light
n=50; //number of fringes
//calculation
x=(n*lambda)/2; //change in path length (in m)
printf("\nchnge in path length is %1.4f mm",x*10^3)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 1.34** calculation of angular radius of 10th bright fringe and chang

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.34
//calculation of (i)angular radius of 10th bright fringe (ii) change of pathlength
//given data
m=10; //tenth bright fringe
lambda=5000*10^-10; //wavelength (in m) of light
d=2.5*10^-3; //difference in the pathlengths (in m)
n=60; //number of fringes
//calculation
theta=acosd(1-(lambda*(m-1))/(2*d)); //angular radius (in degrees)
del_d=n/2*lambda; //change of pathlength (in m)
printf("\n(i)angular radius is %0.2f degrees",theta)
printf("\n(ii)change of pathlengh is %0.3f mm",del_d*10^3)
```

**Scilab code Exa 1.35** calculation of difference between wavelengths

```
clc;clear;
//OS windows 7
```

```
//scilab 6.0.1
//example 1.35
//calculation of difference between wavelengths
//given data
x=0.02945*10^-3; //distance (in m) moved by mirror
lambda=5893*10^-10; //wavelength (in m) of light
//calculation
del_lambda=lambda^2/(2*x); //difference between
    wavelengths (in m)
printf("\ndifference between wavelengths is %1.2f
    Armstrong",del_lambda*10^10)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 1.36** calculaton of wavelength of light

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.36
//calculaton of wavelength of light
//given data
x=0.0589*10^-3; //distance (in m) moved by the
    mirror
n=200; //number of fringes
//calculation
lambda=(2*x)/n; //wavelength (in m) of light
printf("\nwavelength of light is %d nm",lambda*10^9)
```

**Scilab code Exa 1.37** calculation of angular diameter of tenth bright fringe

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 1.37
//calculation of angular diameter of tenth bright
    fringe
//given data
m=10; //tenth bright fringe
lambda=5896*10^-10; //wavelength (in m) of length
d=3*10^-3; //difference between path lengths
//calculation
theta=acosd(1-(lambda*(m-1))/(2*d)); //angular
    radius (in degrees)
twice_theta=2*theta; //angular diameter (in degrees)
printf("\nangular diameter is %0.2f degrees",
    twice_theta)
//the answers vary due to round off error
```

---

**Scilab code Exa 1.38** calculation of pressure within the tube

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.38
//calculation of pressure within the tube
//given data
l=5*10^-2; //length (in m) of tube
lambda=589.3*10^-9; //wavelength (in m) of sodium
    light
n=10; //number of fringes
//calculation
//given that refractive index, mu=1+3*10^-4*p, where
     p is pressure in atm
//from the formula 2*(mu-1)*l=n*lambda
p=(n*lambda-1+1)/(2*l*3*10^-4); //pressure (in atm)
    within the tube (by replacing the value of mu by
    1+3*10^-4*p)
printf("\npressure within the tube is %1.4f atm",p)
```

**Scilab code Exa 1.39** calculation of distance between successive positions of movab

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.39
//calculation of distance between successive
   positions of movable mirror
//given data
lambda1=5896*10^-10; //wavelength (in m) of light
lambda2=5890*10^-10; //another wavelength (in m) of
   light
//calculation
del_lambda=lambda1-lambda2; //difference of
   wavelengths (in m)
d=(lambda1*lambda2)/(2*del_lambda); //distance
   between successive positions of movable mirror (
   in m)
printf("\ndistance between successive positions of
   movable mirror is %0.2f Armstrong",d*10^10)
//the answer provided in the book is wrong
```

**Scilab code Exa 1.40** calculation of distance to be moved by the mirror that fringe

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.40
//calculation of distance to be moved by the mirror
   that fringes disappear
//given data
```

```
lambda1=4882*10^-10; //first wavelength (in m) of
    light
lambda2=4886*10^-10; //second wavelength (in m) of
    light
//calculation
lambda_av=(lambda1+lambda2)/2; //average wavelength
    (in m)
del_lambda=lambda2-lambda1; //difference of
    wavelength (in m)
d=lambda_av^2/(4*del_lambda); //distance (in m) to
    be moved by the mirror that fringes disappear
printf("\ndistance to be moved by the mirror that
    fringes disappear is %0.3f mm",d*10^3)
```

**Scilab code Exa 1.41** calculation of thickness of plate

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.41
//calculation of thickness of plate
//given data
mu=1.5; //refractive index of glass plate
n=100; //number of fringes
lambda=6328*10^-10; //wavelength (in m) of laser
format(16)
t=(n*lambda)/(2*(mu-1)); //thickness (in m) of the
    plate
printf("\nthickness of the plate is %1.3e m",t)
```

**Scilab code Exa 1.42** calculation of minimum thickness of a layer of cryolite and c

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 1.42
//calculation of minimum thickness of a layer of
    cryolite and change in wavelength
//given data
mu=1.35; //refractive index of cryolite
lambda0=5940*10^-10; //wavelength (in m) of light
n=1
mu0=1; //refractive index of air
i=10; //angle of incidence (in degrees)
//calculation
t=(n*lambda0)/(2*mu); //minimum thickness of a layer
     of cryolite (in m)
lambda=lambda0*sqrt(1-(sind(i)^2)/mu^2); //
    transmitted wavelength (in m)
del_lambda=lambda0-lambda; //change in wavelength (
    in m)
printf("\nminimum thickness of a layer of cryolite
    is %d Armstrong",t*10^10)
printf("\nchange of wavelength is %d Armstrong",
    del_lambda*10^10)
//answer varies due to round off error
```

**Scilab code Exa 1.43** calculation of refractive index of second layer

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 1.43
//calculation of refractive index of second layer
//given data
mu_1=1.38; //refractive index of first layer
mu_air=1; //refractive index of air
mu_g=1.52; //refractive index of glass
```

```
//calculation
mu_2=mu_1*sqrt(mu_g/mu_air); //refractive index of
    second layer
printf("\nrefractive index of second layer is %0.2f"
    ,mu_2)
```

# Chapter 2

# Polarization of Light

**Scilab code Exa 2.3** calculation of angle either sheet must be turned

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.3
//calculation of angle either sheet must be turned
//given data
I0=1; //assuming I0 to be 1 for simplicity of
    calculation
I=I0/2; //intensity is half the initial intensity
//calculation
theta=acosd(sqrt(I/I0)); //from Malus law, I=I0cos
    ^2(theta)
theta1=acosd(-sqrt(I/I0)); //from Malus law, I=I0cos
    ^2(theta)
printf("\nangle either sheet must be turned is %d
    degree",theta)
printf("\nor")
printf("\nangle either sheet must be turned is %d
    degree",theta1)
```

**Scilab code Exa 2.6** calculation of degree of polarization

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.6
//calculation of degree of polarization
I0=1; //assuming maximum intensity to be 1 for
    simplicity of calculation
I=75/100; //intensity change
I_max=I0; //maximum intensity
I_min=I0-I*I0; //minimum intensity
D_p=(I_max-I_min)/(I_max+I_min); //degree of
    polarization
printf("\ndegree of polarization is %d percent",D_p
    *100)
```

---

**Scilab code Exa 2.7** calculation of intensity of light emerging out of Nicol B

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.7
//calculation of intensity of light emerging out of
    Nicol B
//given data
theta=30; //angle (in degrees) made by Nicol C with
    Nicol A
I0=32; //intensity (in W/m^2) of unpolarized light
//calculation
I_t=I0/8*(sind(2*theta)); //intensity (in W/m^2) of
    light emerging out of Nicol B
```

```
printf("\nintensity of light emerging out of Nicol B
    is %d W/m^2",I_t)
```

**Scilab code Exa 2.8** calculation of ratio of intensities of two beams

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.8
//calculation of ratio of intensities of two beams
theta1=90-30; //rotation of B (in degrees)
theta2=90-60; //rotation of A (in degrees)
IA_by_IB=cosd(theta1)^2/cosd(theta2)^2; //ratio of
    intensities of two beams
printf("\nratio of intensities of two beams is %0.2f
    ",IA_by_IB)
```

**Scilab code Exa 2.10** calculation of thickness of a quarter wave plate

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.10
//calculation of thickness of a quarter wave plate
//given data
mu_o=1.553; //refractive index of ordinary light
mu_e=1.544; //refractive index of extraordinary
    light
lambda=5000*10^-10; //wavelength (in m) of light
//calculation
t_QWP=lambda/(4*(mu_o-mu_e)); //thickness (in m) of
    a quarter wave plate
```

```
printf("\nthickness of a quarter wave plate is %2.2f micrometre",t_QWP*10^6)
//the value of refractive index of extraordinary light is given different in the question than the calculation
```

**Scilab code Exa 2.11** calculation of nature of retardation plate

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.11
//calculation of nature of retardation plate
//given data
t=8.56*10^-7; //thickness (in m) of calcite plate
lambda=5890*10^-10; //wavelength (in m) of light
mu_o=1.658; //refractive index of extraordinary light
mu_e=1.486; //ordinary index of ordinary light
//calculation
delta_by_lambda=((mu_o-mu_e)*t)/lambda; //path difference
printf("\npath difference creted by plate is %1.1f",delta_by_lambda)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 2.12** calculation of wavelengths in visible region for which it wil

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.12
```

```
//calculation of wavelengths in visible region for
    which it will act as (i)QW plate (ii)HW plate
//given data
l=0.1436*10^-3; //thickness (in m) of plate
mu_o=1.5443; //refractive index of ordinary light
mu_e=1.5533; //refractive index of extraordinary
    light
n1=3
n2=4
n3=5
n=2
//calculation
//for QWP
lambda1=(4*l*(mu_e-mu_o))/(2*n1+1)
lambda2=(4*l*(mu_e-mu_o))/(2*n2+1)
lambda3=(4*l*(mu_e-mu_o))/(2*n3+1)
//for HWP
lambda=(2*l*(mu_e-mu_o))/(2*n+1)
printf("\n(a)visible wavelength when n=3 is %d
    Armstrong",lambda1*10^10)
printf("\n(a)visible wavelength when n=4 is %d
    Armstrong",lambda2*10^10)
printf("\n(a)visible wavelength when n=5 is %d
    Armstrong",lambda3*10^10)
printf("\n(b)visible wavelength when n=2 is %d
    Armstrong",lambda*10^10)
//the second part of answer (a) is given wrong in
    the textbook
//the third part of answer (a) varies due to round
    off error
//the answer (b) varies due to round off error
```

**Scilab code Exa 2.15** calculation of wavelengths in visible region for which it wil

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 2.15
//calculation of wavelengths in visible region for
   which it will act as (i)HW plate (ii)QW plate
//given data
l=0.25*10^-3; //thickness (in m)
mu_e_minus_mu_o=0.009; //difference between the
   refractive indices of ordinary and extraordinary
    light
n1=3
n2=4
n3=6
n4=7
n5=8
n6=9
//calculation
//for HWP
lambda1=(2*l*(mu_e_minus_mu_o))/(2*n1+1)
lambda2=(2*l*(mu_e_minus_mu_o))/(2*n2+1)
//for QWP
lambda3=(4*l*(mu_e_minus_mu_o))/(2*n3+1)
lambda4=(4*l*(mu_e_minus_mu_o))/(2*n4+1)
lambda5=(4*l*(mu_e_minus_mu_o))/(2*n5+1)
lambda6=(4*l*(mu_e_minus_mu_o))/(2*n6+1)
printf("\n(a)visible wavelength when n=3 is %d
   Armstrong",lambda1*10^10)
printf("\n(a)visible wavelength when n=4 is %d
   Armstrong",lambda2*10^10)
printf("\n(b)visible wavelength when n=6 is %d
   Armstrong",lambda3*10^10)
printf("\n(b)visible wavelength when n=7 is %d
   Armstrong",lambda4*10^10)
printf("\n(b)visible wavelength when n=8 is %d
   Armstrong",lambda5*10^10)
printf("\n(b)visible wavelength when n=9 is %d
   Armstrong",lambda6*10^10)
//the answer (b) when n=9 varies due to round off
```

```
   error
```

**Scilab code Exa 2.16** calculation of thickness of a calcite plate which would conve

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.16
//calculation of thickness of a calcite plate which
   would convert plane polarized light into
   circularly polarized light
//given data
mu_o=1.658; //refractive index of o-ray
mu_e=1.486; //refractive index of e-ray
lambda=5890*10^-10; //wavelength (in m) of light
//calculation
l_QWP=lambda/(4*(mu_o-mu_e)); //thickness (in m) of
   quarter wave plate
printf("\nthickness of a calcite plate which would
   convert plane polarized light into circularly
   polarized light is %1.3f micrometer or its odd
   multiple",l_QWP*10^6)
```

**Scilab code Exa 2.18** calculation of maximum thickness of the plate to to experienc

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.18
//calculation of maximum thickness of the plate to (
   i)to experience only rotation of polarization
   plane (ii)to acquire circular polarization after
   passing through the plate
```

```
//given data
lambda=5890*10^-10; //wavelength (in m)
mu_e_minus_mu_o=0.009; //difference of refractive indices of quartz
n=1
t=5*10^-4; //thickness (in m)
//calculation
twice_n_plus_1=round((t*2*mu_e_minus_mu_o)/lambda)
I_max=(twice_n_plus_1*lambda)/(2*mu_e_minus_mu_o)
twice_n_plus_one=(t*4*mu_e_minus_mu_o)/lambda
I_max_dash=(twice_n_plus_one*lambda)/(4*mu_e_minus_mu_o)
printf("\n(i)maximum thickness of the plate to experience only rotation of polarization plane is %1.2f mm",I_max*10^3)
printf("\n(ii)maximum thickness of the plate to experience only rotation of polarization plane is %1.2f mm",I_max_dash*10^3)
//the answer (ii) varies due to round off error
```

**Scilab code Exa 2.19** calculation of least thickness of the plate for which the eme

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.19
//calculation of least thickness of the plate for which the emergent beam will be (i)plane polarized (ii)circularly polarized
//given data
mu_e=1.5533; //refractive index of extraordinary light
mu_o=1.5442; //refractive index of ordinary light
lambda=5000*10^-10; //wavelength (in m) of light
n=1; //taking 1st order
```

```
//calculation
t=((n+1/2)*lambda)/(mu_e-mu_o); //thickness (in m) of the plate for which the emergent beam will be plane polarized
t_QWP=lambda/(4*(mu_e-mu_o)); //thickness (in m) of the plate for which the emergent beam will be circularly polarized
printf("\n(i)thickness of the plate for which the emergent beam will be plane polarized is %2.1f micrometer or its odd multiple",t*10^6)
printf("\n(ii)thickness of the plate for which the emergent beam will be circularly polarized is %2.2f micrometer or its odd multiple",t_QWP*10^6)
//the first part of the answer is given wrong in the textbook
//the second part of the answer varies due to round off error
```

---

**Scilab code Exa 2.20** calculation of minimum thickness of plate required to produce

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.20
//calculation of minimum thickness of plate required to produce (i)zero outgoing intensity (ii) intensity of incident light
//given data
mu_0=1.658; //refractive index of ordinary light
mu_e=1.486; //refractive index of extra ordinary light
lambda=5893*10^-10; //wavelength (in m) of sodium light
//calculation
l=lambda/(mu_0-mu_e); //minimum thickness (in m) of
```

```
plate required to produce zero outgoing intensity
l_dash=lambda/(4*(mu_0-mu_e)); //minimum thickness (in m) of plate required to produce intensity of incident light
printf("\n(i)minimum thickness of plate required to produce zero outgoing intensity is %1.2f micrometre",l*10^6)
printf("\n(i)minimum thickness of plate required to produce intensity of incident light is %1.3f micrometre",l_dash*10^6)
//the answers vary due to round off error
```

**Scilab code Exa 2.29** calculation of angle of rotation

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.29
//calculation of angle of rotation
//given data
l=2; //length of tube (in dm)
c=15/100; //concentration of water (in gm/cc)
S_T_lambda=66.5; //specific rotation of sugar (in (decimeter^-1)(gm/cc)^-1)
//calculation
theta=S_T_lambda*l*c; //optical rotation (in degrees)
printf("\noptical rotation is %2.2f degrees",theta)
```

**Scilab code Exa 2.30** calculation of concentration in a solution

```
clc;clear;
//OS windows 7
```

```
//scilab 6.0.1
//example 2.30
//calculation of concentration in a solution
//given data
//first case
theta=20; //rotation (in degrees)
l=1; //path length (in m)
c=20; //concentration of solution (in gm/litre)
//second case
theta_dash=33; //rotation (in degrees)
l_dash=0.5; //path length (in m)
//calculation
c_dash=(l*c*theta_dash)/(l_dash*theta); //
    concentration of solution (in gm/litre)
printf("\nconcentration of solution is %d gm/litre",
    c_dash)
```

**Scilab code Exa 2.31** calculation of strength of solution

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.31
//calculation of strength of solution
//given data
S=66; //specific rotation (in dm/gm/lit)
theta=11; //angle of rotation of plane of
    polarization (in degrees)
l=2; //length (in m)
//calculation
c=theta/(S*l); //strength of solution (in gm/litre)
printf("\nstrength of solution is %0.3f gm/litre",c)
```

**Scilab code Exa 2.32** calculation of optical rotation

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.32
//calculation of optical rotation
//given data
l1=20; //length of a certain solution (in cm)
l2=30; //length of another solution (in cm)
l=30; //length of a mixture of both the solutions (
    in cm)
r1=+42; //right handed rotation (in degrees)
r2=-27; //left handed rotation (in degrees)
ratio1=1; //ratio of first solution
ratio2=2; //ratio of second solution
//calculation
theta1=r1/l1*l*ratio1/(ratio1+ratio2); //rotation of
    right handed solution
theta2=r2/l2*ratio2/(ratio1+ratio2)*l; //rotation of
    left handed solutioncal rotation is
theta=theta1+theta2; //total optical rotation
printf("\ntotal optical rotation is %d",theta)
if theta>0 then
    disp("3 degree right handed rotation")
else
    disp("3 degree left handed rotation")
end
```

**Scilab code Exa 2.33** calculation of strength of solution

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.33
```

```
//calculation of strength of solution
//given data
theta=13; //optical rotation (in degrees)
S_T_lambda=65; //specific rotation (in degree/dm/(g/
    cc))
l=2; //length (in cm) of tube of sugar solution
//calculation
c=theta/(S_T_lambda*l); //strength of solution (in
    gm/cc)
printf("\nstrength of solution is %1.1f gm/cc",c)
```

**Scilab code Exa 2.34** calculation of rotation of plane of polarization of light

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.34
//calculation of rotation of plane of polarization
    of light
//given data
lambda=7620*10^-10; //wavelength (in m) of light
mu_R=1.53914; //refractive index for right handed
    polarized light
mu_L=1.53920; //refractive index for left handed
    polarized light
l=0.5*10^-3; //thickness (in m) of plate
//calculation
theta=%pi/lambda*l*(mu_L-mu_R)*180/%pi; //optical
    rotation (in degrees)
printf("\nrotation of plane of polarization of light
     is %1.1f degree",theta)
```

**Scilab code Exa 2.35** calculation of angle of rotation of hydrogen red and mercury

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.35
//calculation of angle  of rotation of hydrogen red
   and mercury blue
//given data
lambda_hr=6438; //wavelength (in m) of hydrogen red
   light
lambda_mb=4358; //wavelength (in m) of mercury blue
   light
S_6438=18.02; //specific rotation (in deg/mm) of
   hydrogen red light
S_4358=41.55; //specific rotation (in deg/mm) of
   mercury blue light
S_5893=21.72; //specific rotation (in deg/mm) of
   sodium yellow light
pi=180; //value of pi (in degrees)
//calculation
theta_5893=pi/2; //angle of rotation (in degrees) of
    sodium yellow light
l=theta_5893/S_5893; //path length (in mm)
theta_6438=S_6438*l; //angle of rotation (in degrees
   ) of hydrogen red light
theta_4358=S_4358*l; //angle of rotation (in degrees
   ) of mercury blue light
printf("\nangle of rotation of hydrogen red light is
    %0.1f degrees",theta_6438)
printf("\nangle of rotation of mercury blue light is
    %0.1f degrees",theta_4358)
//the answers vary due to round off error
```

**Scilab code Exa 2.36** calculation of thickness of plate

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 2.36
//calculation of thickness of plate
//given data
theta=90; //rotation (in degrees) of plane of
    vibration of plane polarized light
S=21.72; //specific rotation (in degree/mm)
//calculation
l=theta/S; //thickness (in mm) of plate
printf("\nthickness of plate is %0.2f mm",l)
```

**Scilab code Exa 2.37** calculation of specific rotation for quartz yellow light

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.37
//calculation of specific rotation for quartz yellow
     light
//given data
theta=90; //rotation (in degrees)
l=3.5; //thickness (in mm) of the plate
//calculation
S=theta/l; //specific rotation (in degree/mm)
printf("\nspecific rotation for quartz yellow light
    is %0.2f degree/mm",S)
```

**Scilab code Exa 2.38** calculation of thickness of quartz

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 2.38
//calculation of thickness of quartz
//given data
theta=90; //angle of rotation (in degrees)
S=18; //specific rotation (degree/mm)
//calculation
l=theta/S; //thickness (in mm) of quartz
printf("\nthickness of quartz is %d mm",l)
```

**Scilab code Exa 2.39** calculation of percentage of purity of sugar

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.39
//calculation of percentage of purity of sugar
//given data
S=68; //specific rotation (in degrees) of sugar
l=2; //length (in dm) of tube containing sugar
    solution
theta=10.2; //optical rotation (in degrees)
d=90; //amount of sugar dissolved (in gm)
//calculation
c=theta/(S*l)*1000; //concentration (in gm per litre
    )
p=c/d*100; //per cent purity
printf("\npercentage of purity of sugar is %2.2f
    percent",p)
```

**Scilab code Exa 2.40** calculation of specific rotation

```
clc;clear;
//OS windows 7
```

```
//scilab 6.0.1
//example 2.40
//calculation of specific rotation
//given data
mu_L=1.54427; //refractive index for left handed
    polarization
mu_R=1.54420; //refractive index for right handed
    polarization
lambda=5893*10^-10; //wavelength (in m) of sodium
    light
//calculation
pi=180; //value of pi (in degrees)
S=pi*(mu_L-mu_R)*10^-3/lambda; //specific rotation (
    in deg/mm)
printf("\nspecific rotation is %0.2f deg/mm",S)
```

**Scilab code Exa 2.41** calculation of specific rotation

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 2.41
//calculation of specific rotation
//given data
muL_minus_muR=7*10^-5; //difference of refractive
    indices between left and right circular
    polarizations
lambda=6000*10^-10; //wavelength (in m) of light
pi=180; //value of pi (in degrees)
//calculation
S=pi/lambda*(muL_minus_muR)*10^-3; //specific
    rotation
printf("\nspecific rotation is %d deg/mm",S)
```

# Chapter 3

# Diffraction of Light and Holography

**Scilab code Exa 3.1** `calculation of distances of the first dark band and the next b`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.1
//calculation of distances of the first dark band and the next bright band
//given data
lambda=4890*10^-10; //wavelength (in m) of light
a=0.5*10^-2; //width (in m) of slit
f=40*10^-2; //focal length (in m) of lens
//calculation
y_1_d=lambda/a*f; //position of first dark band
y_1_b=3/2*lambda/a*f; //position of the first bright band next to dark
del_y=y_1_b-y_1_d; //distances of the first dark band and the next bright band
printf("\ndistances of the first dark band and the next bright band is %1.3e m",del_y)
```

**Scilab code Exa 3.2** calculation of width of the slit

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.2
//calculation of width of the slit
//given data
lambda=5000*10^-10; //wavelength (in m) of light
y=5*10^-3; //distance (in m) of first minimum
f=2; //focal length (in m)
//calculation
a=(lambda*f)/y; //width (in m) of the slit
printf("\nwidth of the slit is %1.1f mm",a*10^3)
```

**Scilab code Exa 3.3** calculation of width of slit for which receiver would show zer

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.3
//calculation of width of slit for which receiver
    would show zero intensity
//given data
nu=6000*10^6; //frequency (in Hz)
theta=30; //angle (in degrees)
n1=1
n2=2
n3=3
c=3*10^8; //speed (in m/s) of light
//calculation
lambda=c/nu; //wavelength (in m)
```

```
//for n=1
a1=(n1*lambda)/sind(theta); //width of slit (in m)
//for n=2
a2=(n2*lambda)/sind(theta); //width of slit (in m)
//for n=3
a3=(n3*lambda)/sind(theta); //width of slit (in m)
printf("\nwidth of slit for which receiver would
    show zero intensity when n is 1 is %1.1f",a1)
printf("\nwidth of slit for which receiver would
    show zero intensity when n is 2 is %1.1f",a2)
printf("\nwidth of slit for which receiver would
    show zero intensity when n is 3 is %1.1f",a3)
printf("\nthe width of slit thus increases in the
    same way with the increase in n")
```

**Scilab code Exa 3.4** calculation of lambda1 and lambda2

```
clc;clear;
//example 3.4
//OS windows 7
//scilab 6.0.1
//calculation of lambda1 and lambda2
//given data
f=1; //focal length (in m) of lens
a=0.04*10^-2; //width (in m) of slit
y_d_4=0.5*10^-2; //position (in m) of fourth minima
    corresposding to lambda1
y_d_5=0.5*10^-2; //position (in m) of fifth minima
    corresposding to lambda2
//calculation
lambda1=(y_d_4*a)/(4*f); //wavelength (in m)
    corresponding to fourth minima
lambda2=(y_d_5*a)/(5*f); //wavelength (in m)
    corresponding to fifth minima
printf("\nlambda1 is %d Armstrong",lambda1*10^10)
```

```
printf("\nlambda2 is %d Armstrong",lambda2*10^10)
```

**Scilab code Exa 3.5** `calculation of angular position of first and second minima if`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.5
//calculation of angular position of first and
   second minima if observation point is far away
//given data
lambda=5890*10^-10; //wavelength (in m) of light
a=0.003*10^-3; //width of slit (in m)
n=1; //order of minima
n_dash=2; //order of minima
//calculation
theta_d1=n*lambda/a*180/%pi; //angular position of
   first minima (in degrees) if observation point is
    far away
theta_d2=n_dash*lambda/a*180/%pi; //angular position
    of second minima (in degrees) if observation
   point is far away
theta1=asind((n*lambda)/a); //angular position of
   first minima if the observation point is near
theta2=asind((n_dash*lambda)/a); //angular position
   of second minima if the observation point is near
printf("\nangular position of first minima if
   observation point is far away is %0.2f degrees",
   theta_d1)
printf("\nangular position of second minima if
   observation point is far away is %0.1f degrees",
   theta_d2)
//the answers vary due to round off error
```

**Scilab code Exa 3.6** calculation of width of slit

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.6
//calculation of width of slit
//given data
del_y=0.4*10^-3; //distance (in m) between second
    and fifth minimum
lambda=6000*10^-10; //wavelenth (in m) of light used
f=60*10^-2; //distance (in m) of screen from slit
//calculation
a=(3*lambda*f)/del_y; //width (in m) of slit
printf("\nwidth of slit is %1.1f mm",a*10^3)
```

**Scilab code Exa 3.7** calculation of wavelength of light used

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.7
//calculation of wavelength of light used
//given data
a=0.2; //width of slit (in mm)
f=2; //distance of slit from screen (in m)
y=6; //distance of first minimum on either side of
    central maxima (in mm)
//calculation
lambda=y*10^-3*a*10^-3/f; //wavelength of light (in
    m)
```

```
printf("\nwavelength of light is %d Armstrong",lambda*10^10)
```

**Scilab code Exa 3.10** to show if white light source of 4000 Armstrong to 7000 Armst

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.10
//(i)to show if white light source (4000 Armstrong to 7000 Armstrong) is used, the second and third order spectra overlap (ii) calculation of angular separation of D1 and D2 lines
//given data
N=15000; //number of lines per inch
dlambda=6000*10^-10; //average wavelength of light (in m)
lambda_v=4000*10^-10; //wavelength (in m) of violet light
lambda_r=7000*10^-10; //wavelength (in m) of red light
n=2; //D2 line
//calculation
a_plus_b=2.54/N*10^-2; //grating element (in m)
theta_2v=asind((2*lambda_v)/a_plus_b); //second order spectra of violet light
theta_2r=asind((2*lambda_r)/a_plus_b); //second order spectra of red light
theta_3v=asind((2*lambda_v)/a_plus_b); //second order spectra of violet light
format(16)
d=a_plus_b;
dtheta=(n*6*10^-10)/(sqrt(1-(n*dlambda/d)^2)*a_plus_b); //angular separation of D1 and D2 lines (in radians)
```

```
if theta_3v<theta_2r then
    disp("second and third order spectra overlap")
else
    disp("second and third order spectra do not
       overlap")
end
printf("\nangular separation of D1 and D2 lines is
   %0.4f radians",dtheta)
```

**Scilab code Exa 3.12** calculation of angles at which first and second order maxima

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.12
//calculation of angles at which first and second
   order maxima are observed
//given data
lambda=632.8*10^-9; //wavelength (in m) of light
a_plus_b=10^-2/6000; //number of lines per m
//calculation
theta1=asind(lambda/a_plus_b); //angle at which
   first order maxima is observed
theta2=asind(2*lambda/a_plus_b); //angle at which
   second order maxima is observed
printf("\nangle at which first order maxima is
   observed is %0.2f degrees",theta1)
printf("\nangle at which second order maxima is
   observed is %0.2f degrees",theta2)
//result
//angle at which first order maxima is observed is
   22.31 degrees
//angle at which second order maxima is observed is
   49.41 degrees
//the answers vary due to round off error
```

**Scilab code Exa 3.13** calculation of separation of lines in first order spectrum

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.13
//calculation of separation of lines in first order
    spectrum
//given data
lambda1=5000*10^-10; //first wavelength (in m)
lambda2=3200*10^-10; //second wavelength (in m)
a_plus_b=10^-2/10000; //grating lines per m
f=150*10^-2; //focal length (in m)
//calculation
theta1=asind(lambda1/a_plus_b); //first diffraction
    angle (in degrees)
theta2=asind(lambda2/a_plus_b); //second diffraction
     angle (in degrees)
x1=f*tand(theta1); //position of first diffraction
    angle (in m)
x2=f*tand(theta2); //position of second diffraction
    angle (in m)
x=x1-x2; //separation of lines in first order
    spectrum
printf("\nseparation of lines in first order
    spectrum is %0.2f cm",x*10^2)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 3.14** calculation of wavelength of used

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 3.14
//calculation of wavelength of used
//given data
theta=%pi/6; //angle (in degrees) at which second
    order spectral line is observed
n=2; //order of spectral line
a_plus_b=10^-2/4250; //number of lines
//calculation
lambda=a_plus_b*sin(theta)/n; //wavelength of light
    (in m)
printf("\nThe wavelength of light is %2d Armstrong",
    lambda*10^10)
```

**Scilab code Exa 3.15** calculation of lines per cm

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.15
//calculation of lines per cm
//given data
theta=%pi/6; //angle of diffraction (in degrees)
lambda=6*10^-5; //wavelength of light used (in cm)
n=1; //order of line
//calculation
a_plus_b=sin(theta)/(n*lambda); //number of lines
    per cm
printf("\nThe number of lines per cm is %2d",
    a_plus_b)
```

**Scilab code Exa 3.16** calculation of number of lines on grating

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.16
//calculation of number of lines on grating
//given data
W=5; //width of diffraction grating (in cm)
lambda1=6000; //wavelength (in Armstrong) of one
    line
lambda2=4500; //wavelength (in Armstrong) of another
    line
theta=%pi/6; //angle of diffraction (in degrees)
//calculation
n=lambda2*10^-10/((lambda1-lambda2)*10^-10); //order
    of spectrum
a_plus_b=n*lambda1*10^-10/sin(theta); //number of
    lines per m
N=round(W*10^-2/a_plus_b); //number of lines
printf("\nNumber of lines on grating are %0.3d",N)
```

**Scilab code Exa 3.17** calculation of highest order spectrum and longest wavelength

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.17
//calculation of (i)highest order spectrum and (ii)
    longest wavelength of light
//given data
a_plus_b=10^-2/6000; //number of lines per m
theta=%pi/2; //maximum angle (in degrees)
lambda=4000; //wavelength of light (in Armstrong)
n=a_plus_b/(lambda*10^-10); //highest order of
    spectrum
n1=1; //minimum order of spectrum
```

```
lambda_max=round((a_plus_b)*sin(theta)*10^10/n1); //maximum wavelength (i Armstrong)
printf("\n(i)Highest order of spectrum is %d",n)
printf("\n(ii)Longest wavelength of light is %5.2d Armstrong", lambda_max)
```

**Scilab code Exa 3.18** calculation of number of lines per cm

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.18
//calculation of number of lines per cm
//given data
lambda1=5400*10^-10; //wavelength (in m) of green light
lambda2=4050*10^-10; //wavelength (in m) of violet light
theta=30; //angle of diffraction (in degrees)
//calculation
n1=lambda2/(lambda1-lambda2); //order of light
//the above equation was derived from the formula n1*lambda1=n2*lambda2 where n2=n1+1
a_plus_b=(n1*lambda1)/sind(theta)*10^2; //grating lines (in cm)
n=1/a_plus_b; //number of lines per cm
printf('\nnumber of lines per cm is %d',n)
```

**Scilab code Exa 3.19** calculation of angular positions of first two maxima and half

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 3.19
//calculation of (i) angular positions of first two maxima (ii) half-widths of maxima
//given data
a_plus_b=(0.5+1.9)*10^-2; //grating lines (in m)
lambda=0.6*10^-2; //wavelength (in m) of light
N=10; //number of slits
//calculation
theta1=asind(lambda/a_plus_b); //angular position of first maximum (in degrees)
dtheta1=tand(theta1)/N; //half width of first maximum (in degres)
theta2=asind(2*lambda/a_plus_b); //angular position of second maximum (in degrees)
dtheta2=tand(theta2)/(2*N); //half width of second maximum (in degrees)
printf("\n(i)angular position of first minimum is %0.2f degrees",theta1)
printf("\nangular position of second minimum is %d degrees",theta2)
printf("\n(ii)half width of first maximum is %0.2f degrees",dtheta1*180/%pi)
printf("\nhalf width of second maximum is %0.2f degrees",dtheta2*180/%pi)
```

---

**Scilab code Exa 3.20** calculation of orders of spectra and which wavelengths in the

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.20
//calculation of (a)orders of spectra that will be shown by a plane diffraction grating (b)which wavelengths in the range of 3500 Armstrong and 8000 Armstrong will overlap with the third order
```

```
//given data
lambda=5000*10^-10; //wavelength (in m)
a_plus_b=10^-2/6000; //grating lines per m
theta=90; //maximum angle
n=3; //third order in which wavelengths will overlap
n1=1; //first order
n2=2; //second order
n3=3; //third order
n4=4; //fourth order
n5=5; //fifth order
//calculation
n_max=(a_plus_b*sind(theta))/lambda; //maximum n
lambda1=n*lambda*10^10/n1; //wavelength (in
    Armstrong) for n=1
lambda2=n*lambda*10^10/n2; //wavelength (in
    Armstrong) for n=2
lambda3=n*lambda*10^10/n3; //wavelength (in
    Armstrong) for n=3
lambda4=n*lambda*10^10/n4; //wavelength (in
    Armstrong) for n=4
lambda5=n*lambda*10^10/n5; //wavelength (in
    Armstrong) for n=5
printf("\n(a)orders of spectra that will be shown by
     a plane diffraction grating is %d",n_max)
printf("\n(b)wavelength (in Armstrong) for n=1 is %d
     Armstrong",lambda1)
printf("\nwavelength (in Armstrong) for n=2 is %d
    Armstrong",lambda2)
printf("\nwavelength (in Armstrong) for n=2 is %d
    Armstrong",lambda3)
printf("\nwavelength (in Armstrong) for n=2 is %d
    Armstrong",lambda4)
printf("\nwavelength (in Armstrong) for n=2 is %d
    Armstrong",lambda5)
printf("\nthus, amongst the above wavelengths, 7500
    Armstrong and 3750 Armstrong fall in the visible
    region (3500 to 8000 Armstrong)")
```

**Scilab code Exa 3.21** calculation of distance between two stars

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.21
//calculation of distance between two stars
//given data
D=0.5; //diameter (in m) of objective of a telescope
lambda=5600*10^-10; //wavelength (in m) of light
d=5*365*24*60*60*3*10^8; //distance (in m) of two stars from earth
//calculation
theta=(1.22*lambda)/D; //resolving limit (in radians)
x=theta*d; //distance (in m) between two stars
printf("\ndistance between two stars is %1.2e m",x)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 3.22** calculation of resolving limit

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.22
//calculation of resolving limit
//given data
lambda=5500*10^-10; //wavelength (in m) of light
alpha=atand(0.5/0.25)
x_min=(0.61*lambda)/sind(alpha); //resolving limit
printf("\nresolving limit is %1.1e m",x_min)
//the answer provided in the book is wrong
```

**Scilab code Exa 3.24** to check if transmission grating with 1200 elements resolve p

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.24
//to check if transmission grating with 1200 elements resolve principal maxima in the first order
//given data
lambda1=589.0*10^-9; //first wavelength (in m) of sodium discharge lamp
lambda2=589.6*10^-9; //second wavelength (in m) of sodium discharge lamp
n=1
lambda=(lambda1+lambda2)/2; //average wavelength (in m)
d_lambda=lambda2-lambda1; //difference between wavelengths
N=lambda/(n*d_lambda); //minimum number of lines
printf("\nminimum number of lines needed to resolve given spectral lines is %3.1f",N)
if N<1200 then
    disp("spectral lines will be resolved in first order")
else
    disp("spectral lines will not be resolved in first order")
end
```

**Scilab code Exa 3.25** calculation of minimum number of lines per cm in a half inch

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.25
//calculation of minimum number of lines per cm in a
    half inch wide grating
//given data
lambda_D1=5896*10^-10; //wavelength (in m) of D1
   line
lambda_D2=5890*10^-10; //wavelength (in m) of D2
   line
W=1.27; //width of grating (in cm)
//calculation
lambda=(lambda_D1+lambda_D2)/2; //average wavelength
    (in m) of light
dlambda=lambda_D1-lambda_D2; //difference between
   the two wavelengths (in m)
N=lambda/dlambda; //number of lines
N_dash=N/W; //number of lines per cm in a half inch
   wide grating
printf("\nnumber of lines per cm in a half inch wide
    grating is %1.2f",N_dash)
//the answers vary due to round off error
```

**Scilab code Exa 3.26** calculation of number of lines per cm

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.26
//calculation of number of lines per cm
//given data
W=2.5; //width of grating (in cm)
lambda1=5890; //wavelength (in Armstrong) of line D1
lambda2=5896; //wvelength (in Armstrong) of line D2
```

```
n=2; //second order
//calculation
lambda=(lambda1+lambda2)/2; //average wavelength (in m)
dlambda=lambda2-lambda1; //difference of wavelengths (in m)
N=lambda/(n*dlambda); //number of lines
N_dash=N/W; //number of lines per cm
printf("\nnumber of lines per cm is %3.1f",N_dash)
//the answers vary due to round off error
```

**Scilab code Exa 3.27** calculation of minimum number of lines required for grating a

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.27
//calculation of (a) minimum number of lines required for grating (b) slit spacing for a grating
//given data
lambda1=531.62; //wavelength (nm) of light
lambda2=531.81; //wavelength (nm) of light
W=1.32; //number of lines (in cm)
//calculation
lambda=(lambda1+lambda2)/2; //average wavelength (in nm)
dlambda=lambda2-lambda1; //difference of wavelengths (in m)
n=1; //first order
N=round(lambda/dlambda); //number of lines
a_plus_b=W*10^-2/N; //slit spacing for a grating (in m)
printf("\n(a)minimum number of lines required for grating is %d",N)
```

```
printf("\n(b)slit spacing for a grating is %0.2f
    micrometer",a_plus_b*10^6)
```

**Scilab code Exa 3.28** calculation of resolving power of grating in the first three

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.28
//calculation of (a)resolving power of grating in
    the first three orders (b)wavelength separation
//given data
W=4; //width per cm
N_dash=3000; //number of lines/cm
n=3; //third order
lambda=400*10^-9; //wavelength (in m) of light
//calculation
N=W*N_dash; //number of lines
RP_I=1*N; //resolving powder of first order
RP_II=2*N; //resolving power of second order
RP_III=3*N; //resolving power of third order
dlambda=lambda/(n*N); //wavelength separation (in m)
//result
printf("\n(a)resolving power of first order is %d",
    RP_I)
printf("\nresolving power of second order is %d",
    RP_II)
printf("\nresolving power of third order is %d",
    RP_III)
printf("\n(b)wavelength separation is %0.3f
    Armstrong",dlambda*10^10)
//The answers vry due to round off error
```

**Scilab code Exa 3.29** calculation of wavelength of lines and minimum grating width

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.29
//calculation of wavelength of lines and minimum
    grating width required to resolve
//given data
dlambda=0.5*10^-10; //difference between wavelength
    of lines (in m)
theta=10; //angle (in degrees) at which spectral
    line appears
dtheta=3/3600*%pi/180; //angle at which it appears
    higher by 3 second (in radians)
//calculation
lambda=tand(theta)*dlambda/dtheta; //wavelength (in
    m) of lines
lambda1=lambda+dlambda; //wavelength (in m) of light
    higher than the other wavelength
W=lambda^2/(dlambda*sind(theta)); //resolving power
//result
printf("\nwavelength of one line is %0.1f Armstrong"
    ,lambda*10^10)
printf("\nwavelength of another line is %0.1f
    Armstrong",lambda1*10^10)
printf("\nresolving power is %0.3f",W)
//the first part of the answer varies due to round
    off error
//second part of the answer provided in the textbook
    is wrong
```

**Scilab code Exa 3.30** calculation of angle at which first order maximum occurs AND

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 3.30
//calculation of (a)angle at which first order
   maximum occurs for wavelength 5890 Armstrong (b)
   angular separation of these lines in first order
   (c)maximum wavelength separation resolvable by
   grating in its first order (d)minimum number of
   lines on the grating needed to resolve doublet in
    first order
function[mint, secnd]=degmin(theta)
    mint=(theta-floor(theta))*60;
    secnd=(mint-floor(mint))*60;
    endfunction
//given data
n=1; //order of maximum
lambda1=5890*10^-10; //wavelength (in m) of one line
    of light
lambda2=5895.59*10^-10; //wavelength (in m) of
   another line of light
W=2.50*10^-2; //width of grating (in m)
N=1.20*10^4; //number of rulings
n=1; //order of maximum
//calculation
a_plus_b=W/N; //slit spacing for grating (in m)
theta=asind((n*lambda1)/a_plus_b); //angle (in
   degrees) at which first order maximum occurs for
   wavelength 5890 Armstrong
dlambda=lambda2-lambda1; //difference between
   wavelengths (in m)
dtheta=dlambda*n/(a_plus_b*cosd(theta)); //angular
   separation of these lines in first order
[mint, secnd]=degmin(dtheta); //call deg_2_degmin
   function
RP=N*n; //maximum wavelength separation resolvable
   by grating in its first order
N=(lambda2/dlambda)/n; //minimum number of lines on
   the grating needed to resolve doublet in first
```

```
order
printf("\n(a)angle at which first order maximum occurs for wavelength 5890 Armstrong is %0.1f degrees",theta)
printf("\n(b)angular separation of these lines in first order is %0.4f minutes %0.2f seconds",mint,secnd)
printf("\n(c)maximum wavelength separation resolvable by grating in its first order is %d",RP)
printf("\n(d)minimum number of lines on the grating needed to resolve doublet in first order is %d",N)
//the answers vary due to round off error
//the value of lambda2 is taken different in the question and the calculation
```

**Scilab code Exa 3.32** calculation of dispersion AND resolving power AND smallest di

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.32
//calculation of (i)dispersion (ii)resolving power (iii)smallest difference in wavelength resolved and when half the ruling is covered
//given data
a_plus_b=10^-2/5000; //number of lines per m
lambda=5000*10^-10; //wavelength (in m)
dlambda=0.1*10^-10; //difference of wavelengths (in m)
n=2; //order of diffraction
N=5000*5; //number of lines
//calculation
dtheta=(n*dlambda)/sqrt(a_plus_b^2-n^2*lambda^2); //
```

```
dispersion (in radian)
RP=n*N; //resolving power
d_lambda=lambda/(n*N); //smallest difference in wavelength resolved (in m)
//when half the ruling is covered
N_dash=50000/2; //number of lines
RP_dash=RP/2; //resolving power
d_lambda_dash=lambda/(n*N_dash); //difference in wavelength (in m)
dtheta_dash=2*2.38; //dispersion (in minutes) since the value of dispersion is 2.38 in minutes
printf("\n(i)dispersion is %1.2e rad",dtheta)
printf("\n(ii)resolving power is %d",RP)
printf("\n(iii)smallest difference in wavelength resolved is %1.1f Armstrong",d_lambda*10^10)
printf("\n(i)resolving power when half the ruling is covered is %d",RP_dash)
printf("\n(ii)difference in wavelength is when half the ruling is covered is %1.1f Armstrong",d_lambda_dash*10^10)
printf("\n(iii)dispersion when half the ruling is covered is %1.2f minutes",dtheta_dash)
//the answer provided for difference in wavelength is when half the ruling is covered is wrong
```

**Scilab code Exa 3.33** calculation of separation of wavelengths in first order and r

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.33
//calculation of separation of wavelengths in first order and resolving power in second order
//given data
N=1000; //number of lines a grating has
```

```
lambda=6000; //wavelength (in Armstrong)
n=1; //for first order
n_dash=2; //for second order
//calculation
d_lambda=lambda/N; //separation between two wavelengths (in Armstrong) for first order
RP=n_dash*N; //resolving power for second order
printf("\nseparation between two wavelengths for first order is %d Armstrong",d_lambda)
printf("\nresolving power for second order is %d",RP)
```

**Scilab code Exa 3.34** calculation of diffraction angle

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.34
//calculation of diffraction angle
//given data
lambda1=6000*10^-10; //wavelenght (in m) of one line of light
lambda2=6000.5*10^-10; //wavelength (in m) of another line of light
W=10*10^-3; //width of grating (in m)
//calculation
dlambda=lambda2-lambda1; //difference between the two wavelengths (in m)
theta=asind((lambda1^2)/(W*dlambda)); //diffraction angle (in degrees)
printf('diffraction angle is %d degrees',theta)
```

**Scilab code Exa 3.35** calculation of minimum wavelength separation

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.35
//calculation of minimum wavelength separation
//given data
lambda1=5890; //wavelength (in Armstrong) of light
lambda2=5896; //another wavelength (in Armstrong) of
    light
W=3; //width of grating (in cm)
W_dash=0.3; //new width of grating (in cm)
//calculation
a_plus_b=1; //assuming for simplicity of calculation
N_dash=(W_dash*a_plus_b)/(W*a_plus_b); //number of
   lines per cm
dlambda=lambda2-lambda1; //separation of wavelengths
    (in m)
lambda_by_n=dlambda; //formula of separation of
   wavelengths
dlambda_dash=lambda_by_n/N_dash; //minimum
   wavelength separation (in Armstrong)
printf("\nminimum wavelength separation is %d
   Armstrong",dlambda_dash)
```

**Scilab code Exa 3.36** calculation of wavelength separation

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.36
//calculation of wavelength separation
//given data
del_lambda=6; //wavelength (in Armstrong) of light
W=1; //assuming width to be 1 for simplicity of
   calculation
```

```
//calculation
lambdaa_plus_b_by_n=del_lambda; //formula for
    separation of wavelength
W_dash=10*W; //new width
dlambda_dash=lambdaa_plus_b_by_n/W_dash; //
    wavelength separation (in Armstrong)
printf("\nwavelength separation is %0.1f Armstrong",
    dlambda_dash)
```

**Scilab code Exa 3.37** calculation of angular dispersion and effective width of grat

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.37
//calculation of angular dispersion and effective
    width of grating
//given data
lambda=5000*10^-10; //wavelength (in m) of light
n=2; //order of diffraction
dlambda=10^-10; //difference of wavelengths (in m)
a_plus_b=10^-2/6000; //number of lines per m
//calculation
dtheta=(n*dlambda)/sqrt(a_plus_b^2-n^2*lambda^2); //
    angular dispersion (in radians)
N=lambda/(n*dlambda); //resolving power
W=N*a_plus_b; //width of grating
printf("\nangular dispersion is %1.1e rad",dtheta)
printf("\nwidth of grating is %0.3f cm",W*10^2)
```

**Scilab code Exa 3.38** calculation of resolving power

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 3.38
//calculation of resolving power
//given data
lambda=5000*10^-10; //wavelength (in m) of light
n=2; //order of diffraction
W=2.5*10^-2; //width of grating (in m)
//calculation
a_plus_b=lambda/0.1; //grating element (in m)
N=W/a_plus_b; //number of lines on grating
RP=n*N; //resolving power
printf("\nResolving power is %d",RP)
```

**Scilab code Exa 3.39** calculation of dispersion around 5460 Armstrong in third orde

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.39
//calculation of (a)dispersion around 5460 Armstrong in third order (b)resolving power in its fifth order
//given data
N=9600; //number of lines
W=3*10^-2; //width of grating (in m)
n=3; //third order
n_dash=5; //fifth order
lambda=5460*10^-10; //wavelength (in m) of light
//calculation
a_plus_b=W/N; //grating element (in m)
theta=asind(n*lambda)/a_plus_b; //from the gratig equation (a+b)*sin(theta)=n*lambda (in degrees)
dtheta_by_dlambda=n/((a_plus_b)*cosd(theta)); //dispersion (in rad/nm)
```

```
RP=n_dash*N; //resolving power
printf("\n(a)dispersion around 5460 Armstrong is %1
   .4f rad/nm",dtheta_by_dlambda/10^9)
printf("\n(b)resolving power in its fifth order is
   %d",RP)
//the first part of the answer provided in the
   textbook is wrong
```

**Scilab code Exa 3.40** calculation of minimum number of lines of grating

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.40
//calculation of minimum number of lines of grating
//given data
lambda1=589.593*10^-9; //first wavelength of sodium
   D lines (in m)
lambda2=588.996*10^-9; //second wavelength of sodium
    D lines (in m)
n=1; //order of grating
//calculation
lambda=(lambda1+lambda2)/2; //average wavelength of
   both thewavelengths of sodium D lines (in m)
d_lambda=lambda1-lambda2; //difference of both the
   wavelengths (in m)
N=lambda/(n*d_lambda); //minimum number of lines of
   grating
printf("\nminimum number of lines of grating is %d",
   N)
//the answers vary due to round off error
```

**Scilab code Exa 3.41** to check if spectral lines will be resolved

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.41
//to check if spectral lines will be resolved
//given data
//in first order
lambda1=5160.34; //first wavelength of spectral line
    a plane diffraction just resolves (in Armstrong)
lambda2=5160.85; //second wavelength of spectral
   line a plane diffraction just resolves (in
   Armstrong)
//calculation
lambda=(lambda1+lambda2)/2; //average wavelength (in
    Armstrong)
d_lambda=lambda2-lambda1; //difference of
   wavelengths (in Armstrong)
N=lambda/d_lambda; //number of lines
//in second order
lambda_1=8037.20; //first wavelength (in Armstrong)
lambda_2=8037.50; //second wavelength (in Armstrong)
lambda_dash=(lambda_1+lambda_2)/2; //average
   wavelength (in Armstrong)
n=2; //order of grating
dlambda=lambda_2-lambda_1; //difference of
   wavelengths (in Armstrong)
N_dash=lambda_dash/(n*dlambda); //number of lines
printf("\nfor first order number of lines is %d",N)
printf("\nfor first order number of lines is %d",
   N_dash)
if N>N_dash then
    disp("the given lines are resolved")
else
    disp("the given lines remain unresolved")
end
```

**Scilab code Exa 3.42** to check if spectral lines will be resolved

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.42
//to check if spectral lines will be resolved
//given data
//in first order
lambda1=5140.34; //first wavelength of spectral line
    a plane diffraction just resolves (in Armstrong)
lambda2=5140.85; //second wavelength of spectral
   line a plane diffraction just resolves (in
   Armstrong)
//calculation
lambda=(lambda1+lambda2)/2; //average wavelength (in
    Armstrong)
d_lambda=lambda2-lambda1; //difference of
   wavelengths (in Armstrong)
N=lambda/d_lambda; //number of lines
//in second order
lambda_1=8037.2; //first wavelength (in Armstrong)
lambda_2=8037.50; //second wavelength (in Armstrong)
lambda_dash=(lambda_1+lambda_2)/2; //average
   wavelength (in Armstrong)
n=2; //order of grating
dlambda=lambda_2-lambda_1; //difference of
   wavelengths (in Armstrong)
N_dash=lambda_dash/(n*dlambda); //number of lines
printf("\nfor first order number of lines is %d",N)
printf("\nfor first order number of lines is %d",
   N_dash)
if N>N_dash then
    disp("the given lines are resolved")
```

```
else
    disp("the given lines remain unresolved")
end
```

**Scilab code Exa 3.43** calculation of fringe spacing

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 3.43
//calculation of fringe spacing
//given data
n=1; //order of grating
lambda=632.8*10^-9; //wavelength (in m) of laser
theta=0.1/1
//calculation
d=lambda/theta; //fringe spacing (in m)
printf("\nfringe spacing is %1.3f micrometer",d
   *10^6)
```

# Chapter 4

# Coherence Lasers and Optical Fibres

**Scilab code Exa 4.1** calculation of coherent length and coherent time

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.1
//calculation of coherent length and coherent time
//given data
lambda=4800*10^-10; //wavelength of light (in m)
n=25; //number of waves
c=3*10^8; //speed of light (in m/s)
//calculation
l_c=n*lambda; //coherent length (in metre)
tau_c=l_c/c; //coherent time (in s)
printf("\ncoherent length is %d micrometer",l_c
   *10^6)
printf("\ncoherent time is %1.0e s",tau_c)
```

**Scilab code Exa 4.2** calcuation of coherent time coherent length and Q value for so

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.2
//calcuation of (a)coherent time, coherent length and Q value for sodium D1 line(b)coherent time, coherent length and Q value for sodium light of D1 and D2 line
//(a)for sodium D1 line
//given data
lambda=5890*10^-10; //wavelength (in m) of line D1
w=0.1*10^-10; //spectral width (in m)
c=3*10^8; //speed (in m/s) of light
//calculation
Q=round(lambda/w); //Q value
l_c=Q*lambda; //coherent length (in m)
tau_c=l_c/c; //coherent time (in sec)
//(b)for sodium D1 and D2 lines
//given data
lambda1=5890*10^-10; //wavelength (in m) of line D1
lambda2=5896*10^-10; //wavelength (in m) of line D2
//calculation
w_dash=lambda2-lambda1; //spectral width (in m)
Q_dash=((lambda1+lambda2)/2)/w_dash; //Q value
format(16)
lambda_dash=(lambda1+lambda2)/2; //average wavelength of D1 and D2 lines
lc=lambda_dash*Q_dash; //coherent length (in m)
tauc=lc/c; //coherent time (in s)
printf("\n(a)for sodium D1 line")
printf("\nQ value is %d",Q)
printf("\ncoherent length is %1.2f cm",l_c*10^2)
printf("\ncoherent time is %1.2e s",tau_c)
printf("\n(b)for sodium D1 and D2 lines")
printf("\nQ value is %3.2f",Q_dash)
printf("\ncoherent length is %1.2f micrometre",lc
```

```
    *10^6)
printf("\ncoherent time is %1.2e s",tauc)
//the answer for coherent time in (a) varies due to
    round off error
//the answers provided in the textbook for coherent
    length and coherent time in (b) are wrong
```

**Scilab code Exa 4.3** calculation of coherent length and quality factor

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.3
//calculation of (i)coherent length (ii)quality
    factor
//given data
lambda=6000*10^-10; //wavelength (in m)
del_nu=10^-2; //spectral width (in Hz)
c=3*10^8; //speed of light (in m/s)
//calculation
nu=c/lambda; //frequency (in Hz)
Q=nu/del_nu; //quality factor
l_c=Q*lambda; //coherent length (in m)
printf("\nquality factor is %1.0e",Q)
printf("\ncoherent length is %d km",l_c*10^-3)
//the answer provided in the txtbook is wrong
```

**Scilab code Exa 4.4** calculation of coherent length and coherent time and Q value

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.4
```

```
//calculation of coherent length, coherent time and
   Q value
//given data
lambda1=0.4*10^-6; //first wavelength (in m)
lambda2=0.8*10^-6; //second wavelength (in m)
c=3*10^8; //speed of light (in m/s)
//calculation
lambda=(lambda1+lambda2)/2; //mean wavelength (in m)
del_lambda=lambda2-lambda1; //difference between
    wavelengths(in m)
Q=lambda/del_lambda; //Q value
l_c=Q*lambda; //coherent length (in m)
tau_c=l_c/c; //coherent time (in sec)
printf("\nQ value is %1.1f",Q)
printf("\ncoherent length is %1.1f micrometer",l_c
    *10^6)
printf("\ncoherent time is %1.0e s",tau_c)
//answer provided in the textbook is wrong
```

**Scilab code Exa 4.5** calculation of spectral purity of line

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.5
//calculation of spectral purity of line
//given data
lambda=643.8*10^-9; //wavelength (in m)
tau_c=10^-9; //coherent time (in sec)
c=3*10^8; //speed of light (in m/s)
//calculation
del_lambda=lambda^2/(c*tau_c); //spectral spread (in
    nm)
sp=lambda/del_lambda; //spectral purity of the line
printf("\nThe spectral purity of the line is %d",sp)
```

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

**Scilab code Exa 4.6** calculation of separation of two slits

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.6
//calculation of separation of two slits
//given data
lambda=5000*10^-10; //wavelength (in m) f light
theta=(32*%pi)/(180*60); //angle (in minutes)
    subtended by the sun on the earth
//calculation
l_w=lambda/theta; //separation of two slits (in m)
printf("\nseparation of two slits is %1.0e m",l_w)
```

**Scilab code Exa 4.7** calculation of units of light received by minima

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.7
//calculation of units of light received by minima
//given data
visibility=0.29
I_max=20; //maximum intensity of resulting
    interference pattern (in units)
//calculation
I_min=(I_max-0.29*I_max)/1.29; //units of light
    received by minima obtained from the formula,
    visibility=(I_max-I_min)/(I_max+I_min)
```

```
printf("\nlight received by minima is %d units",
   I_min)
```

**Scilab code Exa 4.8** calculation of coherent length of wavetrain and to check wheth

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.8
//(i)calculation of coherent length of wavetrain (ii
   ) to check whether interference is observed by
   human eye
//given data
tau_c=10^-8; //coherent time (in s)
c=3*10^8; //speed of light (in m/s)
x1=5; //distance travelled by one beam (in m)
x2=10; //distance travelled by other beam (in m)
//calculation
l_c=c*tau_c; //coheret length (in m)
del_x=x2-x1; //path difference between two beams (in
    m)
printf("\n(i)coherent length of wavetrain is %d m",
   l_c)
printf("\n(ii)path difference between two beams is
   %d m",del_x)
if del_x<l_c then
    disp("interference is observable by human eye")
else
    disp("interference is not observable by human
       eye")
end
```

**Scilab code Exa 4.9** calculation of spectral purity factor and coherence length and

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.9
//calculation of spectral purity factor, coherence
   length and coherence time
//given data
lambda=643.8; //wavelength of light (in nm)
del_lambda=0.001; //sectral spread (in nm)
c=3*10^8; //speed of light (in m/s)
//calculation
Q=round(lambda/del_lambda); //spectral purity factor
l_c=Q*lambda*10^-9; //coherence length (in m)
tau_c=l_c/c; //coherence time (in s)
printf("\nSpectral purity factor is %d",Q)
printf("\nCoherence length is %0.1f cm",l_c*10^2)
printf("\nCoherence time is %4.2e s",tau_c)
```

**Scilab code Exa 4.10** calculation of size of pinhole

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.10
//calculation of size of pin-hole
//given data
F=60; //focal length (in cm)
l=10; //length of aperture (in cm)
lambda=633*10^-4; //wavelength (in cm) of light
//calculation
//for zero coherence at the periphery over lens
   aperture
a=1.22*lambda*F/l; //size of hole (in cm)
//for degree of coherence not below 80% over the
   aperture
```

```
a_dash=0.36*F/l*lambda; //size of hole (in cm)
printf("\nsize of the hole for zero coherence at the periphery over lens aperture is %0.1f micrometer",a*10^4)
printf("\nsize of the hole for degree of coherence not below 80 percent over the aperture is %0.2f micrometer",a_dash*10^4)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 4.11** calculation of efficiency of laser

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.11
//calculation of efficiency of laser
//given data
C=1000*10^-6; //capacitance (in Farad)
V=4000; //voltage (in volts)
E=10; //energy (in Joule)
//calculation
opo=E; //optical power output
ip=1/2*C*V^2; //input power
nu=opo/ip; //efficiency of laser
printf ("\nefficiency of laser is %0.3f percent",nu*100)
```

**Scilab code Exa 4.12** calculation of ratio of stimulated emission to spontaneous em

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.12
```

```
//calculation of ratio of stimulated emission to spontaneous emission
//given data
lambda=632.8*10^-9; //wavelength (in m) of He-Ne laser
c=3*10^8; //speed (in m/s) of light
h=6.63*10^-34; //Planck's constant (in Js)
r=(1*10^-3)/2; //beam radius (in m) of laser
d_nu=1.5*10^8; //line width of laser line (in Hz)
E=99*10^-3; //energy (in W) within resonator
//calculation
nu=c/lambda; //frequency (in Hz) of laser
B21_by_A21=(c^3)/(8*%pi*h*nu^3); //ratio of Einstein's coefficients (in m^3/J.s)
I=E/(%pi*r^2); //intensity
rho_v=I/(c*d_nu); //density (in J.s/m^3)
rho_vB21_by_A21=(B21_by_A21)*rho_v; //ratio of stimulated emission to spontaneous emission
printf("\nratio of stimulated emission to spontaneous emission is %2.1f",rho_vB21_by_A21)
```

**Scilab code Exa 4.13** calculation of angular speed and areal spread

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.13
//calculation of (i)angular speed and (ii)areal spread
//given data
lambda=7000*10^-10; //wavelength (in m) of light
a=5*10^-3; //aperture (in m)
D=4*10^8; //distance from the earth (in m)
//calculation
theta=(1.22*lambda)/a; //angle (in radian) of
```

```
diffraction
A=%pi*(D*theta)^2; //areal spread (in m^2)
printf('\n(i)angular spread is %1.1e rad',theta)
printf('\n(ii)areal spead is %1.2e m^2',A)
//(ii)the answers vary due to round off error
```

**Scilab code Exa 4.14** calculation of area and intensity of image

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.14
//calculation of area and intensity of image
//given data
lambda=720*10^-9; //wavelength (in m) of light
d=5*10^-3; //aperture (in m)
f=0.1; //focal length (in m)
P=50*10^-3; //power (in Watt)of laser beam
//calculation
d_theta=(1.22*lambda)/d; //angular spread
D=f
A=%pi*(D*d_theta)^2 //area of image (in m^2)
I=P/A; //intensity of image (in W/m^2)
printf("\narea of image is %1.2e m^2",A)
printf("\nintensity of image is %1.2e W/m^2",I)
//the answers provided in the textbook are wrong
```

**Scilab code Exa 4.15** calculation of coherent length and band width and line width

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.15
```

```
//calculation of coherent length,band width and line width of laser
//given data
lambda=650.0*10^-9; //wavelength (in m) of light
del_tau=0.5*10^-9; //time of pulses (in seconds)
c=3*10^8; //speed (in m/s) of light
//calculation
l_c=c*del_tau; //coherent length (in m)
del_nu=1/del_tau; //band width (in Hz)
del_lambda=lambda^2/c*del_nu; //line width (in m)
printf("\ncoherent length is %0.2f m",l_c)
printf("\nband width is %1.0e Hz",del_nu)
printf("\nline width is %0.3f Armstrong",del_lambda*10^10)
```

**Scilab code Exa 4.16** to show if diffraction contribute seriously to energy loss at

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.16
//to show if diffraction contribute seriously to energy loss at lassing wavelength 694.3 nm
//given data
lambda=694.3*10^-9; //wavelength (in m)
a=0.1; //in m
L=0.1; //length (in m)
//calculation
theta=a^2/lambda; //diffraction angle
if theta>L then
    disp("diffraction is not an important energy loss mechanism")
else
    disp("diffraction is an important energy loss mechanism")
```

```
end
```

**Scilab code Exa 4.17** calculation of minimum number of ions present

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.17
//calculation of minimum number of ions present
//given data
lambda=720; //wavelength (in nm)
E=0.1; //energy (in J)
h=6.62*10^-34; //planck's constant (in m sq kg/s)
c=3*10^8; //speed of light (in m/s)
//calculation
nu=c/(lambda*10^-9); //frequency (in Hz)
n=E/(h*nu); //number of ions present
printf("Number of ions present is %1.3e",n)
```

**Scilab code Exa 4.18** calculation of number of passes radiation has to make before

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.18
//calculation of number of passes radiation has to
   make before the threshold
//given data
E=0.1; //energy (in Joule)
beta_th=0.15; //threshold gain (per m)
l=0.2; //length (in m)
//calculation
```

```
m=12/(beta_th*l); //number of passes radiation has
    to make before the threshold
printf("\nnumber of passes radiation has to make
    before the threshold is %d",m)
```

**Scilab code Exa 4.19** calculation of number of longitudinal modes for gas lasers A

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.19
//calculation of number of longitudinal modes for
    gas lasers A and B
//given data
lambda_A=700*10^-9; //wavelength (in m) of red
    region
lambda_B=400*10^-9; //wavelength (in m) of blue
    region
L_A=0.35; //minor separation (in m) of red region
L_B=0.40; //minor separation (in m) of blue region
n0=1; //refractive index of laser medium
//calculation
qA=round(2*n0*L_A/lambda_A); //number of
    longitudinal models for gas laser A
qB=round(2*n0*L_B/lambda_B); //number of
    longitudinal models for gas laser B
printf("\nnumber of longitudinal models for gas
    laser A is %1.0e",qA)
printf("\nnumber of longitudinal models for gas
    laser B is %1.0e",qB)
//answer for qB provided in the textbook is wrong
```

**Scilab code Exa 4.20** calculation of number of modes operating in the cavity region

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.20
//calculation of number of modes operating in the
   cavity region
//given data
l=0.5; //separation of laser (in m)
Dg=1.5; //Doppler's gain band width (in Ghz)
c=3*10^8; //speed (in m/s) of light
n0=1; //refractive index of laser
//calculation
del_nu=c*10^-9/(2*l); //frequency (in GHz)
m=Dg/del_nu; //number of modes
disp(m,'number of modes operating in the cavity
   region is')
//the answer provided in the textbook is wrong
```

**Scilab code Exa 4.21** calculation of wavelength of laser emitted out

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.21
//calculation of wavelength of laser emitted out
//given data
h=6.626*10^-34; //planck's constant (in m sq kg/s)
c=3*10^8; //speed of light (in m/s)
E=1.85; //energy (in eV)
//calculation
lambda=(h*c)/(E*1.602*10^-19); //wavelength (in m)
printf("\nWavelength of laser emitted out is %d
   Armstrong",lambda*10^10)
```

**Scilab code Exa 4.22** calculation of minimum population for He Ne laser

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.22
//calculation of minimum population for He−Ne laser
//given data
lambda0=6328*10^-10; //wavelength (in m)
tau_m=10^-7; //transition time of metastable state (in s)
n0=1; //refractive index of laser
l=20*10^-2; //length (in m)
R1=0.98
del_nu=10^9; //frequency (in Hz)
c=3*10^8; //speed (in m/s) of light
alpha=0; //neglecting losses
//calculation
R2=R1
tau_c=(2*l*n0)/(c*(2*alpha*l-log(R1*R2))); //characteristic time of resonator (in s)
N2_minus_N1=(4*n0^3*tau_m*del_nu)/(c*lambda0^2*tau_c); //threshold population (per m^3)
printf("\nminimum population for He−Ne laser is %1.2e per cm^3",N2_minus_N1*10^2)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 4.23** to calculate threshold population inversion density and thres

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 4.23
//to calculate threshold population inversion density and threshold pump power
//given data
lambda0=632.8*10^-9; //in m
del_nu=10*9; //in Hz
tau_m=10^-7; //in s
l=10*10^-2; //in m
R1=0.98
R2=0.98
n0=1
c=3*10^8; //speed (in m/s) of light
nu=5*10^15
h=6.6*10^-34; //Planck's constant
//calculation
tau=-(2*n0*l)/(c*log(R1*R2)); //in s
N2_minus_N1=(4*n0^3*tau_m*del_nu)/(c*lambda0^2*tau); //threshold popularity inversion density (in m^3)
P_th=((N2_minus_N1)*h*nu)/tau_m; //threshold power (in W/m^3)
printf("\nthreshold popularity inversion density is %1.1e m^3",N2_minus_N1)
disp(P_th,'threshold power (in W/m^3) is')
//the answers provided in the textbook is wrong
```

**Scilab code Exa 4.24** comparison between HeNe laser and dye laser

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.24
//comparison between He-Ne laser and dye-laser
//given data
//for He-Ne laser
```

```
L=0.5; //mirror separation (in m)
lambda=632.8; //wavelength (in nm)
del_nu=1.5*10^3; //frequency (in Hz)
n=1
c=3*10^8; //speed (in m/s) of light
//for dye-laser
L_dash=2.0; //mirror separation (in m)
lambda1=570*10^-9; //wavelength (in m)
lambda2=640*10^-9; //wavelength (in m)
n_dash=1.4
//calculation
del_tsp_He_Ne=(2*n*L)/c; //separation between pulses (in s) in He-Ne
del_tsp_dye=(2*n_dash*L_dash)/c; //separation between pulses (in s) in dye
del_tp_He_Ne=1/del_nu; //pulse width of He_Ne (in s)
del_lambda=lambda2-lambda1; //wavelength range (in nm)
lambda_dash=(lambda1+lambda2)/2; //wavelength (in nm)
del_tp_dye=1/((c/lambda_dash^2)*del_lambda); //pulse-width of dye (in s)
printf("\nseparation between pulses in He-Ne is %1.2e s",del_tsp_He_Ne)
printf("\nseparation between pulses in dye is %1.2e s",del_tsp_dye)
printf("\npulse width of He_Ne is %1.3f ns",del_tp_He_Ne*10^9)
printf("\npulse-width of dye is %2.1f fs",del_tp_dye*10^15)
if del_tsp_He_Ne >  del_tsp_dye then
    disp("He-Ne lasers are more suited for mode-locking")
else
    disp("Dye lasers are more suited for mode-locking")
end
if del_tp_He_Ne <  del_tp_dye then
```

```
    disp("He-Ne lasers are more suited for mode-locking")
else
    disp("Dye lasers are more suited for mode-locking")
end
//the answer of pulse width of He_Ne laser is wrong because the value of frequency is given different in the question and the answer in the textbook
//the answer of pulse width of dye varies due to round off error
```

**Scilab code Exa 4.25** calculation of numerical aperture and maximum acceptance angl

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.25
//calculation of numerical aperture and maximum acceptance angle
//given data
n1=1.45; //refractive index of core
delta=0.01; //relative refractive index of core-cladding
//calculation
NA=n1*sqrt(2*delta); //numerical aperture
i_m=asind(NA); //maximum acceptance angle
printf("\nnumerical aperture is %0.3f",NA)
printf("\nmaximum acceptance angle is %0.2f degrees",i_m)
```

**Scilab code Exa 4.27** calculation of numerical aperture and maximum acceptance angl

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.27
//calculation of numerical aperture and maximum
   acceptance angle
//given data
n1=1.5; //refractive index of core
delta=0.01; //relative refractive index of core-
   cladding
//calculation
NA=n1*sqrt(2*delta); //numerical aperture
i_m=asind(NA); //maximum acceptance angle
printf("\nnumerical aperture is %0.2f",NA)
printf("\nmaximum acceptance angle is %0.2f degrees"
   ,i_m)
```

**Scilab code Exa 4.28** calculation of refractive indices of core and cladding

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.28
//calculation of refractive indices of core and
   cladding
//given data
NA=0.22; //numerical aperture
delta=0.012 //relative refractive index difference
//calculation
n1=NA/sqrt(2*delta); //refractive index of core
n2=n1-delta*n1; //refractive index of cladding
printf("\nrefractive index of core is %0.2f",n1)
printf("\nrefractive index of cladding is %0.2f",n2)
```

**Scilab code Exa 4.29** calculation of numerical aperture and angle of acceptance

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.29
//calculation of numerical aperture and angle of
    acceptance
//given data
n1=1.50; //refractive index of the core
n2=1.47; //refractive index of the cladding
//calculation
NA=(sqrt(n1^2- n2^2)); //numerical aperture
i_m=asind(NA); //acceptance angle (in degrees)
printf("\nnumerical aperture is %0.2f",NA)
printf("\nacceptance angle is %0.2f degrees",i_m)
//the answers vary due to round off error
```

---

**Scilab code Exa 4.30** to compare the maximum angle of acceptance and light gatherin

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 4.30
//to compare the maximum angle of acceptance and
    light gathering power of two fibres
//given data
n1=1.6; //core indice
n2=1.5; //cladding indice
n1_dash=2.1; //core indice
n2_dash=1.5; //cladding indice
//calculation
```

```
NA=sqrt(n1^2-n2^2); //light gathering power
i_m=asind(NA); //maximum angle of acceptance
NA_dash=sqrt(n1_dash^2-n2_dash^2); //light gathering
    power
printf("\n(a)light gathering power is %1.3f",NA)
printf("\n(a)maximum angle of acceptance is %2.2f
   degree",i_m)
printf("\n(b)light gathering power is %1.2f",NA_dash
   )
//(b)there is no limit to maximum angle of
   acceptance as its sin inverse does not exist
```

# Chapter 5

# Quantum Mechanics

**Scilab code Exa 5.1** calculation of maximum kinetic energy of photoelectrons and th

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.1
//calculation of maximum kinetic energy of
    photoelectrons and the stopping potential
//given data
phi=2; //work function (in eV)
h=6.63*10^-34; //Planck's constant (in Js)
e=1.6*10^-19; //charge of an electron (in C)
//from the equation E=(100V/m)[(sin 5*10^13)*t+sin
    (8*10^15)*t]
omega=8*10^15; //angular frequency (in rad per s)
//calculation
nu_0=(phi*e)/h; //threshold frequency (in Hz)
nu=omega/(2*%pi); //larger frequency (in Hz)
E=(h*nu)/e; //energy of photon (in eV)
Kmax=E-phi; //maximum kinetic energy (in eV)
V=(Kmax*e)/e; //stopping potential (in V)
printf("\nmaximum kinetic energy is %0.2f eV",Kmax)
printf("\nstopping potential is %0.2f V",V)
```

```
//the answers vary due to round off error
```

**Scilab code Exa 5.2** `calculation of velocity of electrons`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.2
//calculation of velocity of electrons
//given data
d=0.9086*10^-10; //first Bragg maximum (in m)
theta=65; //glazing angle (in degrees)
h=6.6*10^-34; //Planck's constant (in Js)
m_e=9.1*10^-31; //mass of electron (in kg)
//calculation
v=h/(2*d*sin(theta)*m_e); //velocity (in m/s) of
    electrons
printf("\nvelocity of electrons is %1.2e m/s",v)
//the answers vary due to round off error
```

**Scilab code Exa 5.3** `calculation of Plancks constant`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.3
//calculation of Planck's constant
//given data
e=1.6*10^-19; //charge of electron (in C)
//first case
V1=4.6; //stopping potential (in V)
nu1=2*10^15; //frequency (in Hz)
//second case
```

```
V2=12.9; //stopping potential (in V)
nu2=4*10^15; //frequency (in Hz)
//calculation
h=(e*(V2-V1))/(nu2-nu1); //Planck's constant (in Js)
printf("\nPlancks constant is %1.2e J.s",h)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 5.4** calculation of de Broglie wavelength

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.4
//calculation of de Broglie wavelength
//given data
m0=1.66*10^-27; //mass of proton (in kg)
h=6.62*10^-34; //Planck's constant (in Js)
c=3*10^8; //speed of light (in m/s)
K=1.6*10^-19*10^6; //kinetic energy (in V)
//calculation
lambda=(h*c)/(sqrt(K*(K+2*m0*c^2))); //de Broglie
    wavelength (in m)
printf("\nde Broglie wavelength is %1.2e Armstrong",
    lambda*10^10)
```

**Scilab code Exa 5.5** calculation of uncertainty in the determining position of the

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.5
//calculation of uncertainty in the determining
    position of the electron
```

```
//given data
h=6.62*10^-34; //Planck's constant (in Js)
delta_p=5*10^-27*0.003/100; //uncertainty in
    momentum (in kg-m/s)
//calculation
delta_x=h/(4*%pi*delta_p); //uncertainty in position
     (in m)
printf("\nuncertainty in position is %1.1e m",
    delta_x)
```

**Scilab code Exa 5.6** calculation of uncertainty in position

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.6
//calculation of uncertainty in position
//given data
t=1; //time(in s)
m0=1.67*10^-27; //mass of proton (in kg)
delta_x0=10^-11; //accuracy of position (in m)
h=6.62*10^-34; //Planck's constant (in Js)
//calculation
delta_x=(h*t)/(4*%pi*m0*delta_x0); //uncertainty in
    position (in m)
printf("\nuncertainty in position after 1s is %1.2f
    km",delta_x*10^-3)
```

**Scilab code Exa 5.7** calculation of precision with which position of electron is me

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 5.7
//calculation of precision with which position of
   electron is measured simultaneously and precision
    if electron is replaced by a golf ball
//given data
v=1.88*10^6; //speed (in m/s) of electron
m=9.1*10^-31; //mass (in kg) of electron
h=6.62*10^-34; //Plank's constant (in Js)
v1=40; //speed (in m/s) of golf ball
m1=45*10^-3; //mass (in kg) of golf ball
//calculation
//for electron
p_x=m*v;
del_p_x=1/100*p_x;
del_x=h/(4*%pi*del_p_x); //uncertainty in position
//for golf ball
p_x_dash=m1*v1;
del_p_x_dash=1/100*p_x_dash;
del_x_dash=h/(4*%pi*del_p_x_dash);
printf("\nprecision with which position of electron
   is measured is %1.1f nm",del_x*10^9)
printf("\nprecision with which position of golf ball
    is measured is %1.0e m",del_x_dash)
//the second part of the answer is given wrong in
   the textbook
```

**Scilab code Exa 5.8** calculation of uncertainty in momentum and frequency of photon

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.8
//calculation of uncertainty in momentum and
   frequency of photons transmitted
//given data
```

```
c=3*10^8; //speed (in m/s) of light
delta_t=10^-3; //time (in s)
h=6.6*10^-34; //Planck's constant (in Js)
//calculation
delta_x=c*delta_t; //uncertainty in position (in m)
delta_p=h/(4*%pi*delta_x); //uncertainty in momentum (in kg m/s)
delta_nu=round(1/(4*%pi*delta_t)); //frequency (in Hz) of photons
printf("\nuncertainty in momentum is %1.2e kg m/s",delta_p)
printf("\nfrequency of photons is %d Hz",delta_nu)
```

**Scilab code Exa 5.9** calculation of wavelength of scattered photon and recoil energ

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.9
//calculation of wavelength of scattered photon and recoil energy of the electron
//given data
lambda=0.3*10^-10; //wavelength (in m) of X-ray photon
h=6.6*10^-34; //Planck's constant (in Js)
m0=9.1*10^-31; //mass of electron (in kg)
theta=60; //angle (in degree) of scattering
c=3*10^8; //speed (in m/s) of light
//calculation
lambda0=h/(m0*c); //Compton wavelength (in m)
lambda_dash=lambda+h/(m0*c)*(1-cosd(theta)); //wavelength of scattered photon (in m)
E_k=h*c*(lambda_dash-lambda)/(lambda*lambda_dash); //recoil energy of the electron (in J)
printf("\nwavelength of scattered photon is %1.5f
```

```
    Armstrong",lambda_dash*10^10)
printf("\nrecoil energy of the electron is %1.1f keV
    ",E_k/(1.6*10^-19*10^3))
//the first part of answer varies due to round off
    error
```

**Scilab code Exa 5.10** calculation of maximum percentage change in wavelength

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.10
//calculation of maximum percentage change in
    wavelength
//given data
h=6.6*10^-34; //Planck's constant
lambda=1; //wavelength (in Armstrong) of photon
lambda1=10; //wavelength (in Armstrong) of incident
    photons
m0=9.1*10^-31; //mass of electron (in kg)
c=3*10^8; //speed (in m/s) of light
//calculation
del_lambda_max=(2*h*10^10)/(m0*c); //maximum
    wavelength (in Armstrong)
p=del_lambda_max/lambda*100; //percent change in
    wavelength (in %)
p1=del_lambda_max/lambda1*100; //percent change in
    wavelength (in %)
printf("\npercent change in wavelength 1A is %1.1f
    percent",p)
printf("\npercent change in wavelength 10A is %1.2f
    percent",p1)
```

**Scilab code Exa 5.11** calculation of Compton shift and the energy imparted to recoi

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.11
//calculation of (a)Compton shift (b)the energy
   imparted to recoil electron in joules
//given data
m0=9.1*10^-31; //rest mass (in kg) of electron
c=3*10^8; //velocity (in m/s) of light
h=6.6*10^-34; //Planck's constant (in Js)
theta=90; //angle (in degrees) of scattering
lambda=1*10^-10; //wavelength (in m)
//calculation
del_lambda=h/(m0*c)*(1-cosd(theta)); //wavelength (
   in m)
lambda_dash=lambda+del_lambda; //Compton shift (in m
   )
K=h*c*(1/lambda-1/lambda_dash); //recoil energy of
   electron (in Joules)
printf("\n(a)Compton shift is %1.5f Armstrong",
   lambda_dash*10^10)
printf("\n(b)recoil energy of electron is %1.3e
   Joule",K)
//the answers vary due to round off error
```

**Scilab code Exa 5.12** calculation of angle of scattering

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.12
//calculation of angle of scattering
//given data
```

```
h=6.62*10^-34; //Planck's constant
c=3*10^8; //speed (in m/s) of light
E=150*10^3; //energy of incident X-ray photon (in eV)
E_dash=130*10^3; //energy of scattered photon (in eV)
m0=9.1*10^-31; //rest mass (in kg) of electron
//calculation
lambda=(h*c)/(E*1.6*10^-19); //wavelength (in m) of incident photons
lambda_dash=(h*c)/(E_dash*1.6*10^-19); //wavelength (in m) of scattered photon
theta=acosd(1-(lambda_dash-lambda)/(h/(m0*c))); // angle of scattering from Compton scattering formula
printf("\nangle of scattering is %2.1f degrees", theta)
//the answers vary due to round off error
```

**Scilab code Exa 5.13** calculation of wavelength of scattered gama rays at 90 degree

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.13
//calculation of wavelength of scattered gama rays at 90 degree and energy and direction of recoil electron of Al atom
//given data
h=6.6*10^-34; ////Planck's constant (in Js)
c=3*10^8; //speed (in m/s) of light
E=450*10^3; //energy of gama rays (in J)
theta=90; //angle (in degree) of scattering
m0=9.1*10^-31; //mass of electron (in kg)
//calcultion
```

```
lambda=(h*c)/(E*1.6*10^-19); //wavelength (in m) of incident photon
lambda_dash=lambda+h/(m0*c)*(1-cosd(theta)); //wavelength (in m) from Compton formula
E_dash=(h*c)/(lambda_dash*1.6*10^-19); //energy (in eV) of scattered photon
E_k=E-E_dash; //energy of recoil electron
printf("\nwavelength of scattered gama rays at 90 degree is %1.5f Armstrong",lambda_dash*10^10)
printf("\nenergy of recoil electron is %d keV",E_k*10^-3)
//the first part of the answer varies due to round off error
```

**Scilab code Exa 5.16** to show that the energy which a photon must have so that it m

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.16
//to show that the energy which a photon must have so that it may transfer half of its energy to electron at rest is about 256 keV in Compton scattering experiment
//given data
theta=180; //angle (in degree) of scattering
h=6.62*10^-34; ////Planck's constant (in Js)
c=3*10^8; //speed (in m/s) of light
m0=9.1*10^-31; //mass of electron (in kg)
//calculation
lambda_max=h/(m0*c)*(1-cosd(theta)); //maximum value of wavelength (in m)
hnu_min=round((h*c)*10^-3/((1.6*10^-19)*lambda_max)); //energy of photon (in keV)
printf("\nenergy of photon is %d keV",hnu_min)
```

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

**Scilab code Exa 5.19** calculation of probability of finding the particle

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.19
//calculation of probability of finding the particle
//given data
L=10e-010; //length of box (in m)
x1=0.45*L; //lower limit
x2=0.55*L; //upper limit
P=integrate('2/L*(sin(%pi*x/L))^2','x',x1,x2)
printf("\nprobability of finding the particle is %2
    .1f percent",P*100)
```

**Scilab code Exa 5.20** calculation of lowest energy of an electron

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.20
//calculation of lowest energy of an electron
//given data
h=6.62*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of electron (in kg)
a=1*10^-10; //length (in m)
//calculation
El=h^2/(8*m*a^2); //lowest energy (in Joule)
printf("\nlowest energy is %0.1f eV",El
    /(1.602*10^-19))
//the answers vary due to round off error
```

**Scilab code Exa 5.21** calculation of fraction of time an electron spends in central

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.21
//calculation of fraction of time an electron spends in central one-third of the well
//given data
a=1; //For simplicity assuming length of well to be unity
//calculation
x1=a/3; //lower limit
x2=2*a/3; //upper limit
P=integrate('2/a*(sin(%pi*x/a))^2','x',x1,x2)
printf("\nfraction of time an electron spends in central one-third of the well is %d percent",P*100)
//the answers vary due to round off error
```

**Scilab code Exa 5.22** calculation of release of energy

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.22
//calculation of release of energy
//given data
m=1.6*10^-27; //mass of proton (in kg)
a=1.0*10^-14; //size of nucleus (in  m)
h=6.62*10^-34; //Planck's constant (in Js)
```

```
n=1; //for ground state
e=1.6*10^-19; //electronic charge (in C)
n_dash=2; //for excited state
//calculation
del_E=h^2/(8*m*a^2)*(n_dash^2-n^2); //release of
    energy (in J)
printf("\nrelease of energy is %0.1f MeV",(del_E
    *10^-6)/e)
```

**Scilab code Exa 5.26** calculation of zero point energy and energy gap between first

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.26
//calculation of zero point energy,energy gap
    between first two energy states and wavelength of
     absorption
//given data
h=6.626*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of an electron (in kg)
a=5*10^-10; //length of molecule (in m)
e=1.6*10^-19; //electronic charge (in C)
c=3*10^8; //speed of light (in m/s)
//calculation
E_1=h^2/(8*m*a^2); //zero point energy (in J)
del_E=4*h^2/(8*m*a^2)-h^2/(8*m*a^2); //energy gap (
    in J) between first two energy states
lambda=(h*c)/(4.5*e); //wavelength (in m) of
    absorption
printf('\nzero point energy is %0.1f eV',E_1/e)
printf('\nenergy gap (in eV) between first two
    energy states is %0.1f eV',del_E/e)
printf('\nwavelength of absorption is %d Armstrong',
    lambda*10^10)
```

**Scilab code Exa 5.28** calculation of energy and momentum for ground state and first

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.28
//calculation of energy and momentum for ground
    state and first excited state
//given data
a=1*10^-10; //side of cubical box (in m)
m=9.1*10^-31; //mass (in kg) of an electron
h=6.626*10^-34; //Planck's constant (in Js)
e=1.6*10^-19; //electronic charge (in C)
//for ground state
nx=1; //quantum number in x direction
ny=1; //quantum number in y direction
nz=1; //quantum number in z direction
//for first excited state
nx1=2; //quantum number in x direction
ny1=1; //quantum number in y direction
nz1=1; //quantum number in z direction
//calculation
E_111=h^2/(8*m*a^2*e)*(nx^2+ny^2+nz^2); //energy of
    the particle (in eV) for ground state
P_111=sqrt(2*m*E_111*e); //momentum (in kg m/s) for
    ground state
E_211=h^2/(8*m*a^2*e)*(nx1^2+ny1^2+nz1^2); //energy
    of the particle (in eV) for first excited state
P_211=sqrt(2*m*E_211*e); //momentum (in kg m/s) for
    first excited state
printf('\nenergy of the particle for ground state is
     %d eV',E_111)
printf('\nmomentum for ground state is %1.2e kg m/s'
    ,P_111)
```

```
printf('\nenergy of the particle for first excited
    state is %d eV',E_211)
printf('\nmomentum for first excited state is %1.2e
    kg m/s',P_211)
```

**Scilab code Exa 5.30** calculation of binding energy of the neutron in the neucleus

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.30
//calculation of binding energy of the neutron in
    the neucleus
//given data
a=2*10^-15; //side of cubical box (in m)
m=1.6*10^-27; //mass (in kg) of an neutron
h=6.626*10^-34; //Planck's constant (in Js)
e=1.6*10^-19; //electronic charge (in C)
//calculation
E_111=(3*h^2)/(8*m*a^2); //binding energy (in J) of
    the neutron in the neucleus
printf("\nbinding energy of the neutron in the
    neucleus is %0.1f Mev",E_111/(e*10^6)')
//the answer provided in the textbook is wrong
```

**Scilab code Exa 5.31** calculation of probability that electron will tunnel through

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.31
```

```
//calculation of probability that electron will tunnel through the barrier if its width is (a) 0.10 nm (b)1.0 nm
//given data
m=9.1*10^-31; //mass (in kg) of electron
U=40*1.6*10^-19
E=30*1.6*10^-19; //energy (in J) of electron
h=1.054*10^-34; //height (in m)
a1=0.1*10^-9; //width (in m) of barrier
a2=1.0*10^-9; //width (in m) of barrier
//(i)when a=0.1 nm
twoalpha_a1=(2*sqrt(2*m*(U-E)))/h*a1
T1=exp(-twoalpha_a1)*100; //probability that electron will tunnel through the barrier if its width is 0.10 nm
//(ii)when a=1.0 nm
twoalpha_a2=(2*sqrt(2*m*(U-E)))/h*a2
T2=exp(-twoalpha_a2); //probability that electron will tunnel through the barrier if its width is 1.0 nm
printf("\n(i)probability that electron will tunnel through the barrier if its width is 0.10 nm is %1.1f percent",T1)
printf("\n(ii)probability that electron will tunnel through the barrier if its width is 1.0 nm is %1.1e",T2)
//the answer (ii) varies due to round off error
```

**Scilab code Exa 5.33** `calculation of width of potential barrier and number of colli`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.33
//calculation of width of potential barrier and
```

```
number of collisions alpha particle makes per second
//given data
A=222; //atomic mass
Z=86-2; //atomic number
E=5.5; //energy (in eV)
e=1.6*10^-19; //charge of an electron
c=3*10^8; //speed (in m/s) of light
m0=9.1*10^-31; //mass of an electron (in kg)
four_pi_epsilon0=1/(9*10^9)
//calculation
r0=1.2*A^(1/3); //radius of nucleus (in fm)
r=(Z*2*e^2)/(E*1.6*10^-13*four_pi_epsilon0)*10^15; //in m
w=(r-r0)*10^-15; //width of the barrier (in m)
mc_sq=E+m0*c^2*10^6
v=c*sqrt(1+1/12^2); //velocity of electron (in m/s)
n=v/(2*r0*10^-15); //number of collisions
printf("\nwidth of potential barrier is %2.2e m",w)
printf("\nnumber of collisions alpha particle makes per second is %1.2e",n)
//the answers vary due to round off error
```

---

**Scilab code Exa 5.34** calculation of de Broglie wavelength of incident electron and

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.34
//calculation of de Broglie wavelength of incident electron and transmission probabilities
//given data
h=6.63*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of an electron (in kg)
U0=6*1.6*10^-19; //barrier height
```

```
a=7*10^-10; //width of barrier (in m)
E=5*1.6*10^-19; //energy (in eV)
e=1.6*10^-19; //charge of electrons
V=5; //in eV
//calculation
lambda=h/sqrt(2*m*e*V); //de Broglie wavelength of
    incident electron
alpha=sqrt(2*m*(U0-E))/(h/(2*%pi))
T=16*E/U0*(1-E/U0)*exp(-2*alpha*a); //transmission
    probability
printf("\nde Broglie wavelength of incident electron
    is %1.1f Armstrong",lambda*10^10)
printf("\ntransmission probabilities is %1.2e",T)
//the second part of the answer varies due to round
    off error
```

**Scilab code Exa 5.35** calculation of transmission probabilities

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.35
//calculation of transmission probabilities
//given data
h=6.63*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of an electron (in kg)
U0=5*1.6*10^-19; //barrier height
a=5*10^-10; //width of barrier (in m)
E1=1*1.6*10^-19; //first energy of electron
E2=2*1.6*10^-19 //second energy of electrons (in J)
//calculation
//for 1 eV electrons
alpha=sqrt(2*m*(U0-E1))/(h/(2*%pi))
T1=16*E1/U0*(1-E1/U0)*exp(-2*alpha*a); //
    transmission probability of 1 eV electrons
```

```
//for 2 eV electrons
alpha1=sqrt(2*m*(U0-E2))/(h/(2*%pi))
T2=16*E2/U0*(1-E2/U0)*exp(-2*alpha1*a); //transmission probability of 2 eV electrons
//when barrier width is doubled
a_dash=10*10^-10; //width of barrier when doubled (in m)
T1_dash=16*E1/U0*(1-E1/U0)*exp(-2*alpha*a_dash); //transmission probability of 1 eV electrons
T2_dash=16*E2/U0*(1-E2/U0)*exp(-2*alpha1*a_dash); //transmission probability of 2 eV electrons
//when barrier height is doubled
U0_dash=10*1.6*10^-19; //barrier height when doubled
T1_dashdash=16*E1/U0_dash*(1-E1/U0_dash)*exp(-2*alpha*a); //transmission probability of 1 eV electrons
T2_dashdash=16*E2/U0_dash*(1-E2/U0_dash)*exp(-2*alpha1*a); //transmission probability of 2 eV electrons
printf("\ntransmission probability of 1 eV electrons is %1.1e",T1)
printf("\ntransmission probability of 2 eV electrons is %1.1e",T2)
printf("\n(i)transmission probability of 1 eV electrons when barrier width is doubled is %1.1e",T1_dash)
printf("\ntransmission probability of 2 eV electrons when barrier width is doubled is %1.1e",T2_dash)
printf("\n(ii)transmission probability of 1 eV electrons when barrier height is doubled is %1.1e",T1_dashdash)
printf("\ntransmission probability of 2 eV electrons when barrier height is doubled is %1.1e",T2_dashdash)
//the first four answers vary due to round off error
//the last two answers given in the textbook are wrong and the variables are given wrong
```

**Scilab code Exa 5.36** calculation of electron density of the metal

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.36
//calculation of electron density of the metal for
    which the Fermi energy is 4.72 eV
//given data
EF1=2.14; //Fermi energy of potassium (in eV)
EF2=4.72; //Fermi energy of another metal (in eV)
n_e1=1.4*10^28; //electron density (per m^3) of
    potassium
//calculation
n_e2=n_e1*(EF2/EF1)^(3/2); //electron density (per m
    ^3) of the metal
printf("\nelectron density of the metal is %1.2e m
    ^-3",n_e2)
//theanswers vary due to round off error
```

**Scilab code Exa 5.37** calculation of Fermi energy for copper

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.37
//calculation of Fermi energy for copper
//given data
h=6.626*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of electron (in kg)
n=8.49*10^28; //number of conduction electrons per
    unit volume
```

```
e=1.6*10^-19; //electronic charge (in C)
//calculation
E_F=h^2/(2*m)*((3*n)/(8*%pi))^(2/3); //Fermi energy (in J) for copper
printf("\nFermi energy for copper is %0.2f eV",E_F/e)
```

**Scilab code Exa 5.38** calculation of Fermi energy and speed of electron at Fermi en

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.38
//calculation of Fermi energy and speed of electron at Fermi energy
//given data
n_e=5.8*10^28; //electron density (in m^-3)
m=9.1*10^-31; //mass (in kg) of an electron
h=6.6*10^-34; //Planck's constant (in Js)
e=1.602*10^-19; //electronic charge (in C)
//calculation
E_F=(h^2/(2*m)*(3*n_e/(8*%pi))^(2/3))/e; //Fermi energy (in eV)
v_F=sqrt((2*E_F*e)/m); //Fermi velocity (in m/s)
printf("\nFermi energy is %0.1f eV",E_F)
printf("\nFermi velocity is %1.2e m/s",v_F)
```

**Scilab code Exa 5.39** calculation of probability of occupancy for a state

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.39
```

```
//calculation of probability of occupancy for a state whose energy is (a)0.1 eV above Fermi energy (b)0.1 eV below the Fermi energy (c)equal to Fermi energy
//given data
T=800; //temperature (in K)
k=1.38*10^-23; //Boltzmann constant in J/K
E_minus_EF=0.1*1.6*10^-19; //energy above Fermi level
EF_minus_E=-E_minus_EF; //energy below Fermi level
E1_minus_EF=0; //energy equal to Fermi energy
//calculation
a_f_FD=1/(exp(E_minus_EF/(k*T))+1)*100
b_f_FD=round(1/(exp(EF_minus_E/(k*T))+1)*100)
c_f_FD=1/(exp(E1_minus_EF/(k*T))+1)*100
printf("\nprobability of occupancy for a state whose energy is")
printf("\n(a)0.1 eV above Fermi energy is %d percent",a_f_FD)
printf("\n(b)0.1 eV below the Fermi energy is %d percent",b_f_FD)
printf("\n(c)equal to Fermi energy is %d percent",c_f_FD)
```

**Scilab code Exa 5.40** calculation of number of states available for conduction

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.40
//calculation of number of states available for conduction
//given data
h=6.63*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of an electron (in kg)
```

```
E=5*1.6*10^-19; //energy (in joules)
E_dash=5.01*1.6*10^-19; //energy (in joules)
a=10^-2; //edge of copper cube (in m)
//calculation
g_of_E=(8*sqrt(2)*%pi*m^(3/2))/h^3*E^(1/2)*1.6*10^-19; //number of energy states per unit energy
dE=E_dash-E; //energy interval
N=g_of_E*dE*a^3; //number of desired states
del_Eadj=dE/N; //average energy interval between successive level
printf("\nnumber of states available for conduction is %1.0e eV",del_Eadj)
```

---

**Scilab code Exa 5.41** calculation of number of energy states available for conducti

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.41
//calculation of (i)number of energy states available for conduction (ii)density of conduction electrons
//given data
h=6.63*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of an electron (in kg)
V=10^-6; //volume of the cube (in m)
E_F=7*1.6*10^-19; //Fermi energy (in joules)
//calculation
N=(8*sqrt(8))/3*(%pi*V)/h^3*m^(3/2)*E_F^(3/2)
n=((7*8*m*%pi^(2/3))/3^(2/3))^(3/2)
printf("\n(i)number of energy states available for conduction is %1.1e",N)
printf("\n(ii)density of conduction electrons is %1.1e m^3",n)
```

```
//the answer (ii) provided in the textbook is wrong
```

**Scilab code Exa 5.42** calculation of Fermi energy and Fermi velocity and Fermi temp

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 5.42
//calculation of Fermi energy, Fermi velocity and Fermi temperature
//given data
h=6.62*10^-34; //Planck's constant (in Js)
m=9.1*10^-31; //mass of an electron (in kg)
e=1.6*10^-19; //electronic charge (in C)
K_B=1.38*10^-23; //Boltzmann constant (in J/K)
n=5.8*10^28; //number of electrons per unit volume (in m^-3)
//calculation
E_F=(h^2/(2*m)*(3*n/(8*%pi))^(2/3))/e; //Fermi energy (in eV)
v_F=sqrt((2*E_F*e)/m); //Fermi velocity (in m/s)
T_F=(E_F*e)/K_B; //Fermi temperature (in K)
printf("\nFermi energy is %0.2f eV",E_F)
printf("\nFermi velocity is %1.2e m/s",v_F)
printf("\nFermi temperature is %1.2e K",T_F)
```

# Chapter 6

# Special Theory of Relativity

**Scilab code Exa 6.3** calculation of speed

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.3
//calculation of speed
//given data
l=48; //length (in m) of rocket in flight
l0=50; //length (in m) of rocket on ground
c=3*10^8; //speed (in m) of light
//calculation
v=c*sqrt(1-(l/l0)^2); //speed (in m)
v0=v/c
printf("\nspeed is %0.1f times the speed of light",
    v0)
//the answers vary due to round off error
```

**Scilab code Exa 6.4** calculation of per cent contraction in length

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.4
//calculation of per cent contraction in length
//given data
v=0.9; //velocity is 0.9 times the speed of light
theta=45; //inclination (in degrees)
//calculation
l0=1; //assuming l0 to be 1 for simplicity of
    calculation
lx=l0/sqrt(2)*sqrt(1-v^2); //length of rod moving in
    x-direction
ly=l0/sqrt(2); //length of rod moving in y-direction
l=sqrt(lx^2+ly^2); //length
p=(l0-l)/l0; //percent contraction in length
printf("\npercent contraction in length is %2.1f
    percent",p*100)
```

**Scilab code Exa 6.5** calculation of proper halftime of the particles and contracted

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.5
//calculation of proper half-time of the particles
    and contracted length
//given data
T=4.14*10^-8; //improper time (in s)
l_0=12; //length of the lab (in m)
c=3*10^8; //speed of light (in m/s)
//calculation
v=l_0/T; //velocity (in m/s) of the particles
T0=T*sqrt(1-v^2/c^2); //proper half-time (in s)
l=l_0*sqrt(1-v^2/c^2); //contracted length (in m)
```

```
printf("\nproper half−time is %1.2e s",T0)
printf("\ncontracted length is %0.2f m",l)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 6.6** calculation of proper halftime of the particles and traversed

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.6
//calculation of proper half−time of the particles
    and traversed length
//given data
l_0=2; //length (in m) observed by lab observer
T=1.0*10^-8; //improper time (in s)
c=3*10^8; //speed of light (in m/s)
//calculation
v=l_0/T; //velocity (in m/s) of particles
T0=T*sqrt(1-v^2/c^2); //proper half−time (in s)
l=l_0*sqrt(1-v^2/c^2); //contracted length (in m)
printf("\nproper half−time is %1.2e s",T0)
printf("\ncontracted length is %0.2f m",l)
```

**Scilab code Exa 6.7** calculation of speed of an object

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.7
//calculation of speed of an object
l0=1; //assumption made for simplicity of
    calculation
c=3*10^8; //speed (in m/s) of light
```

```
//calculation
//for 50% contraction
v=c*(sqrt(1-(10/(2*10))^2)); //speed (in m/s) of
    object
v0=v/c
printf("\nspeed of object is %1.3f times the speed
    of light",v0)
```

**Scilab code Exa 6.8** calculation of percentage change in the area

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.8
//calculation of percentage change in the area if
    its speed is c/2
//given data
c=3*10^8; //speed (in m/s) of light
v=c/2; //speed (in m/s) of square
P=[(1-v^2/c^2)^(1/2)-1]*100; //P is the percentage
    change in area. In book it is given as (S'-S)/S
    *100
printf("\npercentage change in the area is %0.1f
    percent",P)
printf("\nthus, area decreases by %0.1f percent",-P)
```

**Scilab code Exa 6.9** calculation of time necessary and rest length of the rocket an

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.9
```

```
//calculation of (a)time necessary according to S for the rocket to pass a particular point on the platform (b)rest length of the rocket (c)length D of the platform according to observer S_dash (d)time taken for oberver S to pass the entire length of the rocket, according to S_dash (e)to check whether the ends of the rocket simultaneously line up with the ends of the platform are simultaneous to S_dash
//given data
c=3*10^8; //speed (in m/s) of light
L=65; //length (in m) of platform
//calculation
D0=L; //length (in m) of platform
v=0.8*c; //speed (in m/s) of rocket
T0=L/v; //time (in s) necessary according to S for the rocket to pass a particular point on the platform
L0=L/sqrt(1-v^2/c^2); //proper length (in m)
D=D0*sqrt(1-v^2/c^2); //length of platform (in m) as appeared to S_dash
T_dash=L0/v; //time (in s) taken for oberver S to pass the entire length of the rocket, according to S_dash
del_T_dash=(L0-D)/v; //time interval (in s) to pass the difference in the length of rocket and platform
printf("\n(a))time necessary according to S for the rocket to pass a particular point on the platform is %1.3f microsecond",T0*10^6)
printf("\n(b)rest length of the rocket is %d m",L0)
printf("\n(c)length D of the platform according to observer S_dash is %d m",D)
printf("\n(d)time taken for oberver S to pass the entire length of the rocket, according to S_dash is %1.2f microsecond",T_dash*10^6)
printf("\n(e)time interval (in s) to pass the difference in the length of rocket and platform
```

```
is %1.2f microsecond",del_T_dash*10^6)
//the answer (a) varies due to round off error
```

**Scilab code Exa 6.10** calculation of distance and altitude covered by muons using t

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.10
//calculation of distance and altitude covered by muons using (a)time dilation (b)length contraction
//given data
c=3*10^8; //speed (in m/s) of light
v=0.998*c; //speed (in m/s) of muons
T0=2*10^-6; //mean life (in s) of muons
h0=6000; //altitude (in m)
//calculation
T=T0/sqrt(1-v^2/c^2); //average life (in s)
s=v*T; //distance covered by muons (in m)
h=h0*sqrt(1-v^2/c^2); //altitude covered by muons (in m)
H=9.5*sqrt(1-v^2/c^2); //altitude of 9.5km shall appear to muons as (in km)
printf("\ndistance covered by muons using (a)time dilation is %d m",s)
printf("\ndistance covered by muons using (b)length contraction is %d m",h)
printf("\naltitude of 9.5km shall appear to muons as %1.1f km",H)
//the first part of answer varies due to round off error
```

**Scilab code Exa 6.11** calculation of distance travelled by in one mean time

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.11
//calculation of distance travelled by in one mean
   time
//given data
T0=2*10^-6; //mean life (in s)
v=2.994*10^8; //speed of particle (in m/s)
c=3*10^8; //speed of light (in m/s)
//calculation
T=T0/sqrt(1-v^2/c^2); //time (in s) taken by a
   particle
d=T*v; //distance travelled (in m) by in one mean
   time
printf("\ndistance travelled by in one mean time is
   %0.2f km",d*10^-3)
//the answers vary due to round off error
```

**Scilab code Exa 6.12** calculation of speed of spacecraft

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.12
//calculation of speed of spacecraft
del_t=3600; //time (in s) of a clock on earth
del_t_dash=3599; //time (in s) of a clock on
   spacecraft
c=3*10^8; //speed of light (in m/s)
//calculation
v=c*sqrt(1-del_t_dash^2/del_t^2); //speed (in m/s)
   of spacecraft
```

```
printf("\nspeed of spacecraft is %1.0e m/s",v)
```

**Scilab code Exa 6.13** calculation of speed of muons

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.13
//calculation of speed of muons
//given data
T0=2.2*10^-6; //average life (in s) of muons
T=15.6*10^-6; //time (in s)
c=3*10^8; //speed of light (in m/s)
//calculation
v=c*sqrt(1-T0^2/T^2); //velocity of muons (in m/s)
printf("\nvelocity of muons is %1.2e m/s",v)
```

**Scilab code Exa 6.14** calculation of mean lifetime

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.14
//calculation of mean lifetime
//given data
T0=2.5*10^-8; //proper mean life (in s)
v_by_c=0.73; //speed of mason is 0.73 times speed of
    light
//calculation
T=T0/sqrt(1-v_by_c^2); //mean lifetime (in s)
printf("\nmean lifetime of mason is %1.1e s",T)
```

**Scilab code Exa 6.15** calculation of velocity relative to the earth

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.15
//calculation of velocity relative to the earth
//given data
l0=50; //proper length (in m) of rocket
c=3*10^8; //speed (in m/s) of light
t=2*10^-6; //time (in s) taken to move length l
//calculation
v=c/sqrt(((t*c)/l0)^2+1); //speed (in m/s) of rocket
printf("\nvelocity relative to the earth is %1.2f
   times the speed of light",v/c)
```

---

**Scilab code Exa 6.16** calculation of length and orientation of the rod

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.16
//calculation of length and orientation of the rod
Lx0=5*cosd(30); //component of length in x direction
Ly0=5*sind(30); //component of length in y direction
c=3*10^8; //speed (in m/s) of light
v=0.6*c; //velocity (in m/s) of rod
Lx=Lx0*sqrt(1-v^2/c^2); //length in x direction as
   seen by the observer
Ly=Ly0; //length in y direction as seen by the
   observer
L=sqrt(Lx^2+Ly^2); //length (in m)
```

```
theta=atand(Ly/Lx); //orientation of the rod (in
    degree)
printf("\nlength of rod is %1.1f m",L)
printf("\norientation of the rod is %2.1f degree",
    theta)
//the answers of length vary due to round off error
```

**Scilab code Exa 6.17** calculation of distance travelled

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.17
//calculation of distance travelled
//given data
T0=2.0*10^-8; //half life (in s)
v_by_c=0.96; //speed of mason is 0.967 times speed
    of light
c=3*10^8; //speed (in m/s) of light
//calculation
v=0.96*c; //speed of mason is 0.96 times speed of
    light
T=T0/sqrt(1-v_by_c^2); //half-life (in s)
d=v*T; //distance (in m)
printf("\ndistance travelled is %0.2f m",d)
//the answers vary due to round off error
```

**Scilab code Exa 6.18** calculation of velocity of spaceship

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.18
```

```
//calculation of velocity of spaceship
//given data
a=55; //age of Narayan on earth (in years)
a_dash=45; //age of Narayan in spaceship (in years)
b=75; //age of Raghu on earth (in years)
b_dash=25; //age of Raghu in spaceship (in years)
c=3*10^8; //speed (in m/s) of light
//calculation
T0=a_dash-a; //proper time interval (in years)
T=b-b_dash; //dilated time interval (in years)
v=c*sqrt(1-(T0/T)^2); //velocity (in m/s) of
    spaceship
printf("\nvelocity of spaceship is %1.2e m/s",v)
//the answers vary due to round off error
```

**Scilab code Exa 6.19** calculation of length and diameter of spacecraft measured by

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.19
//calculation of length and diameter of spacecraft
    measured by observer on earth
//given data
L0=120; //length (in m) of spacecraft
c=3*10^8; //speed (in m/s) of light
D0=20; //diameter (in m) of spacecraft
//calculation
v=0.99*c; //speed (in m/s) of spacecraft
L=round(L0*sqrt(1-v^2/c^2)); //length (in m) of
    spacecraft observed on earth
D=D0; //diameter being perperpendicular to motion is
     observed to be the same
printf("\nlength of spacecraft measured by observer
    on earth is %d m",L)
```

```
printf("\ndiameter of spacecraft measured by
    observer on earth is %2.1f m",D)
```

**Scilab code Exa 6.20** calculation of time interval measured by astronaut in his clo

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.20
//calculation of time interval measured by astronaut
     in his clock and time observed by observer on
    earth
//given data
L0=8; //number of light years
c=3*10^8; //speed (in m/s) of light
//calculation
v=0.8*c; //speed (in m/s) of spaceship
L=L0*sqrt(1-v^2/c^2); //contracted length (in light
    years)
del_t=round((L*3*10^8)/v); //time interval (in years
    ) measured by astronaut in his clock
t=10+8; //time (in years) observed by observer on
    earth
printf("\ntime interval measured by astronaut in his
     clock is %d years",del_t)
printf("\ntime observed by observer on earth is %d
    years",t)
```

**Scilab code Exa 6.21** calculation of space and time interval in Sdash frame

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 6.21
//calculation of space and time interval in S_dash
    frame
//given data
del_x=3*10^3; //distance (in m) apart of two sources
     of light A and B
c=3*10^8; //speed (in m/s) of light
t1=1; //assuming t to be 1 for simplicity of
    calculation
//calculation
t2=t1; //since light from A and B flashes
    simultaneously
del_t=t2-t1; //difference between the time flashed
    by A and B
v=c/3; //velocity (in m/s) of observer
alpha=1/sqrt(1-v^2/c^2)
del_x_dash=alpha*del_x; //space interval in S_dash
    frame (in m)
del_t_dash=-alpha*v*del_x/c^2; //time interval (in s
    )
printf("\nspace interval in S_dash frame is %1.2f km
    ",del_x_dash*10^-3)
printf("\ntime interval is %1.2e s",del_t_dash)
//the answers vary due to round off error
```

**Scilab code Exa 6.22** calculation of age of A and B when determined in A s frame of

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.22
//calculation of age of A and B when determined in (
    a)A's frame of reference (b)B's frame of
    reference
//given data
```

```
c=3*10^8; //speed (in m/s) of light
l0=30; //distance of star (in light years)
a=20; //age (in years) of A and B
//calculation
v=0.8*c; //speed (in m/s) of A
//in B's frame of reference
t=(2*l0*3*10^8)/v; //time (in years) spent in the journey
T=t*sqrt(1-v^2/c^2); //time (in years) spend by A
bA=a+T; //age of A (in years)
bB=a+t; //age of B (in years)
//in A's frame of reference
L=l0*sqrt(1-v^2/c^2); //contracted length (in light years)
t_dash=(L*3*10^8)/v; //time (in years) taken in outward trip
T_dash=T/sqrt(1-v^2/c^2); //time dilation (in years)
aA=a+T; //age of A (in years)
aB=a+T_dash; //age of B (in years)
printf("\n(a)age of A when determined in Bs frame of reference is %d years",bA)
printf("\nage of B when determined in Bs frame of reference is %d years",bB)
printf("\n(b)age of A when determined in As frame of reference is %d years",aA)
printf("\nage of B when determined in As frame of reference is %d years",aB)
```

**Scilab code Exa 6.23** `calculation of time difference`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.23
//calculation of time difference
```

```
//given data
c=3*10^8; //speed (in m/s) of light
del_x=1400*10^3; //nautical distance between Delhi and Mumbai
//calculation
v=0.9*c; //speed (in m/s) of travelling from Delhi to Mumbai
del_t_dash=(-v*del_x/c^2)/sqrt(1-v^2/c^2); //time difference in the reference frme of travelling from Delhi to Mumbai
printf("\ntime difference in the reference frme of travelling from Delhi to Mumbai is %1.1f ms", del_t_dash*10^3)
```

**Scilab code Exa 6.24** to show how time dilation accounts for the given result of di

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.24
//to show how time dilation accounts for the given result of distance of 39 m
//given data
T0=1.77*10^-8; //half-life (in s) of pions
c=3*10^8; //speed (in m/s) of light
//calculation
v=0.99*c; //speed (in m/s) of collimated ion beam
d=T0*v; //distance (in m) travelled by pions in Lab frame
T=T0/sqrt(1-v^2/c^2); //half-life of pion observed in pion frame
d_dash=T*v; //distance (in m) travelled by pions in time T
printf("\nDistance travelled by pions in time T is %2.1f m", d_dash)
```

```
printf("\nThus,time dilation accounts for the given
    result of distance of 39 m")
//the answers vary due to round off error
```

**Scilab code Exa 6.25** calculation of period of time measured by moving observer

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.25
//calculation of period of time measured by moving
    observer
//given data
T0=3; //proper period of pendulum (in s)
c=3*10^8; //speed of light (in m/s)
//calculation
v=0.95*c; //speed of pendulum is 0.95 times speed of
    light
T=T0/sqrt(1-(v/c)^2); //period (in s) of time
    measured by moving observer
printf("\nperiod of time measured by moving observer
    is %1.2f s",T)
//the answers vary due to round off error
```

**Scilab code Exa 6.26** calculation of length of laboratory

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.26
//calculation of length of laboratory
//given data
T=4*10^-8; //proper time (in s)
```

```
l0=10; //length (in m) beam crosses
v=l0/T; //velocity (in m/s)
c=3*10^8; //speed (in m/s)of light
l=l0*sqrt(1-v^2/c^2); //length of laboratory (in m)
printf("\nlength of laboratory is %1.2f m",l)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 6.27** calculation of time taken to reach the star with respect to e

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.27
//calculation of time taken to reach the star with respect to (a)earth (b)spaceship
//given data
v=3*10^6; //speed (in m/s) of spaceship
c=3*10^8; //speed (in m/s) of light
d=10; //distance (in light years) of star
//calculation
x=d*c
T0=x/v; //time (in years) measured by the earth
T=x/v*sqrt(1-v^2/c^2); //time (in years) measured on the spaceship
printf("\ntime measured by the earth is %d years",T0)
printf("\ntime measured on the spaceship is %3.1f years",T)
//the formula given in the textbook is wrong
```

**Scilab code Exa 6.28** calculation of increase of time dilation

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 6.28
//calculation of increase of time dilation
//given data
c=3*10^8; //speed (in m/s) of light
v0=1+5/100; //increase in speed is 5% (in m/s)
//calculation
del_T0=1; //assuming T0 to be 1 for simplicity of
    calculation
v=0.95*c; //speed (in m/s) of clock
del_T=del_T0/sqrt(1-v^2/c^2); //time dilation
v_dash=v0*v; //increasing speed (in m/s)
del_T_dash=del_T0/sqrt(1-(v_dash/c)^2); //new time
    dilation
T=round((del_T_dash-del_T)/del_T*100); //percent
    increase in time dilation
printf("\npercent increase in time dilation is %d
    percent",T)
```

**Scilab code Exa 6.29** calculation of speed of spacecraft moving relative to earth

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.29
//calculation of speed of spacecraft moving relative
    to earth
//given data
T0=3600; //proper time (in s)
T=3601; //dilated time (in s)
c=3*10^8; //speed (in m/s) of light
//calculation
v=c*sqrt(1-(T0/T)^2); //speed (in m/s) of spacecraft
printf("\nspeed of spacecraft moving relative to
```

```
earth is %1.1e m/s",v)
```

**Scilab code Exa 6.30** calculation of shift in the wavelength

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.30
//calculation of shift in the wavelength
//given data
v=6.12*10^7; //speed (in m/s) of a distant galaxy receding from earth
c=3*10^8; //speed (in m/s) of light
lambda0=500; //wavelength (in nm) of green spectral line
//calculation
v_by_c=v/c
lambda=lambda0*sqrt((1+v/c)/(1-v/c)); //wavelength (in nm)
s=round(lambda-lambda0); //shift in wavelength (in nm)
printf("\nshift in the wavelength is %d nm",s)
```

**Scilab code Exa 6.31** calculation of interval at which B receives signals from A AN

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.31
//calculation of (a)interval at which B receives signals from A (b)interval at which C receives signals from A (c)interval at which B receive C's flashes
```

```
//given data
c=3*10^8; //speed (in m/s) of light
T0=6; //interval (in minutes) observer A signals
    with flash light
//calculation
v=0.6*c; //speed (in m/s) of a spaceship
//using longitudinal Doppler's effect
T1=T0; //since there is no relative motion between A
     and B, B receives signal in the same time as A
    flashes
T=T0*sqrt((1+v/c)/(1-v/c)); //interval (in minutes)
    at which C receives signals from A
T_dash=T*sqrt((1-v/c)/(1+v/c)); //interval (in
    minutes)at which B receives signals from C
printf("\n(a)interval at which B receives signals
    from A is %d minutes",T1)
printf("\n(b)interval at which C receives signals
    from A is %d minutes",T)
printf("\n(c)interval at which B receives signals
    from C is %d minutes",T1)
```

**Scilab code Exa 6.32** examination of twin paradox

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.32
//examination of twin paradox
//given data
c=3*10^8; //speed (in m/s) of light
v=0.8*c; //separation speed (in m/s) of A and B on
    onward trip
T0=1; //time period as per Doppler's effect (in year
    )
L_by_v=22.5; //time taken in outward trip (as given
```

```
in ex 6.22)
//calculation
T=T0*sqrt((1+v/c)/(1-v/c)); //time period (in years) on onward trip
T_dash=T0*sqrt((1+v/c)/(1-v/c)); //time period (in years) on return trip
A=L_by_v/3; //signals A receive from B
A_dash=L_by_v*3; //signals A receive in return trip
B_yr=20+7.5+67.5; //age of sister (in years) A perceives
A_yr=20+22.5+22.5; //age of A (in years)
L0_by_v=30/0.8; //no of years B's brother needs to reach the star because the star is is 30 light years away
B1=(L0_by_v+30)/3; //no of signals B receives at the interval of three years for the totl duration
B2=7.5/(1/3); //for the remaining 7.5 years,signals received by B in years
A_age=20+22.5+22.5; //age of A (in years)
B_age=20+37.5+37.5; //age of B (in years)
printf("\nfrom the point of view of the travelling twin, age of A is %d years",A_yr)
printf("\nfrom the point of view of the travelling twin, age of B is %d years",B_yr)
printf("\nfrom the point of view of the earth bound twin, age of A is %d years",A_age)
printf("\nfrom the point of view of the earth bound twin, age of B is %d years",B_age)
printf("\nthus,the observations of both yield the same result in the domain of Doppler effect")
```

**Scilab code Exa 6.33** calculation of velocity of Hydra in moving away from the eart

```
clc;clear;
//OS windows 7
```

```
//scilab 6.0.1
//example 6.33
//calculation of velocity of Hydra in moving away
   from the earth
//given data
lambda=394; //wavelength (in nm) of absorbtion of
   light by ionized calcium atom
lambda_dash=475; //wavelength (in nm) of absorption
   line of Hydra
c=3*10^8; //speed (in m/s) of light
v_r=c*(lambda_dash^2-lambda^2)/(lambda_dash^2+lambda
   ^2); //velocity of Hydra in moving away from the
   earth
printf("\nvelocity of Hydra in moving away from the
   earth is %1.2e m/s",v_r)
```

**Scilab code Exa 6.35** calculation of velocity of particle B relative to A

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.35
//calculation of velocity of particle B relative to
   A
//given data
c=3*10^8; //speed (in m/s) of light
Vx=0; //velocity along x-direction
//calculation
v=0.5*c; //velocity (in m/s) of particle A along x-
   direction
Vy=0.4*c; //velocity (in m/s) of particle B along y-
   direction
Vx_dash=(Vx-v)/(1-(v*Vx)/c^2); //velocity component
   along x-direction of particle B in reference
   frame of A
```

```
Vy_dash=(Vy*sqrt(1-v^2/c^2))/(1-(v*Vx)/c^2); //
   velocity component along y-direction of particle
   B in reference frame of A
V_dash=sqrt(Vx_dash^2+Vy_dash^2); //velocity of
   particle B relative to A (in m/s)
theta=atand(Vx_dash/Vy_dash); //nangle at which it
   is inclined with the y_dash axis along the
   negative x_dash direction (in degrees)
printf("\nvelocity of particle B relative to A is %0
   .2f times the speed of light",V_dash/c)
printf("\nangle at which it is inclined with the
   y_dash axis along the negative x_dash direction
   is %d degree",theta)
```

**Scilab code Exa 6.36** calculation of velocity of electron wrt laboratory observer w

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.36
//calculation of velocity of electron w.r.t
   laboratory observer when (a)electron is ejected
   in the x-direction and (b)it is ejected in the y-
   direction
//given data
//for x-direction
c=3*10^8; //speed (in m/s) of light
Vx_dash=0.8*c; //speed of electron (in m/s) in rest
   frame of the atom
v=0.3*c; //speed (in m/s) of radioactive atom w.r,t
   laboratory in x-direction
//for y-direction
Vx_dash_=0
Vy_dash=0.8*c; //speed of electron (in m/s) in rest
   frame of the atom
```

```
//calculation
Vx=(Vx_dash+v)/(1+(v*Vx_dash)/c^2); //velocity (in m/s) of electron w.r.t laboratory observer when electron is ejected in the x-direction
Vy=(Vy_dash*sqrt(1-v^2/c^2))/(1+(v*Vx_dash_)/c^2); //velocity of electron w.r.t laboratory observer when electron is ejected in the y-direction
Vx_dash1=0
Vx1=(Vx_dash1+v)/(1+(v*Vx_dash1)/c^2); //velocity (in m/s) in x-direction
V=sqrt(Vx1^2+Vy^2); //speed (in m/s) of electron
theta=atand(Vy/Vx1); //angle (in degree) made with x-axis
printf("\n(a)velocity of electron w.r.t laboratory observer when electron is ejected in the x-direction is %1.2f times the speed of light",Vx/c)
printf("\n(b)velocity of electron w.r.t laboratory observer when electron is ejected in the y-direction is %1.2f times the speed of light",Vy/c)
printf("\nvelocity of electron is %0.2f times the speed of light",V/c)
printf("\nangle made with x-axis is %2.1f degree",theta)
```

**Scilab code Exa 6.37** calculation of velocity of spacecraft B relative to the groun

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.37
//calculation of velocity of spacecraft B relative to the ground
c=3*10^8; //speed (in m/s) of light
```

```
Vx_dash=0.5*c; //relative velocity (in m/s) of
    spacecraft B with which it overtakes A
v=0.9*c; //velocity (in m/s) of spacecraft A
Vx=(Vx_dash+v)/(1+(v*Vx_dash)/c^2); //velocity (in m
    /s) of spacecraft B relative to the ground
Vx1=Vx_dash+v; //clasically , velocity of spacecraft
    B relative to the ground , which is
    relativistically impossible
printf("\nvelocity of spacecraft B relative to the
    ground is %0.4f times the speed of light",Vx/c)
printf("\nclasically ,velocity of spacecraft B
    relative to the ground is %0.1f times the speed
    of light , which is relativistically impossible",
    Vx1/c)
```

**Scilab code Exa 6.38** calculation of velocity of rocket as observed from earth movi

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.38
//calculation of velocity of rocket as observed from
    earth moving (a)away from the earth (b)towards
    the earth
//given data
c=3*10^8; //speed (in m/s) of light
Vx_dash=0.8*c; //velocity (in m/s) of rocket
    relative to the spaceship
v=0.5*c; //velocity (in m/s) of spaceship moving
    away from earth
//calculation
V=(Vx_dash+v)/(1+(v*Vx_dash)/c^2); //velocity (in m/
    s) of rocket as observed from earth moving away
    from the earth
V1=(-Vx_dash+v)/(1+(v*-Vx_dash)/c^2); //velocity (in
```

```
m/s) of rocket as observed from earth moving towards the earth
printf("\n(a)velocity of rocket as observed from earth moving away from the earth is %1.2f times the speed of light",V/c)
printf("\n(b)velocity of rocket as observed from earth moving towards the earth is %1.1f times the speed of light",V1/c)
```

**Scilab code Exa 6.39** `calculation of velocity of A relative to B`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.39
//calculation of velocity of A relative to B
//given data
c=3*10^8; //speed (in m/s) of light
Vx=0.9*c; //velocity (in m/s) of particle A in the frame S
v=-0.8*c; //velocity (in m/s) of particle B moving in negative x-direction
//calculation
Vx_dash=(Vx-v)/(1-(v*Vx)/c^2); //velocity (in m/s) of A relative to B
printf("\nvelocity of A relative to B is %0.3f times the speed of light",Vx_dash/c)
if Vx_dash/c>0 then
    disp("velocity of A relative to B is along positive x-direction")
else
    disp("velocity of A relative to B is along negative x-direction")
end
```

**Scilab code Exa 6.40** calculation of velocity of beta particles in lab frame

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.40
//calculation of velocity of beta particles in lab frame
//given data
c=3*10^8; //speed (in m/s) of light
v=0.2*c; //speed (in m/s) of a radioactive nucleus
Vx_dash=0; //speed (in m/s) inn x-direction
Vy_dash=0.6*c; //speed (in m/s) of beta particle relative to the nucleus
//calculation
Vx=(Vx_dash+v)/(1+(v*Vx_dash)/c^2); //component Vx in the lab frame
Vy=(Vy_dash*sqrt(1-v^2/c^2))/(1+(v*Vx_dash)/c^2); //component Vy in the lab frame
V=sqrt(Vx^2+Vy^2); //magnitude of resultant velocity (in m/s)
theta=atand(Vy/Vx); //angle (in degree) made by beta particle with the direction of the nucleus
printf("\nvelocity of beta particles in lab frame is %1.2f times the speed of light",V/c)
printf("\nangle made by beta particle with the direction of the nucleus is %2.1f degrees",theta)
//the answers vary due to round off error
```

**Scilab code Exa 6.41** calculation of rest mass of the bomb

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 6.41
//calculation of rest mass of the bomb
//given data
m_01=2; //rest mass (in kg) of one fragment of
    stationary bomb
m_02=2;//rest mass (in kg) of another fragment of
    stationary bomb
c=3*10^8; //speed of light (in m/s)
//calculation
v1=0.6*c; //speed of one fragment is 0.6 times speed
    of light
v2=0.6*c; //speed of another fragment is 0.6 times
    speed of light
m_0=((m_01*c^2)/sqrt(1-(v1/c)^2)+(m_02*c^2)/sqrt(1-(
    v2/c)^2))/c^2; //rest mass (in kg) of the bomb
printf("\nrest mass of the bomb is %0.1f kg",m_0)
```

**Scilab code Exa 6.42** calculation of momentum of a proton

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.42
//calculation of momentum of a proton
//given data
c=3*10^8; //speed (in m/s) of light
m0=1.67*10^-27; //mass (in kg) of a proton
v=0.86*c; //speed (in m/s) of a proton
//calculation
p=(m0*v)/sqrt(1-v^2/c^2); //momentum (in kg m/s) of
    proton
printf("\nmomentum of a proton is %d MeV/c",p
    *10^-6/(5.36*10^-28))
```

```
//the answers vary due to round off error
```

**Scilab code Exa 6.43** calculation of velocity of particle

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.43
//calculation of velocity of particle
//given data
m0=1; //assuming rest mass to be one for simpicity in solving
c=3.0*10^8; //speed of light (in m/s)
//calculation
m=5*m0; //mass of a particle is five times the rest mass
v=c*sqrt(1-(m0/m)^2); //velocity (in m/s) of the particle
printf("\nvelocity of the particle is %1.3e m/s",v)
```

**Scilab code Exa 6.45** calculation of rest mass of original bomb

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.45
//calculation of rest mass of original bomb
//given data
m_01=1.5; //rest mass (in kg) of one fragment of stationary bomb
m_02=1.5; //rest mass (in kg) of another fragment of stationary bomb
c=3*10^8; //speed of light (in m/s)
```

```
//calculation
v=0.8*c; //speed of one fragment is 0.8 times speed
    of light
m_0=((m_01*c^2)/sqrt(1-(v/c)^2)+(m_02*c^2)/sqrt(1-(v
    /c)^2))/c^2; //rest mass (in kg) of the original
    bomb
printf("\nrest mass of the original bomb is %0.1f kg
    ",m_0)
```

**Scilab code Exa 6.46** calculation of work done

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.46
//calculation of work done
//given data
m0=0.5*10^6; //rest mass (in eV) of an electron
c=3*10^8; //speed of light (in m/s)
e=1.6*10^-19; //value of charge
//calculation
v1=0.6*c; //speed of electron is 0.6 times speed of
    light
v2=0.8*c; //speed of electron is 0.8 times speed of
    light
KE_f=(m0*c^2)/(sqrt(1-(v2/c)^2))-(m0*c^2); //final
    kinetic energy (in eV)
KE_i=(m0*c^2)/(sqrt(1-(v1/c)^2))-(m0*c^2); //initial
     kinetic energy (in eV)
WD=KE_f-KE_i; //work done (in eV)
printf("\nwork done is %1.1e J",WD*e)
//the answer provideed in the textbook is wrong
```

**Scilab code Exa 6.47** calculation of change in mass AND work done on the electron t

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.47
//calculation of (a)change in mass (b)work done on
   the electron to change its velocity (c)
   accelerating potential in volts
//given data
m0=9.1*10^-31; //rest mass (in kg) of an electron
c=3*10^8; //speed (in m/s) of light
//calculation
v1=0.98*c; //initial speed (in m/s) of an electron
v2=0.99*c; //final speed (in m/s) of an electron
m1=m0/sqrt(1-v1^2/c^2); //initial mass (in kg)
m2=m0/sqrt(1-v2^2/c^2); //final mass (i kg)
del_m=m2-m1; //change in mass (in kg)
W=del_m*c^2*6.242*10^12; //work done (in MeV)
V=W*10^6; //accelerating potential (in volts)
printf("\n(a)change in mass is %2.2e kg",del_m)
printf("\n(b)work done is %1.3f MeV",W)
printf("\n(c)accelerting potential is %1.3e volts",V
   )
//answer (b) and (c) vary due to round off error
```

**Scilab code Exa 6.49** calculation of per cent error in classical expression of kine

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.49
//calculation of per cent error in classical
   expression of kinetic energy
//given data
```

```
c=3*10^8; //speed (in m/s) of light
m0=1; //assuming m0 to be 1 for simplicity of
   calculation
//calculation
v=0.5*c; //speed (in m/s) of a body
KE=1/2*m0*v^2; //classical kinetic energy
m=1/sqrt(1-v^2/c^2); //mass of body
KE1=(m-m0)*c^2; //relativistic kinetic energy
P=(KE1-KE)/KE*100; //per cent error in classical
   kinetic energy
printf("\nper cent error in classical kinetic energy
    is %d",P)
//the answers vary due to round off error
```

**Scilab code Exa 6.50** calculation of energy required to remove one neutron

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.50
//calculation of energy required to remove one
   neutron
//given data
BE_C12=7.68; //binding energy per nucleon (in Mev)
   for C12
BE_C13=7.47; //binding energy per nucleon (in Mev)
   for C13
//calculation
E=BE_C13*13-BE_C12*12; //energy (in Mev) required to
    remove one neutron
printf("\nenergy required to remove one neutron is
   %1.2f MeV",E)
```

**Scilab code Exa 6.53** calculation of mass of electron

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.53
//calculation of mass of electron
//given data
m0=9.1*10^-31; //rest mass of electron (in kg)
KE=2*10^6*1.6*10^-19; //kinetic energy (in J)
c=3*10^8; //speed of light (in m/s)
//calculation
m=m0+KE/c^2; //mass (in kg) of an electron
printf("\nmass of an electron is %2.2e kg",m)
//the answers vary due to round off error
```

**Scilab code Exa 6.56** calculation of total energy and kinetic energy

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.56
//calculation of total energy and kinetic energy
//given data
c=3*10^8; //speed (in m/s) of light
m0=(9.1*10^-31)/(1.6*10^-13); //rest mass (in MeV)
    of electron
//calculation
v=0.25*c; //speed (in m/s) of electron
m=m0/sqrt(1-v^2/c^2); //mass (in kg)
E=(m0*c^2)/sqrt(1-v^2/c^2); //total energy (in MeV)
K=m*c^2-m0*c^2; //kinetic energy (in MeV)
printf("\ntotal energy is %1.3f MeV",E)
printf("\nkinetic energy is %1.3f MeV",K)
//the first part of answer varies due to round off
```

```
error
```

**Scilab code Exa 6.59** `calculation of mass of electron`

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 6.59
//calculation of mass of electron
//given data
E=1.5; //kinetic energy (in MeV)
c=3*10^8; //speed (in m/s) of light
m0=9.11*10^-31; //rest mass (in kg) of electron
//calculation
m0csq=(m0*c^2)/(1.6*10^-13)
m=((m0csq+E)*1.6*10^-13)/c^2
printf("\nmass of electron is %2.1e kg",m)
//the answers vary due to round off error
```

# Chapter 7

# Nuclear Radiation Detectors

**Scilab code Exa 7.1** calculation of saturated ion current

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.1
//calculation of saturated ion current
//given data
E_bar=49*10^3; //average energy deposited per unit
    disintegration (eV)
alpha=150*10^3; //total activity of the sample (in
    Bq)
e=1.6*10^-19; //electronic charge (in C)
W=32; //average energy deposited per ion pair in the
    gas (eV)
//calculation
I=(E_bar*alpha*e)/W; //saturated ion current (in
    Ampere)
printf("\nsaturated ion current is %1.2e Ampere",I)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 7.2** calculation of pulse amplitude

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.2
//calculation of pulse amplitude
//given data
W=35; //energy required per ion pair (in eV)
C=10^-10; //capacitance (in F)
E0=10^6; //charge (in eV)
e=1.602*10^-19; //value of charge (in C)
//calculation
n0=E0/W; //number of ion pairs
V0=(n0*e)/C; //pulse amplitude (in V)
printf("\npulse amplitude is %0.1f microvolt",V0
   *10^6)
```

**Scilab code Exa 7.3** calculation of gas multiplication factor

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.3
//calculation of gas multiplication factor
//given data
E0=5*10^3; //energy (in eV) of charged particles
W=26.2; //energy (in eV) required per ion pair
e=1.6*10^-19; //electronic charge (in C)
C=200*10^-12; //capacitance (in F)
V=10^-2; //voltage (in V) (printing mistake in book)
//calculation
n0=E0/W; //number of original ion pairs generated
M=(V*C)/(n0*e); //gas multiplication factor
disp(M,'gas multiplication factor is')
```

**Scilab code Exa 7.4** calculation of ionization current

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.4
//calculation of ionization current
//given data
n=15; //number of alpha particles
E=5*10^6; //energy of alpha particles (in eV)
e=1.6*10^-19; //value of charge
E_dash=35.2; //energy (in eV) needed to produce ion
    pair
//calculation
TE=n*E; //total energy of alpha particles (in eV)
N=TE/E_dash; //number of ion pairs produced
dq_by_dt=N*e; //ionization current (in Ampere)
printf("\nionization current is %1.3f pA",dq_by_dt
    *10^12)
```

**Scilab code Exa 7.5** calculation of change in the voltage of condenser

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.5
//calculation of change in the voltage of condensor
//given data
d=5; //distance moved (in cm)
n_av=3*10^4; //average number of ion pairs
e=1.602*10^-19; //value of charge (in C)
```

```
c=5*10^-12; //capacitance (in F)
//calcultion
n=d*n_av
q=n*e; //charge (in C) deposited on the condenser
V0=q/c; //change in the voltage of condensor (in V)
printf("\nchange in the voltage of condensor is %0.1
    f mV",V0*10^3)
```

**Scilab code Exa 7.6** calculation of output pulse height

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.6
//calculation of output pulse height
//given data
E=11*10^6; //energy (in eV) of alpha particles
E_dash=35.2; //energy (in eV) needed to produce ion
    pair
M=1000; //multiplication factor
e=1.6*10^-19; //value of charge
C=30*10^-12; //effective capacity between the wire
    and the earth (in F)
//calculation
N_T=E/E_dash; //total no. of ion pairs produced
N=N_T*M; //number of secondary ions produced
q=N*e; //charge collected by capacitor (in C)
V_O=q/C; //pulse height (in V)
printf("\noutput pulse height is %1.2f V",V_O)
```

**Scilab code Exa 7.7** calculation of height of pulse

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 7.7
//calculation of height of pulse
//given data
E0=10*10^6; //energy (in eV)  of protons
W=34; //energy (in eV) required per ion pair
M=1000; //gas multiplication factor
e=1.6*10^-19; //electronic charge (in C)
R=10^4; //resistance (in ohm) between the electrodes
t=10*10^-6; //current pulse time (in s)
//calculation
n0=E0/W; //number of primary ions
n_dash=n0*M; //total number of ions in proportionate region
q=n_dash*e; //charge on the electrodes (in C)
i=q/t; //current on the electrodes (in Ampere)
V0=i*R; //height of the pulse (in V)
printf("\nheight of the pulse is %0.3f V",V0)
```

**Scilab code Exa 7.8** calculation of maximum radial field and duration of time the c

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.8
//calculation of maximum radial field and duration of time the counter lasts
//given data
V0=1000; //voltage at which halogen and quenched GM works (in V)
r=0.2*10^-3; //radius (in m) of central wire
b=20*10^-3; //radius (in m) of outer cylinder
n=50; //number of weeks
l=10^9; //certified life of tube (in count)
```

```
h=30; //number of hours
s=60; //seconds
m=3000; //number of counts per minute
//calculation
a=r; //since field is maximum near central wire
E_max=V0/(r*log(b/a)); //maximum radial field (in V/
    m)
t=n*m*h*s; //counts each year
T=l/t; //life of counter
printf("\nmaximum radial field is %1.2e V/m",E_max)
printf("\ntime the counter lasts is %1.2f years",T)
```

**Scilab code Exa 7.9** calculation of actual count rate

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.9
//calculation of actual count rate
////given data
n=1.51*10^4; //number of counts per minute
t_d=(250*10^-6)/60; //dead time (in min)
//calculation
N=n/(1-n*t_d); //count rate (per minute)
T=N+sqrt(N); //total count rate (per minute)
printf("\nactual count rate is %d per minute",T)
//the answers vary due to round off error
```

**Scilab code Exa 7.10** calculation of ionization current

```
clc;clear;
//OS windows 7
//scilab 6.0.1
```

```
//example 7.10
//calculation of ionization current
//given data
N=600/60; //count rate per minute
p=10^8; //number of electrons per count
e=1.6*10^-19; //value of charge (in C)
t=1; //time (in s)
//calculation
n=N*p; //total number of electrons
q=n*e; //charge (in C)
I=q/t; //ionization current (in Ampere)
printf("\nionization current is %1.1e Ampere",I)
```

**Scilab code Exa 7.11** calculation of maximum voltage fluctuation in applied voltage

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.11
//calculation of maximum voltage fluctuation in applied voltage
//given data
n2_minus_n1_by_n_av=0.1/100; //counter error
s=3; //slope of plateau region
//calculation
V2_minus_V1=n2_minus_n1_by_n_av*100*100/s; //maximum voltage fluctuation (in V)
printf("\nmaximum voltage fluctuation in applied voltage is %1.1f V",V2_minus_V1)
```

**Scilab code Exa 7.12** calculation of number of disintegrations per minute within th

```
clc;clear;
```

```
//OS windows 7
//scilab 6.0.1
//example 7.12
//calculation of number of disintegrations per
    minute within the source
//given data
n=5.0*10^4; //number of alpha particles reaching the
     window of counter
A=3.0; //perpendicular area (in sq.cm) of window
d=7.0; //distance (in cm) of nuclide
//calculation
d_ohm=A/d^2; //solid angle subtended by the counter
    at the point source
N=(n*4*%pi)/d_ohm; //number of disintegrations per
    minute within the source (in counts/min)
printf("\nnumber of disintegrations per minute
    within the source is %1.3e counts/min",N)
```

**Scilab code Exa 7.13** calculation of dead time of the counter

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.13
//calculation of dead time of the counter
//given data
nu=80/100; //efficiency of GM counter
n=6000/60; //number of counts per minute
//calculation
t_d=(1-nu)/n; //dead time of the counter (in s)
disp(t_d*10^3,'dead time of the counter (in ms) is')
```

**Scilab code Exa 7.14** calculation of actual count rate

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.14
//calculation of actual count rate
//given data
n=4000; //number of counts per minute
t_d=(300*10^-6)/60; //dead time (in min)
//calculation
N=n/(1-n*t_d); //count rate (per minute)
T=N+sqrt(N); //total count rate (per minute)
printf("\nactual count rate is %d per minute",T)
//the answer provided in the textbook is wrong
```

**Scilab code Exa 7.15** calculation of true count rate and observed count rate when s

```
clc;clear;
//OS windows 7
//scilab 6.0.1
//example 7.15
//calculation of (i)true count rate (ii)observed count rate when source strength is increased bya factor of 10
//given data
t_d=200*10^-6; //dead time of counter (in s)
n=1000; //observed count rate
//calculation
N=n/(1-n*t_d); //count rate
T=N+sqrt(N); //actual count rate (per minute)
n_dash=(N*10)/(1+N*10*t_d); //observed count rate when source strength is increased bya factor of 10
printf("\n(i)true count rate is %d",T)
printf("\n(ii)observed count rate when source strength is increased is %4.1f",n_dash)
```