Scilab Textbook Companion for Modern Engineering Physics by K. Vijaya Kumar, S. Chandralingam[1]

Created by
Ebby George
M.Tech
Electrical Engineering
NITK
College Teacher
None
Cross-Checked by
None

June 7, 2018

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

Book Description

Title: Modern Engineering Physics

Author: K. Vijaya Kumar, S. Chandralingam

Publisher: S. Chand And Company, New Delhi

Edition: 1

Year: 2010

ISBN: 9788121932417

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

Scilab code Exa 1.1 `distance of screen from slits`

```
clear
//
//
//

//Variable declaration
d=0.08*10**-2              //distance between slits(m)
beta1=6*10**-4              //fringe width(m)
c=3*10**8                  //velocity of light(m/sec)
mu=8*10**11*10**3         //frequency(Hz) is in (1/s) where microseconds are converted to seconds

//Calculation
lamda=c/mu                //wavelength(m)
D=beta1*d/lamda             //distance of screen from slits(m)

//Result
printf("\n distance of screen from slits is %0.3f  m ",D)
```

Scilab code Exa 1.2 wavelength of light source

```
clear
//
//
//

//Variable declaration
lamda1=4200*10**-10     //wavelength(m)
beta11=0.64*10**-2       //first fringe width(m)
beta12=0.46*10**-2       //second fringe width(m)

//Calculation
lamda2=lamda1*2*beta12/beta11     //wavelength of
    light source(m)

//Result
printf("\n wavelength of light source is %0.3f
    angstrom",lamda2*10**10)
```

Scilab code Exa 1.3 distance of point on screen from centre

```
clear
//
//
//

//Variable declaration
Y=1*10**-3              //distance between slits(m)
D=1                     //distance between slit and
    screen(m)
d=1*10**-3              //point distance(m)
```

```
lamda=5893*10**-10    //wavelength(angston)

//Calculation
delta1=Y*d/D                  //path difference(m)
Pd=2*%pi*delta1/lamda    //phase difference(radian)
r=(cos(Pd/2))**2        //ratio of intensity
delta2=lamda/4                //path difference(m)
W=delta2*D/d                  //distance of point on screen from centre(m)

//Result
printf("\n ratio of intensity is %0.4f  ",r)
printf("\n distance of point on screen from centre is %0.3f  *10**-4 m",W*10**4)
printf("\n answers in the book varies due to rounding off errors")
```

Scilab code Exa 1.4 ratio of maximum intensity to minimum intensity

```
clear
//
//
//

//Variable declaration
I1=10                         //intensity (Wm**-2)
I2=25                         //intensity (Wm**-2)

//Calculation
a1bya2=sqrt(I1/I2)
ImaxbyImin=(a1bya2+1)**2/(a1bya2-1)**2   //ratio of maximum intensity to minimum intensity

//Result
printf("\n ratio of maximum intensity to minimum
```

```
intensity is %0.3f ",ImaxbyImin)
```

Scilab code Exa 1.6 distance between slits

```
clear
//
//
//

//Variable declaration
lamda=500*10**-9        //wavelength(m)
D=2                     //distance of screen from
    slits(m)
l=5*10**-2              //distance(m)
n=100                   //number of fringes

//Calculation
beta1=l/n
d=lamda*D/beta1           //distance between slits(m)

//Result
printf("\n distance between slits is %0.0f mm",d
    *10**3)
```

Scilab code Exa 1.7 fringe width

```
clear
//
//
//

//Variable declaration
d=0.2*10**-3            //distance between slits(m)
```

```
lamda=550*10**-9        //wavelength(m)
D=1                     //distance of screen from slits(m)

//Calculation
beta1=lamda*D/d         //fringe width(m)

//Result
printf("\n fringe width is %0.3f  mm",beta1*10**3)
```

Scilab code Exa 1.8 `angular position of first maximum`

```
clear
//
//
//

//Variable declaration
n=10
lamda=5460*10**-10            //wavelength(m)
d=0.1*10**-3                  //distance between slits(m)
D=2                           //distance of screen from slits(m)

//Calculation
x10=n*lamda*D/d                       //distance from centre where 10th maximum is obtained(m)
tantheta1=x10/2                       //angular position of 10th maximum(radian)
tantheta1=tantheta1*180/%pi   //angular position of 10th maximum(degrees)
x1=lamda*D/(2*d)                      //distance from centre where 1st maximum is obtained(m)
tantheta2=x1/2                        //angular position
```

```
of 1st maximum(radian)
tantheta2=tantheta2*180/%pi  //angular position of 1st maximum(degrees)

//Result
printf("\n angular position of 10th maximum is %0.2f degrees",tantheta1)
printf("\n angular position of 1st maximum is %0.3f degrees",tantheta2)
```

Scilab code Exa 1.9 `least distance of the point from central maximum`

```
clear
//
//
//

//Variable declaration
lamda1=650*10**-9        //wavelength(m)
lamda2=500*10**-9        //wavelength(m)
n1=10
n2=13
D=1                      //distance(m)
d=0.5*10**-3             //seperation(m)

//Calculation
x=n1*lamda1*D/d          //least distance of the point from central maximum(m)

//Result
printf("\n least distance of the point from central maximum is %0.0f mm",x*10**3)
```

Scilab code Exa 1.10 thcikness of glass plate

```
clear
//
//
//

//Variable declaration
n=5
lamda=4800*10**-10          //wavelength(m)
mew_mewdash=0.3

//Calculation
t=n*lamda/mew_mewdash             //thcikness of glass
    plate(m)

//Result
printf("\n thcikness of glass plate is %0.3f
    *10**-6 m",t*10**6)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 1.11 refractive index of oil

```
clear
//
//
//

//Variable declaration
v=0.2           //volume(cc)
a=1*10**4       //area(cm**2)
r=0
n=1
lamda=5.5*10**-5        //wavelength(cm)
t=2
```

```

//Calculation
d=v/a        //thickness of film(cm)
mew=n*lamda/(2*t*cos(r))       //refractive index of oil

//Result
printf("\n refractive index of oil is %0.2f  *10**-5",mew*10**5)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 1.12 wavelength of light for fourth order

```
clear
//
//
//

//Variable declaration
mew=1.33         //refractive index
i=35*%pi/180       //angle of incidence(radian)
d=5*10**-5             //thickness(cm)
n1=1                   //order
n2=2                   //order
n3=3                   //order
n4=4                   //order

//Calculation
r=180/%pi*asin(sin(i)/mew)      //angle of reflection (degrees)
lamda1=2*mew*d*cos(r)/n1             //wavelength of light for 1st order(cm)
lamda2=2*mew*d*cos(r)/n2             //wavelength of light for 2nd order(cm)
lamda3=2*mew*d*cos(r)/n3             //wavelength of
```

```
light for 3rd order(cm)
lamda4=2*mew*d*cos(r)/n4                //wavelength of light for 4th order(cm)

//Result
printf("\n wavelength of light for 1st order is %0.1f  *10**-5 cm",lamda1*10**5)
printf("\n answer in the book varies due to rounding off errors")
printf("\n wavelength of light for 2nd order is %0.2f  *10**-5 cm",lamda2*10**5)
printf("\n wavelength of light for 3rd order is %0.2f  *10**-5 cm",lamda3*10**5)
printf("\n wavelength of light for 4th order is %0.1f  *10**-5 cm",lamda4*10**5)
```

Scilab code Exa 1.13 `fringe width`

```
clear
//
//
//

//Variable declaration
x=15                //distance(cm)
d=0.005             //diameter(cm)
lamda=6000*10**-8       //wavelength(cm)

//Calculation
alpha=d/x                   //angle(radian)
beta1=lamda/(2*alpha)       //fringe width(cm)

//Result
printf("\n fringe width is %0.3f  cm",beta1)
```

Scilab code Exa 1.14 distance from edge of the wedge

```
clear
//
//
//

//Variable declaration
alpha=0.01                //angle(radian)
n=10
lamda=6000*10**-10        //wavelength(m)

//Calculation
x=((2*n)-1)*lamda/(4*alpha)      //distance from edge
    of the wedge(m)

//Result
printf("\n distance from edge of the wedge is %0.3f
    *10**-4 m",x*10**4)
```

Scilab code Exa 1.15 diameter of fringe

```
clear
//
//
//

//Variable declaration
lamda=5460*10**-8        //wavelength(cm)
f=400               //focal length(cm)
n=5
mew=1.5             //refractive index
```

```
//Calculation
R=2*f*(mew-1)        //radius(cm)
Dn=sqrt(2*((2*n)-1)*lamda*R)     //diameter of 5th
    fringe(cm)

//Result
printf("\n diameter of 5th fringe is %0.2f  m",Dn)
```

Scilab code Exa 1.16 `the wavelengths reflected`

```
clear
//
//
//

//Variable declaration
t=500*10**-9        //thickness(m)
f=400               //focal length(cm)
n1=0
n2=1
n3=2
n4=3
mew=1.33            //refractive index

//Calculation
lamda1=4*mew*t/((2*n1)+1)      //wavelength in
    infrared region(m)
lamda2=4*mew*t/((2*n2)+1)      //wavelength in
    infrared region(m)
lamda3=4*mew*t/((2*n3)+1)      //wavelength in
    visible region(m)
lamda4=4*mew*t/((2*n4)+1)      //wavelength in
    ultraviolet region(m)

```

```
//Result
printf("\n wavelength in infrared region is %0.0f *10**-10 m",lamda1*10**10)
printf("\n wavelength in infrared region is %0.1f *10**-10 m",lamda2*10**10)
printf("\n wavelength in visible region is %0.0f *10**-10 m",lamda3*10**10)
printf("\n wavelength in ultraviolet region is %0.0f *10**-10 m",lamda4*10**10)
printf("\n of all the wavelengths reflected %0.3f angstrom is the wavelength in the visible region",lamda3*10**10)
```

Scilab code Exa 1.17 `order of interference`

```
clear
//
//
//

//Variable declaration
i=60*%pi/180          //angle of incidence(radian)
mew=1.33              //refractive index
t=1.5*10**-6          //thickness(m)
lamda=5*10**-7        //wavelength(m)

//Calculation
r=(180/%pi)*asin(sin(i)/mew)      //angle of reflection(degrees)
r=(r)*%pi/180               //angle of reflection(degrees)

n=2*mew*t*cos(r)/lamda                    //order of interference

```

```
//Result
printf("\n order of interference is %0.3f  ",n)
```

Scilab code Exa 1.18 smallest thickness of the plate

```
clear
//
//
//

//Variable declaration
mew=1.5              //refractive index
lamda=5890*10**-10      //wavelength(m)
r=60*%pi/180         //angle of reflection(radian)

//Calculation
t=lamda/(2*mew*cos(r))         //smallest thickness of
    the plate(m)

//Result
printf("\n smallest thickness of the plate is %0.0f
    angstrom",t*10**10)
```

Scilab code Exa 1.19 diameter of dark ring

```
clear
//
//
//

//Variable declaration
D4=0.4               //diameter of 4th ring(cm)
D12=0.7              //diameter of 12th ring(cm)
```

```
p1=16
p2=8
n=4

//Calculation
x=n*p1/(n*p2)
D20=sqrt((D4**2)+(x*((D12**2)-(D4**2))))        //diameter of 20th dark ring(cm)

//Result
printf("\n diameter of 20th dark ring is %0.3f  cm",D20)
```

Scilab code Exa 1.20 refractive index of the liquid

```
clear
//
//
//

//Variable declaration
D10=1.4              //diameter of 10th ring(cm)
D10_dash=1.27        //changed diameter of 10th ring(cm)

//Calculation
mew=(D10**2)/(D10_dash**2)        //refractive index of the liquid

//Result
printf("\n refractive index of the liquid is %0.3f",mew)
```

Scilab code Exa 1.21 `wavelength of light used`

```
clear
//
//
//

//Variable declaration
D25=0.8         //diameter of 25th ring(cm)
D5=0.3          //diameter of 5th ring(cm)
p=25-5
R=100           //radius of curvature(cm)

//Calculation
Nr=(D25**2)-(D5**2)     //numerator
Dr=4*p*R                //denominator
lamda=Nr/Dr             //wavelength of light used(cm)

//Result
printf("\n wavelength of light used is %0.3f *10**-5 cm",lamda*10**5)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 1.22 `total number of lines in the grating`

```
clear
//
//
//

//Variable declaration
w=0.02              //width(m)
theta=(%pi/180)*(18+(14/60))        //angle(radian)
n=1
```

```
lamda=6.56*10**-7                          //wavelength(m)

//Calculation
M=w*sin(theta)/(n*lamda)    //total number of lines in the grating

//Result
printf("\n total number of lines in the grating is %0.1f  ",M)
printf("\n answer given in the book varies due to rounding off errors")
```

Scilab code Exa 1.23 `required thickness of plate`

```
clear
//
//
//

//Variable declaration
lamda=5890*10**-10                         //wavelength(m)
mew=1.5                       //refractive index
r=60*%pi/180          //angle of reflection(radian)

//Calculation
t=lamda/(2*mew*cos(r))       //required thickness of plate(m)

//Result
printf("\n required thickness of plate is %0.0f angstrom",t*10**10)
```

Scilab code Exa 1.25 `value of slit width`

```
clear
//
//
//

//Variable declaration
lamda=6500           //wavelength(m)
theta=30*%pi/180        //angle(radian)

//Calculation
a=lamda/sin(theta)      //value of slit width(angstrom
    )

//Result
printf("\n value of slit width is %0.0f  angstrom",a
    )
```

Scilab code Exa 1.27 `the wavelengths reflected`

```
clear
//
//
//

//Variable declaration
t=500*10**-9        //thickness(m)
n1=0
n2=1
n3=2
n4=3
mew=1.33            //refractive index

//Calculation
lamda1=4*mew*t/((2*n1)+1)       //wavelength in
    infrared region(m)
```

```
lamda2=4*mew*t/((2*n2)+1)      //wavelength in
   infrared region(m)
lamda3=4*mew*t/((2*n3)+1)      //wavelength in
   visible region(m)
lamda4=4*mew*t/((2*n4)+1)      //wavelength in
   ultraviolet region(m)

//Result
printf("\n wavelength in infrared region is %0.0f
   *10**-10 m",lamda1*10**10)
printf("\n wavelength in infrared region is %0.1f
   *10**-10 m",lamda2*10**10)
printf("\n wavelength in visible region is %0.0f
   *10**-10 m",lamda3*10**10)
printf("\n wavelength in ultraviolet region is %0.0f
     *10**-10 m",lamda4*10**10)
printf("\n of all the wavelengths reflected %0.3f
   angstrom is the wavelength in the visible region"
   ,lamda3*10**10)
```

Scilab code Exa 1.28 order of interference

```
clear
//
//
//

//Variable declaration
i=60*%pi/180        //angle of incidence(radian)
mew=1.33          //refractive index
t=1.5*10**-6      //thickness(m)
lamda=5*10**-7    //wavelength(m)

//Calculation
r=(180/%pi)*asin(sin(i)/mew)      //angle of
```

```
   reflection(degrees)
r=(r)*%pi/180           //angle of reflection(
   degrees)

n=2*mew*t*cos(r)/lamda                       //order of
    interference

//Result
printf("\n order of interference is %0.3f  ",n)
```

Scilab code Exa 1.29 `smallest thickness of the plate`

```
clear
//
//
//

//Variable declaration
mew=1.5            //refractive index
lamda=5890*10**-10     //wavelength(m)
r=60*%pi/180       //angle of reflection(radian)

//Calculation
t=lamda/(2*mew*cos(r))        //smallest thickness of
    the plate(m)

//Result
printf("\n smallest thickness of the plate is %0.0f
    angstrom",t*10**10)
```

Scilab code Exa 1.30 `thickness of air film`

```
clear
```

```
//
//
//

//Variable declaration
Dn=2*10**-3              //diameter of ring(m)
n=10
lamda=500*10**-9         //wavelength(m)

//Calculation
R=Dn**2/(4*n*lamda)        //radius(m)
t=Dn**2/(8*R)              //thickness of air film(m)

//Result
printf("\n thickness of air film is %0.3f  micro m",
   t*10**6)
```

Scilab code Exa 1.31 wavelength of light

```
clear
//
//
//

//Variable declaration
D5=0.336*10**-2         //diameter of 5th ring(m)
D15=0.59*10**-2         //diameter of 15th ring(m)
m=10
R=1         //radius of curvature(m)

//Calculation
lamda=((D15**2)-(D5**2))/(4*m*R)       //wavelength of
    light(m)

//Result
```

```
printf("\n wavelength of light is %0.0f  nm",lamda*10**9)
```

Scilab code Exa 1.32 radius of curvature of lens

```
clear
//
//
//

//Variable declaration
D10=0.5*10**-2        //diameter of 10th ring(m)
n=10
lamda=5900*10**-10    //wavelength(m)

//Calculation
R=D10**2/(4*n*lamda)       //radius of curvature of lens(m)

//Result
printf("\n radius of curvature of lens is %0.3f  m",R)
```

Scilab code Exa 1.34 refractive index of the liquid

```
clear
//
//
//

//Variable declaration
D10=1.4             //diameter of 10th ring(cm)
```

```
D10_dash=1.27       //changed diameter of 10th ring(cm
    )

//Calculation
mew=(D10**2)/(D10_dash**2)        //refractive index
    of the liquid

//Result
printf("\n refractive index of the liquid is %0.3f
    ",mew)
```

Scilab code Exa 1.36 order

```
clear
//
//
//

//Variable declaration
lamda1=6*10**-5       //wavelength(cm)
lamda2=4.5*10**-5       //wavelength(cm)
n1=21

//Calculation
n2=n1*lamda1/lamda2      //order

//Result
printf("\n order is %0.3f  ",n2)
```

Scilab code Exa 1.37 slit separation

```
clear
//
```

```
//
//

//Variable declaration
lamda=51*10**-6         //wavelength(cm)
D=200                //separation between screen and slit(cm)
beta1=1              //fringe width(cm)
n=10

//Calculation
d=lamda*D/beta1        //slit separation(cm)

//Result
printf("\n slit separation is %0.3f  m",d*100)
```

Scilab code Exa 1.38 thickness of mica sheet

```
clear
//
//
//

//Variable declaration
D=50              //separation between screen and slit(cm)
x=0.2             //fringe shift(cm)
d=0.1             //separation between slits(cm)
mew=1.58          //refractive index

//Calculation
tow=x*d/(D*(mew-1))      //thickness of mica sheet(cm)

//Result
```

```
printf("\n thickness of mica sheet is %0.3f  *10**-4
    cm",tow*10**4)
```

Scilab code Exa 1.39 fringe width

```
clear
//
//
//

//Variable declaration
lamda=5000*10**-8      //wavelength(cm)
D=50                  //separation between screen and
    slit(cm)
d=0.05                //separation between slits(cm)

//Calculation
beta1=lamda*D/d       //fringe width(cm)

//Result
printf("\n fringe width is %0.3f  cm",beta1)
```

Scilab code Exa 1.40 wavelength

```
clear
//
//
//

//Variable declaration
D=180                 //separation between screen and
    slit(cm)
d=0.04                //separation between slits(cm)
```

```
beta1=0.3              //fringe width(cm)

//Calculation
lamda=(beta1*d*10**4/D)        //wavelength(cm)


//Result
printf("\n wavelength is %0.0f  angstrom",lamda
    *10**4)
```

Scilab code Exa 1.41 `wavelength`

```
clear
//
//
//

//Variable declaration
D=80                  //separation between screen and
    slit(cm)
d=0.1                 //separation between slits(cm)
beta1=0.04             //fringe width(cm)

//Calculation
lamda=beta1*d/D        //wavelength(cm)

//Result
printf("\n wavelength is %0.0f  angstrom",lamda
    *10**8)
```

Scilab code Exa 1.42 `fringe width`

```
clear
```

```
//
//
//

//Variable declaration
lamda=5000*10**-8       //wavelength(cm)
D=50                 //separation between screen and
   slit(cm)
d=0.05               //separation between slits(cm)

//Calculation
beta1=lamda*D/d        //fringe width(cm)

//Result
printf("\n fringe width is %0.3f  cm",beta1)
```

Scilab code Exa 1.43 thickness of soap film

```
clear
//
//
//

//Variable declaration
lamda=7*10**-5       //wavelength(cm)
n=2
mew=1.33             //refractive index

//Calculation
t=(((2*n)+1)*lamda/2)/(2*mew)      //thickness of
   soap film(cm)

//Result
printf("\n thickness of soap film is %0.4f  *10**-5
   cm",t*10**5)
```

Scilab code Exa 1.45 thickness

```
clear
//
//
//

//Variable declaration
lamda=5000*10**-8        //wavelength(cm)
n=16
mew=1.56        //refractive index

//Calculation
t=n*lamda/(mew-1)             //thickness(cm)

//Result
printf("\n thickness is %0.1f  *10**-4 cm",t*10**4)
```

Scilab code Exa 1.46 least thickness of glass plate

```
clear
//
//
//

//Variable declaration
lamda=6000*10**-8        //wavelength(cm)
n=1
mew=1.5        //refractive index
r=50*%pi/180      //angle of refraction(radian)

```

```
//Calculation
t=n*lamda/(2*mew*cos(r))            //least thickness
    of glass plate(cm)

//Result
printf("\n least thickness of glass plate is %0.2f
    *10**-5 cm",t*10**5)
```

Scilab code Exa 1.47 `least thickness of glass plate`

```
clear
//
//
//

//Variable declaration
lamda=5000*10**-8       //wavelength(cm)
mew=1.5       //refractive index
beta1=1        //assume
S=6*beta1

//Calculation
t=S*lamda/(beta1*(mew-1))            //least thickness
     of glass plate(cm)

//Result
printf("\n least thickness of glass plate is %0.0f
    *10**-4 cm",t*10**4)
```

Scilab code Exa 1.48 `refractive index of liquid`

```
clear
//
```

```
//
//

//Variable declaration
D8=1.42      //diameter of 8th ring(cm)
D8dash=1.25      //changed diameter of 8th ring(cm)

//Calculation
mew=D8**2/D8dash**2      //refractive index of liquid

//Result
printf("\n refractive index of liquid is %0.2f  ",
   mew)
```

Scilab code Exa 1.49 thickness of thinnest film

```
clear
//
//
//

//Variable declaration
lamda=6000*10**-8      //wavelength(cm)
n=1
mew=1.33      //refractive index
r=0*%pi/180     //angle of incidence(radian)

//Calculation
t=n*lamda/(2*mew*cos(r))          //thickness of
   thinnest film(cm)

//Result
printf("\n thickness of thinnest film is %0.4f
   *10**-5 cm",t*10**5)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 1.50 `radius of curvature of lens`

```
clear
//
//
//

//Variable declaration
lamda=6000*10**-8      //wavelength(cm)
Dm=0.65      //diameter of 8th ring(cm)
Dn=0.35      //changed diameter of 8th ring(cm)

//Calculation
R=(Dm**2-Dn**2)/(4*lamda)      //radius of curvature of lens(cm)

//Result
printf("\n radius of curvature of lens is %0.0f  cm",R)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 1.52 `diameter of ring`

```
clear
//
//
//

//Variable declaration
m1=15
n=5
```

```
m2=25
D15=0.62      //diameter of 15th ring(cm)
D5=0.3        //diameter of 5th ring(cm)

//Calculation
x=D15**2-D5**2
y=m1-n
z=m2-n
r=4*z/(4*y)
D25=sqrt((r*x)+(D5**2))      //diameter of 25th ring(cm)

//Result
printf("\n diameter of 25th ring is %0.3f  cm",D25)
```

Scilab code Exa 1.53 `radius of curvature of lens`

```
clear
//
//
//

//Variable declaration
lamda=5890*10**-8      //wavelength(cm)
Dm=0.590      //diameter of 8th ring(cm)
Dn=0.336      //changed diameter of 8th ring(cm)
m=15
n=5

//Calculation
R=(Dm-Dn)/(4*lamda*(m-n))      //radius of curvature of lens(cm)

//Result
printf("\n radius of curvature of lens is %0.1f  cm"
```

```
,R)
```

Scilab code Exa 1.54 wavelength of light

```
clear
//
//
//

//Variable declaration
R=70                  //radius of curvature of lens(cm)
n=10
Dn=0.433              //diameter of 10th dark ring(cm)

//Calculation
lamda=Dn**2/(4*R*n)      //wavelength of light(cm)

//Result
printf("\n wavelength of light is %0.3f  *10**-5 cm"
   ,lamda*10**5)
printf("\n answer given in the book varies due to
   rounding off errors")
```

Chapter 2

diffraction

Scilab code Exa 2.1 `number of lines per centimeter`

```
clear
//
//
//

//Variable declaration
lamda=5*10**-5       //wavelength(cm)
k=2                  //order
theta=30*%pi/180    //angle(radian)

//Calculation
e=k*lamda/sin(theta)      //number of lines(cm)
n=1/e                          //number of lines per centimeter

//Result
printf("\n number of lines per centimeter is %0.3f ",n)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 2.2 `difference in angles of deviation`

```
clear
//
//
//

//Variable declaration
lamda=5000*10**-8        //wavelength(cm)
e=1/6000                 //number of lines(cm)

//Calculation
theta1=asin(lamda/e)*180/%pi         //angle for 1st
    order(degrees)
theta2=asin(3*lamda/e)*180/%pi       //angle for 3rd
    order(degrees)
theta=(theta2)-(theta1)        //difference in angles
    of deviation(degrees)


//Result
printf("\n difference in angles of deviation is %0.3
    f  degrees",theta)
```

Scilab code Exa 2.3 `minimum number of lines per cm`

```
clear
//
//
//

//Variable declaration
```

```
lamda=5890*10**-5         //wavelength(cm)
dlamda=6*10**-5           //difference in wavelength(cm
   )
k=2                       //order
w=2.5                     //width(cm)

//Calculation
N=lamda/(k*dlamda*w)      //minimum number of lines per
    cm

//Result
printf("\n minimum number of lines per cm is %0.2f
   ",N)
printf("\n answer given in the book varies due to
   rounding off errors")
```

Scilab code Exa 2.4 `lines will appear resolved`

```
clear
//
//
//

//Variable declaration
lamda=5890*10**-8         //wavelength(cm)
dlamda=6*10**-8           //difference in wavelength(cm
   )
w=2                       //width(cm)
n=425                     //number of lines on grating
k=2                       //order

//Calculation
N=w*n                     //number of lines on grating
N1=int((lamda/dlamda))            //number of lines
   required for resolution
```

```
N2=int((lamda/(k*dlamda)))                    //number of lines required for resolution


//Result
printf("\n number of lines required for resolution is %0.3f and number of lines on grating is %0.3f ",N1,N)
printf("\n hence lines will not be resolved")
printf("\n number of lines required for resolution is %0.3f and number of lines on grating is %0.3f ",N2,N)
printf("\n hence lines will appear resolved")
```

Scilab code Exa 2.5 `angle of separation`

```
clear
//
//
//

//Variable declaration
lamda1=5016*10**-8       //wavelength(cm)
lamda2=5048*10**-8       //difference in wavelength(cm)
k=2                      //order
n=15000                  //number of lines/inch

//Calculation
e=2.54/n
theta1=asin(2*lamda1/e)*180/%pi     //angle for 1st wavelength(degrees)
theta2=asin(2*lamda2/e)*180/%pi     //angle for 2nd wavelength(degrees)
```

```
theta=int(60*(theta2-theta1))          //angle of
    separation(minutes)

//Result
printf("\n angle of separation is %0.3f  minutes",
    theta)
```

Scilab code Exa 2.6 dispersive power of grating

```
clear
//
//
//

//Variable declaration
n=4000                 //number of lines/cm
lamda=5000*10**-8      //wavelength(cm)
k=3                    //order

//Calculation
e=1/n
sintheta=k*lamda/e
costheta=sqrt(1-sintheta**2)
dthetabydlamda=k*n/costheta           //dispersive
    power of grating

//Result
printf("\n dispersive power of grating is %0.3f  ",
    dthetabydlamda)
```

Scilab code Exa 2.7 highest order of spectrum

```
clear
```

```
//
//
//

//Variable declaration
n=5000                    //number of lines/cm
lamda=6000*10**-8         //wavelength(cm)

//Calculation
e=1/n
k=e/lamda                 //highest order of spectrum

//Result
printf("\n highest order of spectrum is %0.3f  ",k)
```

Scilab code Exa 2.8 minimum grating width required

```
clear
//
//
//

//Variable declaration
theta=10*%pi/180                //angle(radian)
dtheta=3*%pi/(60*60*180)                //difference of
    angle(radian)
dlamda=5*10**-9                     //wavelength(cm)
k=2

//Calculation
lamda=sin(theta)*dlamda/(cos(theta)*dtheta)
lamdanew=lamda+dlamda                  //wavelength of
    lines(cm)
N=lamda/(dlamda*k)
Ne=N*k*lamda/sin(theta)        //minimum grating width
```

```
required(cm)

//Result
printf("\n wavelength of lines is %0.1f *10**-8 cm",lamda*10**8)
printf("\n answer given in the book varies due to rounding off errors")
printf("\n minimum grating width required is %0.1f cm",Ne)
```

Scilab code Exa 2.10 `width of central maxima`

```
clear
//
//
//

//Variable declaration
d=2                     //diffraction observed(m)
lamda=500*10**-9        //wavelength(m)
a=1.5*10**-3            //slit width(m)

//Calculation
w=2*d*lamda/a          //width of central maxima(m)

//Result
printf("\n width of central maxima is %0.2f mm",w*10**3)
```

Scilab code Exa 2.11 `slit width`

```
clear
//
```

```
//
//

//Variable declaration
d=2                       //diffraction observed(m)
lamda=500*10**-9          //wavelength(m)
x=5*10**-3                //width of central maxima(m)

//Calculation
a=d*lamda/x               //slit width(m)

//Result
printf("\n slit width is %0.3f mm",a*10**3)
```

Scilab code Exa 2.12 `half angular width`

```
clear
//
//
//

//Variable declaration
lamda=6000*10**-10          //wavelength(m)
a=12*10**-7                 //slit width(m)

//Calculation
theta=asin(lamda/a)*180/%pi          //half angular
   width(degrees)

//Result
printf("\n half angular width is %0.0f degrees",
   theta)
```

Scilab code Exa 2.13 the orders

```
clear
//
//
//

//Variable declaration
b=0.8                       //distance(mm)
a=0.16                      //slit width(mm)
p1=1
p2=2
p3=3

//Calculation
nbyp=(a+b)/a                //ratio of missing orders
n1=int(nbyp*p1)
n2=int(nbyp*p2)
n3=int(nbyp*p3)             //missing orders

//Result
printf("\n the orders %0.3f %0.3f %0.3f  etc will be
    missing",n1,n2,n3)
```

Scilab code Exa 2.14 angular separation

```
clear
//
//
//

//Variable declaration
N=6000*10**2                //number of lines/m
m=3                         //order
lamda1=500*10**-9           //wavelength(m)
```

```
lamda2=510*10**-9        //wavelength(m)

//Calculation
sintheta1=m*N*lamda1
theta1=asin(sintheta1)*180/%pi          //angle(degrees)
sintheta2=m*N*lamda2
theta2=asin(sintheta2)*180/%pi          //angle(degrees)
theta=theta2-theta1                              //angular separation(degrees)

//Result
printf("\n angular separation is %0.2f  degrees",theta)
```

Scilab code Exa 2.15 highest order that can be seen

```
clear
//
//
//

//Variable declaration
N=15000/2.54*10**2    //number of lines/cm
lamda=600*10**-9       //wavelength(m)

//Calculation
m=1/(N*lamda)          //highest order that can be seen

//Result
printf("\n highest order that can be seen is %0.3f",m)
```

Scilab code Exa 2.16 angular separation

```
clear
//
//
//

//Variable declaration
N=10000/2*10**2           //number of lines/m
m=1                        //order
lamda1=5890*10**-10        //wavelength(m)
lamda2=5896*10**-10        //wavelength(m)

//Calculation
sintheta1=m*N*lamda1
theta1=asin(sintheta1)*180/%pi         //angle(
   degrees)
sintheta2=m*N*lamda2
theta2=asin(sintheta2)*180/%pi         //angle(
   degrees)
theta=theta2-theta1                            //
   angular separation(degrees)

//Result
printf("\n angular separation is %0.3f  degrees",
   theta)
```

Scilab code Exa 2.17 slit width

```
clear
//
//
```

```
//

//Variable declaration
theta=15*%pi/180                //angle(radian)
lamda=6500*10**-8                   //wavelength(cm)
n=1                                 //order

//Calculation
a=n*lamda/sin(theta)        //slit width(cm)

//Result
printf("\n slit width is %0.2f *10**-4 cm",a*10**4)
```

Scilab code Exa 2.18 `wavelength of light`

```
clear
//
//
//

//Variable declaration
theta=15*%pi/180                //angle(radian)
a=2.5*10**-6                        //slit width(m)

//Calculation
lamda=a*%pi*sin(theta)*10**10/(1.43*%pi)        //wavelength of light(angstrom)

//Result
printf("\n wavelength of light is %0.0f angstrom",lamda)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 2.19 `wavelength of spectral line`

```
clear
//
//
//

//Variable declaration
n=2                                //order
N=4250                             //grating lines(lines/cm)
theta=30*%pi/180               //angle(radian)

//Calculation
e=1/N
lamda=e*sin(theta)*10**8/n        //wavelength of spectral line(angstrom)

//Result
printf("\n wavelength of spectral line is %0.0f angstrom",lamda)
```

Scilab code Exa 2.20 `angular separation`

```
clear
//
//
//

//Variable declaration
n=1                  //order
a=1*10**-6           //slit width(m)
lamda=600*10**-9     //wavelength of spectral line(m)

//Calculation
```

```
theta=asin(n*lamda/a)*180/%pi        //angular separation(degrees)

//Result
printf("\n angular separation is %0.4f  degrees",theta)
```

Scilab code Exa 2.21 orders can be seen

```
clear
//
//
//

//Variable declaration
N=10520                              //grating lines(lines/cm)
theta=90*%pi/180                //angle(radian)
lamda=5*10**-5                       //wavelength of spectral line(cm)

//Calculation
e=1/N
n=e*sin(theta)/lamda      //order

//Result
printf("\n int((n)) orders can be seen")
```

Scilab code Exa 2.22 slit width

```
clear
//
//
```

```
//

//Variable declaration
x=4.2*10**-3                          //distance(m)
D=60*10**-2                           //screen slit
    distance(m)
lamda=6000*10**-10                    //wavelength(m)

//Calculation
d=D*lamda/x                           //slit width(m)

//Result
printf("\n slit width is %0.3f *10**-4 m",d*10**4)
```

Scilab code Exa 2.23 possible order of spectra

```
clear
//
//
//

//Variable declaration
N=15000/2.54                          //number of lines(
    per cm)
lamda=6000*10**-8                     //wavelength(cm)

//Calculation
d=1/N                                 //slit width(m)
m=d/lamda                             //possible order of
    spectra

//Result
printf("\n possible order of spectra is %0.3f ",m)
```

Scilab code Exa 2.24 `wavelength of light`

```
clear
//
//
//

//Variable declaration
D=150                                //slit screen
   distance(cm)
d=0.03                               //separation(cm)
beta1=0.3                             //fringe separation
   (cm)

//Calculation
lamda=d*beta1*10**8/D                 //wavelength of
   light(angstrom)

//Result
printf("\n wavelength of light is %0.0f angstrom",
   lamda)
```

Chapter 3

polarization

Scilab code Exa 3.1 thickness of quartz half wave plate

```
clear
//
//
//

//Variable declaration
lamda=500*10**-9         //wavelength(m)
mewe=1.553               //refractive index of e-ray
mew0=1.544               //refractive index of o-ray

//Calculation
t=lamda/(2*(mewe-mew0))       //thickness of quartz
    half wave plate(m)

//Result
printf("\n thickness of quartz half wave plate is %0
    .4f mm",t/10**-3)
```

Scilab code Exa 3.2 thickness of quartz half wave plate

```
clear
//
//
//

//Variable declaration
lamda=589*10**-9        //wavelength(m)
mewe=1.553              //refractive index of e−ray
mew0=1.544              //refractive index of o−ray

//Calculation
t=lamda/(4*(mewe-mew0))       //thickness of quartz
    half wave plate(m)

//Result
printf("\n thickness of quartz half wave plate is %0
    .4f  mm",t/10**-3)
```

Scilab code Exa 3.3 `thickness of quartz half wave plate`

```
clear
//
//
//

//Variable declaration
lamda=600*10**-9        //wavelength(m)
mewe=1.5533              //refractive index of e−ray
mew0=1.5442              //refractive index of o−ray

//Calculation
t=lamda/(4*(mewe-mew0))       //thickness of quartz
    half wave plate(m)

//Result
```

```
printf("\n thickness of quartz half wave plate is %0.4f mm",t/10**-3)
```

Scilab code Exa 3.4 thickness of quarter wave plate

```
clear
//
//
//

//Variable declaration
lamda=589.3*10**-9          //wavelength(m)
mewe=1.65833                //refractive index of e-ray
mew0=1.48640                //refractive index of o-ray

//Calculation
t1=lamda/(2*(mewe-mew0))        //thickness of half wave plate(m)
t2=lamda/(4*(mewe-mew0))        //thickness of quarter wave plate(m)

//Result
printf("\n thickness of half wave plate is %0.7f mm",t1/10**-3)
printf("\n thickness of quarter wave plate is %0.6f mm",t2/10**-3)
```

Scilab code Exa 3.5 wavelength for quarter wave plate

```
clear
//
//
//
```

```
//Variable declaration
t=0.9*10**-6          //thickness(m)
mewe=1.486            //refractive index of e-ray
mew0=1.658            //refractive index of o-ray

//Calculation
lamda1=4*t*(mew0-mewe)      //wavelength for half
   wave plate(m)
lamda2=2*t*(mew0-mewe)      //wavelength for quarter
    wave plate(m)

//Result
printf("\n wavelength for half wave plate is %0.3f
   m",lamda1*10**9)
printf("\n wavelength for quarter wave plate is %0.3
   f  mm",lamda2*10**9)
```

Chapter 4

laser

Scilab code Exa 4.1 relative population

```
clear
//
//
//

//Variable declaration
c=3*10**8                   //velocity of light(m/sec)
lamda=6943*10**-10          //wavelength(m)
h=6.626*10**-34             //planck's constant(Jsec)
Kb=1.38*10**-23             //boltzmann constant
T=300                       //temperature(K)

//Calculation
new=c/lamda              //frequency(Hz)
a=h*new/(Kb*T)
N1byN2=exp(a)    //relative population

//Result
printf("\n relative population is %0.3f  *10**30",
   N1byN2/10**30)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 4.2 `power density`

```
clear
//
//
//

//Variable declaration
c=3*10**8                    //velocity of light(m/sec)
lamda=632.8*10**-9           //wavelength(m)
h=6.626*10**-34              //planck's constant(Jsec)
t=1                          //time(sec)
P=2.3*10**-3                 //power(W)
sa=1*10**-6                  //spot area(m**2)

//Calculation
new=c/lamda                  //frequency(Hz)
n=P*t/(h*new)                //number of photons emitted(
    per sec)
Pd=P/sa                      //power density(kW/m**2)

//Result
printf("\n number of photons emitted is %0.2f
    *10**15 photons/second",n/10**15)
printf("\n power density is %0.3f  kW/m**2",Pd
    /10**3)
```

Scilab code Exa 4.3 `wavelength`

```
clear
//
```

```
//
//

//Variable declaration
c=3*10**8                   //velocity of light(m/sec)
e=1.6*10**-19               //charge of electron(coulomb)
Eg=1.44*e                   //band gap energy(J)
h=6.626*10**-34             //planck's constant(Jsec)

//Calculation
lamda=h*c/Eg                //wavelength(m)

//Result
printf("\n wavelength is %0.0f angstrom",lamda*10**10)
```

Scilab code Exa 4.4 band gap

```
clear
//
//
//

//Variable declaration
lamda=1.55                  //peak emission wavelength(micro m)

//Calculation
Eg=1.24/lamda               //band gap(eV)

//Result
printf("\n band gap is %0.3f eV",Eg)
```

Scilab code Exa 4.5 relative population of states

```
clear
//
//
//

//Variable declaration
c=3*10**8                    //velocity of light(m/sec)
e=1.6*10**-19                //charge of electron(coulomb
    )
lamda=6943*10**-10           //wavelength(m)
h=6.6*10**-34                //planck's constant(Jsec)
kb=1.38*10**-23              //boltzmann constant
T=300                        //temperature(K)

//Calculation
Uv=h*c/(e*lamda)             //energy(eV)
Uvj=Uv*e                     //energy(J)
x=Uvj/(kb*T)
NbyN0=exp(x)            //relative population of 2 states

//Result
printf("\n relative population of 2 states is %0.0f
    *10**29",NbyN0*10**-29)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 4.6 ratio of emission

```
clear
//
//
```

```
//

//Variable declaration
c=2.998*10**8                 //velocity of light(m/sec)
lamda=0.5*10**-9              //wavelength(m)
h=6.626*10**-34               //planck's constant(Jsec)
Kb=1.381*10**-23              //boltzmann constant
T=1000                        //temperature(K)

//Calculation
new=c/lamda                   //operating frequency(Hz)
new=new/10**3                 //operating frequency(kHz)
new=(new/10**14)*10**14

x=h*new/(Kb*T)
B21byA21=1/(exp(x)-1)   //ratio of emission

//Result
printf("\n ratio of emission is %0.1f  *10**-13",B21byA21*10**13)
```

Chapter 5

fiber optics

Scilab code Exa 5.1 acceptance angle

```
clear
//
//
//

//Variable declaration
n1=1.50            //Core refractive index
n2=1.47            //Cladding refractive index

//Calculation
phic=asin(n2/n1)       //critical angle(radian)
phic=phic*180/%pi      //critical angle(degrees)
NA=sqrt(n1**2-n2**2)     //numerical aperture
phimax=asin(NA)          //acceptance angle(radian)
phimax=asin(NA)*180/%pi      //acceptance angle(
   degrees)

//Result
printf("\n critical angle is %0.1f  degrees",phic)
printf("\n numerical aperture is %0.1f  ",NA)
printf("\n acceptance angle is %0.1f  degrees",
```

```
phimax)
```

Scilab code Exa 5.2 numerical aperture

```
clear
//
//
//

//Variable declaration
n1=1.46           //Core refractive index
delta=0.05        //relative refractive index difference

//Calculation
NA=n1*sqrt(2*delta)     //numerical aperture

//Result
printf("\n numerical aperture is %0.2f  ",NA)
```

Scilab code Exa 5.3 acceptance angle for skew rays

```
clear
//
//
//

//Variable declaration
NA=0.3      //numerical aperture
gama=45*%pi/180    //angle(radian)

//Calculation
```

```
thetaa1=asin(NA)         //acceptance angle for meridional rays(radian)
thetaa1=thetaa1*180/%pi          //acceptance angle for meridional rays(degrees)
thetaa2=asin(NA/cos(gama))*180/%pi     //acceptance angle for skew rays(degrees)

//Result
printf("\n acceptance angle for meridional rays is %0.2f  degrees",thetaa1)
printf("\n acceptance angle for skew rays is %0.3f degrees",thetaa2)
printf("\n answer for acceptance angle for skew rays given in the textbook varies due to rounding off errors")
```

Scilab code Exa 5.4 acceptance angle

```
clear
//
//
//

//Variable declaration
n1=1.53           //Core refractive index
delta=0.0196        //relative refractive index difference

//Calculation
NA=n1*sqrt(2*delta)     //numerical aperture
thetaa=asin(NA)*180/%pi       //acceptance angle(degrees)

//Result
printf("\n numerical aperture is %0.3f  ",NA)
```

```
printf("\n acceptance angle is %0.3f  degrees",thetaa)
printf("\n answer for acceptance angle given in the textbook varies due to rounding off errors")
```

Scilab code Exa 5.5 total distance travelled by light

```
clear
//
//
//

//Variable declaration
n1=1.5          //Core refractive index
n2=1.49         //Cladding refractive index
a=25*10**-6     //radius of core(m)
L=1             //distance(m)

//Calculation
phic=(asin(n2/n1)*180/%pi )       //critical angle(degrees)

phicr=phic*%pi/180                        //critical angle(radian)
l=2*a*tan(phicr)                     //fibre length(m)
r=1/l                             //number of reflections
od=L/sin(phicr)           //total distance travelled by light(m)

//Result
printf("\n critical angle is %0.3f  degrees",phic)
printf("\n fibre length is %0.2f  micro m",l*10**6)
printf("\n answer for fibre length given in the book is wrong")
printf("\n number of reflections is %0.3f  ",r)
```

```
printf("\n total distance travelled by light is %0.4
    f  m",od)
```

Scilab code Exa 5.7 bandwidth length product

```
clear
//
//
//

//Variable declaration
n=1.55              //Core refractive index
L=10                //length(m)
delta=0.026         //relative refractive index
    difference
C=3*10**5

//Calculation
deltaT=L*n*delta/C          //total dispersion(s)
blp=L/(2*deltaT)            //bandwidth length
    product(Hz km)

//Result
printf("\n total dispersion is %0.1f  ns",deltaT
    *10**9)
printf("\n bandwidth length product is %0.2f  *10**5
     Hz km",blp/10**5)
printf("\n answer for bandwidth length product given
     in the book is wrong")
```

Scilab code Exa 5.8 numerical aperture

```
clear
```

```
//
//
//

//Variable declaration
n1=1.55           //Core refractive index
n2=1.50           //Cladding refractive index

//Calculation
NA=sqrt(n1**2-n2**2)     //numerical aperture

//Result
printf("\n numerical aperture is %0.3f  ",NA)
```

Scilab code Exa 5.9 `acceptance angle`

```
clear
//
//
//

//Variable declaration
n1=1.563           //Core refractive index
n2=1.498           //Cladding refractive index

//Calculation
NA=sqrt(n1**2-n2**2)     //numerical aperture
phimax=asin(NA)          //acceptance angle(radian)
phimax=asin(NA)*180/%pi
phimaxd=int(phimax)           //acceptance angle(
    degrees)
phimaxm=60*(phimax-phimaxd)   //acceptance angle(
    minutes)

//Result
```

```
printf("\n numerical aperture is %0.4f  ",NA)
printf("\n acceptance angle is %0.3f degrees %0.1f minutes",phimaxd,phimaxm)
printf("\n answer for acceptance angle in minutes given in the book varies due to rounding off errors")
```

Scilab code Exa 5.11 `fractional index change`

```
clear
//
//
//

//Variable declaration
n1=1.563      //Core refractive index
n2=1.498      //Cladding refractive index

//Calculation
delta=(n1-n2)/n1    //fractional index change

//Result
printf("\n fractional index change is %0.4f  ",delta)
```

Scilab code Exa 5.12 `angle of acceptance`

```
clear
//
//
//

//Variable declaration
```

```
n1=1.48        //Core refractive index
n2=1.45        //Cladding refractive index

//Calculation
NA=sqrt(n1**2-n2**2)      //numerical aperture
thetamax=asin(NA)*180/%pi
thetamaxd=int(thetamax)         //angle of acceptance(
   degrees)
thetamaxm=60*(thetamax-thetamaxd)    //angle of
   acceptance(minutes)

//Result
printf("\n numerical aperture is %0.4f  ",NA)
printf("\n angle of acceptance is %0.3f degrees %0.0
   f minutes",thetamaxd,thetamaxm)
```

Scilab code Exa 5.13 refractive index of core

```
clear
//
//
//

//Variable declaration
NA=0.39            //numerical aperture
delta=0.05         //fractional index change

//Calculation
n1=NA/sqrt(2*delta)      //refractive index of core

//Result
printf("\n refractive index of core is %0.3f  ",n1)
```

Scilab code Exa 5.14 `fractional index change`

```
clear
//
//
//

//Variable declaration
n1=1.563       //Core refractive index
n2=1.498       //Cladding refractive index

//Calculation
delta=(n1-n2)/n1     //fractional index change

//Result
printf("\n fractional index change is %0.5f  ",delta
   )
```

Scilab code Exa 5.15 `numerical aperture`

```
clear
//
//
//

//Variable declaration
n1=1.55          //Core refractive index
n2=1.50          //Cladding refractive index

//Calculation
NA=sqrt(n1**2-n2**2)     //numerical aperture

//Result
printf("\n numerical aperture is %0.2f  ",NA)
```

Scilab code Exa 5.16 `acceptance angle`

```
clear
//
//
//

//Variable declaration
n1=1.563           //Core refractive index
n2=1.498           //Cladding refractive index

//Calculation
NA=sqrt(n1**2-n2**2)    //numerical aperture
theta0=asin(NA)*180/%pi          //acceptance angle(
   degrees)

//Result
printf("\n numerical aperture is %0.4f  ",NA)
printf("\n acceptance angle is %0.2f  degrees",
   theta0)
```

Scilab code Exa 5.17 `critical angle`

```
clear
//
//
//

//Variable declaration
n1=1.53            //Core refractive index
n2=1.42            //Cladding refractive index

```

```
//Calculation
thetac=asin(n2/n1)*180/%pi          //critical angle(
   degrees)

//Result
printf("\n critical angle is %0.2f  degrees",thetac)
```

Scilab code Exa 5.18 acceptance angle

```
clear
//
//
//

//Variable declaration
n1=1.6          //Core refractive index
n2=1.4          //Cladding refractive index
n0=1.33         //water refractive index

//Calculation
NA=sqrt(n1**2-n2**2)/n0    //numerical aperture
theta0=asin(NA)*180/%pi          //acceptance angle(
   degrees)

//Result
printf("\n numerical aperture is %0.3f  ",NA)
printf("\n acceptance angle is %0.4f  degrees",
   theta0)
```

Scilab code Exa 5.19 fractional index change

```
clear
//
```

```
//
//

//Variable declaration
n1=1.5      //Core refractive index
n2=1.3      //Cladding refractive index

//Calculation
delta=(n1-n2)/n1    //fractional index change

//Result
printf("\n fractional index change is %0.3f  ",delta
   )
```

Scilab code Exa 5.20 `angle of reflection`

```
clear
//
//
//

//Variable declaration
n1=1.55       //Core refractive index
n2=1.6      //Cladding refractive index
theta1=60*%pi/180     //angle of incidence(radian)

//Calculation
x=n1*sin(theta1)/n2
theta2=asin(x)*180/%pi        //angle of reflection(
   degrees)

//Result
printf("\n angle of reflection is %0.2f  degrees",
   theta2)
```

Scilab code Exa 5.21 `Core refractive index`

```
clear
//
//
//

//Variable declaration
n2=1.3       //Cladding refractive index
delta=0.14     //fractional index change

//Calculation
n1=n2/(1-delta)       //Core refractive index

//Result
printf("\n Core refractive index is %0.2f  ",n1)
```

Scilab code Exa 5.22 `numerical aperture`

```
clear
//
//
//

//Variable declaration
theta0=26.80*%pi/180          //acceptance angle(
   radian)

//Calculation
NA=sin(theta0)                //numerical aperture

//Result
```

```
printf("\n numerical aperture is %0.5f  ",NA)
```

Chapter 6

crystallography

Scilab code Exa 6.2 `lattice constant`

```
clear
//
//
//

//Variable declaration
n=2                //number of atoms per unit cell
M=55.85            //atomic weight(amu)
N=6.02*10**23      //avagadro number(kg/m**3)
rho=7860           //density(kg/m**3)

//Calculation
a=(n*M/(rho*N))**(1/3)      //lattice constant(m)

//Result
printf("\n lattice constant is %0.2f  angstrom",a
   *10**8)
```

Scilab code Exa 6.3 `lattice constant`

```
clear
//
//
//

//Variable declaration
n=2                 //number of atoms per unit cell
M=6.94             //atomic weight(amu)
N=6.02*10**26     //avagadro number(kg/mol)
rho=530            //density(kg/m**3)

//Calculation
a=(n*M/(rho*N))**(1/3)      //lattice constant(m)

//Result
printf("\n lattice constant is %0.2f  angstrom",a*10**10)
```

Scilab code Exa 6.4 `number of atoms per unit cell`

```
clear
//
//
//

//Variable declaration
M=55.85             //atomic weight(amu)
N=6.02*10**26      //avagadro number(kg/mol)
rho=7870            //density(kg/m**3)
a=2.9*10**-10      //lattice constant(m)

//Calculation
n=a**3*rho*N/M      //number of atoms per unit cell

//Result
```

```
printf("\n number of atoms per unit cell is %0.3f  "
    ,n)
```

Scilab code Exa 6.5 density

```
clear
//
//
//

//Variable declaration
n=8                //number of atoms per unit cell
a=5.6*10**-10      //lattice constant(m)
M=710.59           //atomic weight(amu)
N=6.02*10**26      //avagadro number(kg/mol)

//Calculation
rho=n*M/(a**3*N)       //density(kg/m**3)

//Result
printf("\n density is %0.0f  kg/m**3",rho)
```

Scilab code Exa 6.6 lattice constant

```
clear
//
//
//

//Variable declaration
n=2                //number of atoms per unit cell
M=55.85            //atomic weight(amu)
N=6.02*10**23      //avagadro number(kg/m**3)
```

```
rho=7860            //density(kg/m**3)

//Calculation
a=(n*M/(rho*N))**(1/3)       //lattice constant(m)

//Result
printf("\n lattice constant is %0.4f  angstrom",a
   *10**8)
```

Scilab code Exa 6.7 lattice constant

```
clear
//
//
//

//Variable declaration
n=2                 //number of atoms per unit cell
M=6.94             //atomic weight(amu)
N=6.02*10**26      //avagadro number(kg/mol)
rho=530            //density(kg/m**3)

//Calculation
a=(n*M/(rho*N))**(1/3)       //lattice constant(m)

//Result
printf("\n lattice constant is %0.3f  angstrom",a
   *10**10)
printf("\n answer given in the book varies due to
   rounding off errors")
```

Scilab code Exa 6.8 number of atoms per unit cell

```
clear
//
//
//

//Variable declaration
M=55.85          //atomic weight(amu)
N=6.02*10**26    //avagadro number(kg/mol)
rho=7870         //density(kg/m**3)
a=2.9*10**-10    //lattice constant(m)

//Calculation
n=a**3*rho*N/M      //number of atoms per unit cell

//Result
printf("\n number of atoms per unit cell is %0.3f  "
    ,n)
```

Scilab code Exa 6.9 density

```
clear
//
//
//

//Variable declaration
r=0.1278*10**-9    //atomic radius(m)
n=4                //number of atoms per unit cell
M=63.5             //atomic weight(amu)
N=6.02*10**26      //avagadro number(kg/mol)

//Calculation
a=sqrt(8)*r    //lattice constant(m)
rho=n*M/(a**3*N)    //density(kg/m**3)

```

```
//Result
printf("\n density is %0.2f  kg/m**3",rho)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 6.12 `distance between adjacent atoms`

```
clear
//
//
//

//Variable declaration
w=23+35.5               //molecular weight of NaCl(gm/
   mole)
N=6.023*10**23          //avagadro number(gm/mol)
rho=2.18                //density of NaCl(gm/cm**3)
n=2                     //number of atoms

//Calculation
m=w/N                   //mass of NaCl(gm)
nm=rho/m                //number of molecules(mole/cm
   **3)
N_NaCl=n*nm             //number of atoms per unit
   volume(atoms/cm**3)
a=(1/N_NaCl)**(1/3)         //distance between adjacent
   atoms(cm)


//Result
printf("\n number of atoms per unit volume is %0.2f
    *10**22 atoms/cm**3",N_NaCl/10**22)
printf("\n distance between adjacent atoms is %0.2f
    angstrom",a*10**8)
```

Scilab code Exa 6.13 glancing angle

```
clear
//
//
//

//Variable declaration
lamda=0.071*10**-9         //wavelength(m)
h=1
k=1
l=0                        //miller indices
a=0.28*10**-9              //lattice constant(m)
n=2                        //order

//Calculation
d=a/sqrt(h**2+k**2+l**2)
theta=asin(n*lamda/(2*d))*180/%pi       //glancing angle(degrees)

//Result
printf("\n glancing angle is %0.2f  degrees",theta)
```

Scilab code Exa 6.14 volume of unit cell

```
clear
//
//
//

//Variable declaration
lamda=3*10**-10        //wavelength(m)
```

```
h=1
k=0
l=0                          //miller indices
theta=40*%pi/180     //glancing angle(radian)
n=1                          //order

//Calculation
d=n*lamda/(2*sin(theta))        //space of reflecting plane(m)
a=d*sqrt(h**2+k**2+l**2)
V=a**3                       //volume of unit cell(m**3)

//Result
printf("\n space of reflecting plane is %0.4f angstrom",d*10**10)
printf("\n volume of unit cell is %0.2f  *10**-29 m**3",V*10**29)
```

Scilab code Exa 6.15 space of reflecting plane

```
clear
//
//
//

//Variable declaration
lamda=0.82        //wavelength(m)
theta=75.86*%pi/180     //glancing angle(radian)
n=1                          //order
a=3                          //lattice constant(angstrom)

//Calculation
d=n*lamda/(2*sin(theta))        //space of reflecting plane(angstrom)
//here the value of d comes to 0.422 angstrom which
```

```
is not equal to the value of a. hence the problem cannot be solved further

//Result
printf("\n space of reflecting plane is %0.2f angstrom",d)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 6.16 energy of X ray beam

```
clear
//
//
//

//Variable declaration
a=5.63*10**-10              //lattice constant(m)
h=1
k=1
l=1                         //miller indices
theta=27.5*%pi/180        //glancing angle(radian)
n=1                         //order
h=6.625*10**-34             //planck's constant
c=3*10**10                  //velocity of light(m/sec)
e=1.6*10**-19               //charge of electron(c)

//Calculation
d111=a/sqrt(h**2+k**2+l**2)
lamda=2*d111*sin(theta)/n       //wavelength of X-ray beam(m)
lamda=int(lamda*10**10)               //wavelength of X-ray beam(angstrom)
E=h*c/(lamda*10**-10*e)           //energy of X-ray beam(eV)

```

```
//Result
printf("\n wavelength of X-ray beam is %0.3f angstrom",lamda)
printf("\n energy of X-ray beam is %0.2f  *10**5 eV",E/10**5)
printf("\n answer for energy given in the book is wrong")
```

Scilab code Exa 6.17 `lattice constant`

```
clear
//
//
//

//Variable declaration
h=2
k=0
l=2                      //miller indices
theta=34*%pi/180     //glancing angle(radian)
n=1                      //order
lamda=1.5*10**-10        //wavelength of X-ray beam(m)

//Calculation
d=n*lamda/(2*sin(theta))
a=d*sqrt(h**2+k**2+l**2)           //lattice constant(m)

//Result
printf("\n lattice constant is %0.4f  angstrom",a*10**10)
printf("\n answer given in the book varies due to rounding off errors")
```

Scilab code Exa 6.20 `interplanar spacing`

```
clear
//
//
//

//Variable declaration
h=2
k=2
l=0                         //miller indices
a=450                       //length(nm)

//Calculation
d220=a/sqrt(h**2+k**2+l**2)            //interplanar
    spacing(nm)

//Result
printf("\n interplanar spacing is %0.1f  nm",d220)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 6.21 `interplanar spacing`

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=1                         //miller indices
```

```
r=1.278*10**-10             //radius(m)

//Calculation
a=4*r/sqrt(2)
d111=a/sqrt(h**2+k**2+l**2)             //interplanar
   spacing(m)

//Result
printf("\n interplanar spacing is %0.2f  *10**-10 m"
   ,d111*10**10)
```

Scilab code Exa 6.22 energy of X rays

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=1                         //miller indices
n=4
A=107.87                    //atomic weight(amu)
N=10500*6.052*10**26        //density(kg/m**3)
theta=(19+(12/60))*%pi/180        //angle(radian)
r=1.278*10**-10             //radius(m)
hp=6.625*10**-34            //plancks constant(Js)
c=3*10**8                   //velocity of light(m/sec)
e=1.6*10**-19               //charge of electron(coulomb
   )

//Calculation
a=(n*A/N)**(1/3)            //lattice constant(m)
d=a/sqrt(h**2+k**2+l**2)             //interplanar
```

```
   spacing(m)
lamda=2*d*sin(theta)        //wavelength of X-rays(m)
E=hp*c/(e*lamda)                     //energy of X-
   rays(eV)

//Result
printf("\n wavelength of X-rays is %0.3f  angstrom",
   lamda*10**10)
printf("\n answer in the book varies due to rounding
    off errors")
printf("\n energy of X-rays is %0.0f  *10**3 eV",E
   /10**3)
```

Scilab code Exa 6.23 glancing angle

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=0                         //miller indices
d100=0.28                   //lattice constant(nm)
n=2
lamda=0.071                 //wavelength(nm)

//Calculation
d110=d100/sqrt(h**2+k**2+l**2)            //
   interplanar spacing(m)
theta=asin(n*lamda/(2*d110))*180/%pi     //glancing
   angle(degrees)

//Result
```

```
printf("\n glancing angle is %0.0f  degrees",theta)
```

Scilab code Exa 6.24 distance between the planes

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=0                 //miller indices
a=0.38              //lattice constant(nm)

//Calculation
d=a/sqrt(h**2+k**2+l**2)      //distance between the
   planes(nm)

//Result
printf("\n distance between the planes is %0.2f  nm"
   ,d)
```

Scilab code Exa 6.26 intercept along y axis

```
clear
//
//
//

//Variable declaration
h=2
k=3
```

```
l=1                 //miller indices
a=0.121
b=0.184
c=0.197             //parameters(nm)

//Calculation
OB=2*b/3          //intercept along y axis(nm)
OC=2*c            //intercept along z axis(nm)

//Result
printf("\n intercept along y axis is %0.3f  nm",OB)
printf("\n intercept along y axis is %0.3f  nm",OC)
```

Scilab code Exa 6.27 `interplanar distance between planes`

```
clear
//
//
//

//Variable declaration
h1=1
k1=2
l1=3                //miller indices
h2=2
k2=4
l2=6                //miller indices
a=0.82
b=0.94
c=0.75              //parameters(nm)

//Calculation
d123=(((h1/a)**2)+((k1/b)**2)+((l1/c)**2))**(-1/2)
       //interplanar distance between (123) planes
d246=d123/2                   //interplanar distance
```

```
    between (246) planes

//Result
printf("\n interplanar distance between (123) planes
     is %0.3f  nm",d123)
printf("\n interplanar distance between (246) planes
     is %0.4f  nm",d246)
printf("\n answers given in the book are wrong")
```

Scilab code Exa 6.28 wavelength of Xrays

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=1                  //miller indices
a=0.2                //lattice parameter(nm)
theta=(87/2)*%pi/180        //angle(radian)

//Calculation
d=a/sqrt(h**2+k**2+l**2)
lamda=2*d*sin(theta)       //wavelength of Xrays(nm)

//Result
printf("\n wavelength of Xrays is %0.3f  nm",lamda)
```

Scilab code Exa 6.29 effective temperature of neutrons

```
clear
```

```
//
//
//

//Variable declaration
h=1
k=1
l=1                 //miller indices
a=0.352               //lattice parameter(nm)
theta=(28+(30/60))*%pi/180        //angle(radian)
hp=6.626*10**-34         //plancks constant(Js)
m=1.67*10**-27          //mass of proton(kg)
kB=1.38*10**-23         //boltzmann constant

//Calculation
d=a/sqrt(h**2+k**2+l**2)
lamda=2*d*sin(theta)        //wavelength(nm)
T=(hp**2)/(3*m*kB*(lamda*10**-9)**2)            //effective temperature of neutrons(K)

//Result
printf("\n effective temperature of neutrons is %0.0f K",T)
```

Scilab code Exa 6.30 `lattice parameter of sample A`

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=1                 //miller indices
```

```
lamda=0.152                 //wavelength(nm)
D=0.2552                    //diameter(nm)
theta1=21*%pi/180            //angle(radian)
theta2=(21+(23/60))*%pi/180          //angle(radian)

//Calculation
a=D*sqrt(2)                 //lattice parameter for regular crystal(nm)
d111_1=lamda/(2*sin(theta1))
alpha1=d111_1*sqrt(h**2+k**2+l**2)     //lattice parameter for sample A(nm)
d111_2=lamda/(2*sin(theta2))
alpha2=d111_2*sqrt(h**2+k**2+l**2)     //lattice parameter for sample B(nm)

//Result
printf("\n lattice parameter for regular crystal is %0.4f  nm",a)
printf("\n lattice parameter for sample A is %0.4f nm",alpha1)
printf("\n lattice parameter for sample B is %0.3f nm",alpha2)
printf("\n lattice parameter of sample A is 1.75 percent greater than that of pure copper")
```

Chapter 7

x ray diffraction and defects in crystals

Scilab code Exa 7.1 cube edge of unit cell

```
clear
//
//
//

//Variable declaration
n=1      //order
theta=32*%pi/180      //glancing angle(radian)
lamda=1.54                  //wavelength(angstrom)
h=2
k=2
l=0

//Calculation
d=n*lamda/(2*sin(theta))          //lattice parameter(
   angstrom)
a=d*sqrt(h**2+k**2+l**2)          //cube edge of unit
   cell(angstrom)

```

```
//Result
printf("\n cube edge of unit cell is %0.1f  angstrom
   ",a)
```

Scilab code Exa 7.2 interplanar spacing at 3rd angle

```
clear
//
//
//

//Variable declaration
lamda=0.58          //wavelength(angstrom)
theta1=6.45*%pi/180      //glancing angle(radian)
theta2=9.15*%pi/180      //glancing angle(radian)
theta3=13*%pi/180      //glancing angle(radian)
n=1                  //order

//Calculation
d1=lamda/(2*sin(theta1))      //interplanar spacing
   at 1st angle(angstrom)
d2=lamda/(2*sin(theta2))      //interplanar spacing
   at 2nd angle(angstrom)
d3=lamda/(2*sin(theta3))      //interplanar spacing
   at 3rd angle(angstrom)

//Result
printf("\n interplanar spacing at 1st angle is %0.3f
     angstrom",d1)
printf("\n interplanar spacing at 2nd angle is %0.3f
     angstrom",d2)
printf("\n interplanar spacing at 3rd angle is %0.3f
     angstrom",d3)
printf("\n answers given in the book are wrong")
```

Scilab code Exa 7.3 order of diffraction

```
clear
//
//
//

//Variable declaration
d=1.181          //lattice spacing(angstrom)
theta=90*%pi/180      //glancing angle(radian)
lamda=1.540      //wavelength of X-rays(angstrom)

//Calculation
n=2*d*sin(theta)/lamda     //order of diffraction

//Result
printf("\n order of diffraction is %0.3f  ",n)
```

Scilab code Exa 7.4 lattice parameter

```
clear
//
//
//

//Variable declaration
theta=9.5*%pi/180      //glancing angle(radian)
lamda=0.58       //wavelength of X-rays(angstrom)
n=1              //order
h=2
k=0
l=0
```

```
//Calculation
d=n*lamda/(2*sin(theta))
a=d*sqrt(h**2+k**2+l**2)     //lattice parameter(angstrom)

//Result
printf("\n lattice parameter is %0.3f  angstrom",a)
printf("\n answer given in the book varies due to rounding off errors")
```

Scilab code Exa 7.5 glancing angle

```
clear
//
//
//

//Variable declaration
theta1=(8+(35/60))*%pi/180      //glancing angle(radian)
lamda=0.842          //wavelength of X-rays(angstrom)
n1=1                 //order
n2=3                 //order

//Calculation
x=n2*lamda*sin(theta1)/(n1*lamda)
theta2=asin(x)*180/%pi            //glancing angle
theta2d=int(theta2)                       //glancing angle(degrees)
theta2m=(theta2-theta2d)*60               //glancing angle(minutes)

//Result
printf("\n glancing angle is %0.3f degrees %0.3f
```

```
   minutes",theta2d,theta2m)
printf("\n answer for glancing angle in minutes
   given in the book is wrong")
```

Scilab code Exa 7.6 value of miller indices

```
clear
//
//
//

//Variable declaration
a=3.16          //lattice parameter(angstrom)
theta=20.3*%pi/180      //glancing angle(radian)
lamda=1.54          //wavelength of X-rays(angstrom)
n=1                 //order

//Calculation
d=n*lamda/(2*sin(theta))            //interplanar
   spacing(angstrom)
x=(a/d)**2

//Result
printf("\n interplanar spacing is %0.2f  angstrom",d
   )
printf("\n answer for interplanar spacing given in
   the book is wrong")
printf("\n value of h**2+k**2+l**2 is %0.0f  . hence
    the miller indices could be (110) (011) or (101)
   ",x)
```

Scilab code Exa 7.7 maximum order of diffraction

```
clear
//
//
//

//Variable declaration
d=0.282              //lattice spacing(nm)
theta=(8+(35/60))*%pi/180      //glancing angle(
    radian)
n=1                  //order

//Calculation
lamda=2*d*sin(theta)/n         //wavelength(nm)
N=2*d/lamda                          //maximum order
    of diffraction

//Result
printf("\n wavelength is %0.3f  nm",lamda)
printf("\n maximum order of diffraction is %0.3f  ",
    N)
```

Scilab code Exa 7.8 maximum order of diffraction

```
clear
//
//
//

//Variable declaration
lamda=1.5            //wavelength(AU)
d=1.6                //lattice spacing(AU)

//Calculation
n=2*d/lamda                          //maximum order
    of diffraction
```

```

//Result
printf("\n maximum order of diffraction is %0.3f ",n)
```

Scilab code Exa 7.9 interatomic spacing

```
clear
//
//
//

//Variable declaration
theta=30*%pi/180      //glancing angle(radian)
h=1
k=1
l=1
lamda=1.5418               //wavelength(angstrom)
n=1                        //order

//Calculation
d=n*lamda/(2*sin(theta))       //interplanar spacing(angstrom)
a=d*sqrt((h**2)+(k**2)+(l**2))      //interatomic spacing(angstrom)

//Result
printf("\n interatomic spacing is %0.2f angstrom",a)
```

Scilab code Exa 7.10 glancing angle

```
clear
```

```
//
//
//

//Variable declaration
h=1
k=1
l=0
lamda=0.065                    //wavelength(nm)
n=2                            //order
a=0.26                         //axial length(nm)

//Calculation
x=n*lamda*sqrt(h**2+k**2+l**2)/(2*a)
theta=asin(x)*180/%pi          //glancing angle(degrees)

//Result
printf("\n glancing angle is %0.1f  degrees",theta)
```

Scilab code Exa 7.11 cube edge of unit cell

```
clear
//
//
//

//Variable declaration
h=1
k=1
l=1
lamda=1.54              //wavelength(angstrom)
n=1                                 //order
theta=19.2*%pi/180      //glancing angle(radian)

```

```
//Calculation
d=n*lamda/(2*sin(theta))
a=d*sqrt(h**2+k**2+l**2)      //cube edge of unit cell(angstrom)

//Result
printf("\n cube edge of unit cell is %0.3f  angstrom",a)
```

Scilab code Exa 7.12 `cube edge of unit cell`

```
clear
//
//
//

//Variable declaration
lamda=1.54*10**-10            //wavelength(m)
n=1                           //order
theta=19.2*%pi/180      //glancing angle(radian)
h=1
k=1
l=1

//Calculation
d=n*lamda/(2*sin(theta))
a=d*sqrt(h**2+k**2+l**2)      //cube edge of unit cell(m)

//Result
printf("\n cube edge of unit cell is %0.3f  *10**-10 m",a*10**10)
```

Scilab code Exa 7.13 `interplanar spacing`

```
clear
//
//
//

//Variable declaration
lamda=0.12          //wavelength(nm)
n=2                         //order
theta=28*%pi/180        //glancing angle(radian)

//Calculation
d=n*lamda/(2*sin(theta))      //interplanar spacing(
    nm)

//Result
printf("\n interplanar spacing is %0.2f  nm",d)
```

Scilab code Exa 7.14 `interplanar spacing`

```
clear
//
//
//

//Variable declaration
lamda=97          //wavelength(pm)
n1=1              //order
n2=3              //order
theta1=23*%pi/180       //glancing angle(radian)
theta2=60*%pi/180       //glancing angle(radian)

//Calculation
lamda1=n2*lamda*sin(theta1)/(n1*sin(theta2))      //
```

```
wavelength(pm)
d=n2*lamda/(2*sin(theta2))                    //interplanar spacing(pm)

//Result
printf("\n wavelength is %0.1f  pm",lamda1)
printf("\n interplanar spacing is %0.0f  pm",d)
printf("\n answer for wavelength given in the book varies due to rounding off errors")
```

Scilab code Exa 7.15 `wavelength for n`

```
clear
//
//
//

//Variable declaration
theta=45*%pi/180        //glancing angle(radian)
d=275              //interplanar spacing(pm)
n1=3
n2=4

//Calculation
lamda=2*d*sin(theta)      //wavelength(pm)
lamda1=lamda/n1               //wavelength for n=3
lamda2=lamda/n2               //wavelength for n=4

//Result
printf("\n wavelength for n=3 is %0.0f pm and for n=4 is %0.2f pm",lamda1,lamda2)
printf("\n answer for wavelength for n=4 given in the book varies due to rounding off errors")
```

Scilab code Exa 7.17 atomic diameter

```
clear
//
//
//

//Variable declaration
theta1=(30+(0/60))*%pi/180          //glancing angle(radian)
theta2=(35+(17/60))*%pi/180         //glancing angle(radian)
lamda=0.171                               //wavelength(nm)
h1=1
k1=1
l1=0
h2=2
k2=0
l2=0
h3=1
k3=1
l3=1

//Calculation
d100=lamda/(2*sin(theta1))      //wavelength(nm)
d200=lamda/(2*sin(theta2))      //wavelength(nm)
a1=d100*sqrt(h1**2+k1**2+l1**2)
a2=d200*sqrt(h2**2+k2**2+l2**2)     //lattice parameter in case of bcc
a3=d100*sqrt(h3**2+k3**2+l3**2)
a4=d200*sqrt(h2**2+k2**2+l2**2)     //lattice parameter in case of bcc
d=a3/sqrt(2)                        //atomic diameter
```

```
(nm)

//Result
printf("\n lattice parameter in case of bcc are %0.3f nm and %0.3f nm which are not the same. hence the metal is not bcc",a1,a2)
printf("\n lattice parameter in case of fcc are %0.3f nm and %0.3f nm which are the same. hence the metal is fcc",a3,a4)
printf("\n atomic diameter is %0.5f  nm",d)
printf("\n answer for atomic diameter given in the book varies due to rounding off errors")
```

Scilab code Exa 7.18 maximum order of diffraction

```
clear
//
//
//

//Variable declaration
d=0.282*10**-9            //lattice spacing(m)
theta=(8+(35/60))*%pi/180          //glancing angle(radian)
maxtheta=90*%pi/180
n=1                                    //order

//Calculation
lamda=2*d*sin(theta)/n           //wavelength of x-rays(m)
N=2*d*sin(maxtheta)/lamda        //maximum order of diffraction

//Result
printf("\n wavelength of x-rays is %0.3f  angstrom",
```

```
   lamda*10**10)
printf("\n maximum order of diffraction is %0.3f  ",
   N)
printf("\n answer for wavelength of x-rays given in
   the book is wrong")
```

Scilab code Exa 7.19 glancing angle

```
clear
//
//
//

//Variable declaration
d=3.04*10**-10           //lattice spacing(m)
lamda=0.79*10**-10       //wavelength(m)
n=3                      //order

//Calculation
x=n*lamda/(2*d)
theta=asin(x)*180/%pi        //glancing angle(degrees)

//Result
printf("\n glancing angle is %0.3f  degrees",theta)
```