Scilab Textbook Companion for Engineering Physics by S. D. Jain and G. G. Sahasrabudhe[1]

Created by
Kopalle Naga Anjaneyulu
B.TECH
Computer Engineering
Anurag Group of Institutions-CVSR College of Engg.
College Teacher
G. Vishnu Murthy
Cross-Checked by
K. V. P. Pradeep

July 31, 2014

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

Book Description

Title: Engineering Physics

Author: S. D. Jain and G. G. Sahasrabudhe

Publisher: University Press, India

Edition: 1

Year: 2010

ISBN: 9788173716782

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

List of Figures

Chapter 1

Physics and Engineering

Scilab code Exa 1.1 Error Estimation

```
clear;
clc();
//Given:
l=9.3; // length in cm
b=8.5;// breadth in cm
h=5.4;// height in cm
V= l*b*h; // Volume in cm^3
delta_l = 0.1; delta_b = 0.1; delta_h = 0.1; //
    scale has a least count = 0.1 cm
// absolute error
delta_V = (b*h*delta_l + l*h*delta_b +l*b*delta_h);
    // in cm^3
//relative error
re = delta_V/V;
p= re*100; // Evaluating percentage error
printf("Percentage Error = %d percentage.",p);
```

Scilab code Exa 1.2 Error Estimation

```
clear;
clc();
//Given :
M= 10.0; //weight in g
V= 5.80;//volume in cm^3
Rho = M/V; // Density in g/cm^3
delta_M= 0.2 //  apparatus has a least count of 0.2
    g
delta_V= 0.05// apparatus has a least count of 0.05
    cm^3
delta_Rho = (delta_M/V) +((M*delta_V)/V^2);//
    absolute error in g/cm^3
re = delta_Rho/Rho ; //Evaluating Relative Error
p = re*100;// Evaluating Percentage Error
printf("Percentage error = %.1f percentage.",p);
//Result obtained differs from that in textbook,
    because delta_M walue is taken 0.1 g , instead of
     0.2 g as mentioned in the problem statement.
```

Scilab code Exa 1.3 Refinement in experiment

```
clc();
clear;
//Given:
//(a)
lc = 0.1// least count in cm
c = 6.9 //Circumference c in cm
r= 1.1 // radius of circle in cm
val =2*%pi;
// Circumference,c= 2*pi*r or  c/r = 2*pi
// Error in c/r is , delta(c/r)= [(c/r^2)+(1/r)](LC
```

```
/2) , LC is Least Count .
E= ((c/r^2)+(1/r))*(lc/2);//Error in c/r is delta(c/
   r)
ob = c/r; // Observed Value
//Actual Value of c/r ranges between
ac1 = ob-E;// Evaluating Minimum value for c/r
ac2 = ob+E;// Evaluating Maximum value for c/r
p = (E/ob)*100; //Evaluating percentage error
printf("(a)Actual Value of c/r ranges between %.2f -
    %.2f and Percentage error = %.2f percentage.\n
   ",ac1,ac2,p);
//(b)
lc1 = 0.001;//Now the least count is 0.001 cm
c1 = 6.316;//Circumference in cm
r1=1.005;//Circle radius in cm
E1 =((c1/r1^2) + (1/r1))*(lc1/2); // Error in c/r is
    delta(c/r)
ob1= c1/r1; //Observed Value
p1=(E1/ob1)*100;//Evaluating percentage error
//Actual Value of c/r ranges between
a1= ob1-E1;//Evaluating Minimum value for c/r
a2= ob1+E1;//Evaluating Maximum value for c/r
printf("(b)Actual Value of c/r ranges between %.3f -
    %.3f and Percentage error = %.2f percentage.\n",
   a1,a2,p1);
```

Scilab code Exa 1.4 Refinement in theory

```
clc();
clear;
//Given
// (a) Newton's Theory
// v= (P/rho)^2 , P= Pressure , rho = density
```

```
P = 76; // 76 cm of Hg pressure
V= 330 ; // velocity of sound in m/s
rho = 0.001293; // density for dry air at 0 degrees celsius in g/cm^3
g = 980;//gravitational acceleration in cm/s^2
//Density of mercury at room temperature is 13.6 g/cm^3
// 1 cm^2 = 1.0*10^-4 m^2
v = sqrt(((P*13.6*g)/rho)*10^-4); // velocity of sound in m/s
p= ((V-v)/V)*100; // % lower than the experimental value
printf("(a) It is is %d percentage lower than the experimental value.\n\n",p);

// (b) Laplace's Theory
// v= ((gama*P)/rho)^2., gamma = adiabatic index Thus,
//Given :
gama = 1.41 // Adiabatic index
//Density of mercury at room temperature is 13.6 g/cm^3
// 1 cm^2 = 1.0*10^-4 m^2
v1 = sqrt(((gama*P*13.6*g)/rho)*10^-4);// velocity of sound in m/s
p1 = ((V-round(v1))/V)*100;// % higher than the eperimental value
printf("(b) It is %.1f percantage higher than the experimental value.\n",abs(p1));
```

Chapter 2

What is Light

Scilab code Exa 2.2 Space and Time profile

```
clc();
clear;
// wave y= 2*sin(10*pi*t - (pi*x)/40  + pi/4)
// (a) Plot the space profile at t= T/4
// Comapring the given Equation with  y= A*sin(omega
   *t - k*x + phi)
omega = 10*%pi ; //Angular frequency in rad/s
k= %pi/40 ; // Wave number in rad/m
T= 1/5 ; // 2*pi/T = 10*pi , so Time period is 1/5 s
lambda = 80; // Wavelength in m ,  2*pi/lambda = pi
   /40 , so lambda = 80
t1= T/4; //time period in s
x1= 0;//  in m
printf("The Space profile of a wave y= 2*sin(10*pi*t
    - (pi*x)/40 + pi/4) when  t= T/4\n\n")
printf("\tx (in m) \t y1(x)  (in m)\n");
while x1<180
y1= 2*sin((omega*t1)-(k*x1)+ (%pi/4));
printf("\t%d\t\t%.3f\n",x1,y1);
x1 = x1+10;
end
```

```
//Now, we will plot the space profile from the values obtained for y1 for each value of x1
x_1 = [0,10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170];
y_1 = [1.414214,2.000000,1.414214,0.000000,-1.414214,-2.000000,-1.41421
// axis centered at (0,0)
axis=gca(); // Handle on axes entity
axis.x_location = "origin";
axis.y_location = "origin";
plot(x_1,y_1,style=5);
xtitle("Space Profile at t = T/4 for the wave     y= 2*sin(10*pi*t - (pi*x)/40  + pi/4)","x (in m)","y1(x)  (in m)");
xpause(10000000);
//(b)
x2= lambda/8; //in m
t2=0; // time period in s
printf("The time profile of a wave y= 2*sin(10*pi*t - (pi*x)/40 + pi/4) when  x= lambda/8\n\n")
printf("\t t(in s)      \t      y2(t) (in m)\n\n");
while t2<0.4
 y2=2*sin((omega*t2)-(k*x2)+ (%pi/4));
 printf("\t%.3f\t\t%.3f\n",t2,y2);
 t2=t2+0.025;
end
//Now,we will plot the time profile from the values obtained for y2 ,for each value of t2
x_2 =[0,0.025,0.05,0.075,0.1,0.125,0.15,0.175,0.2,0.22500,0.250000,0.
y_2 =[0.000000,1.414214,2.000000,1.414214,0.000000,-1.414214,-2.000000
// axis centered at (0,0)
axis1=gca(); // Handle on axes entity
```

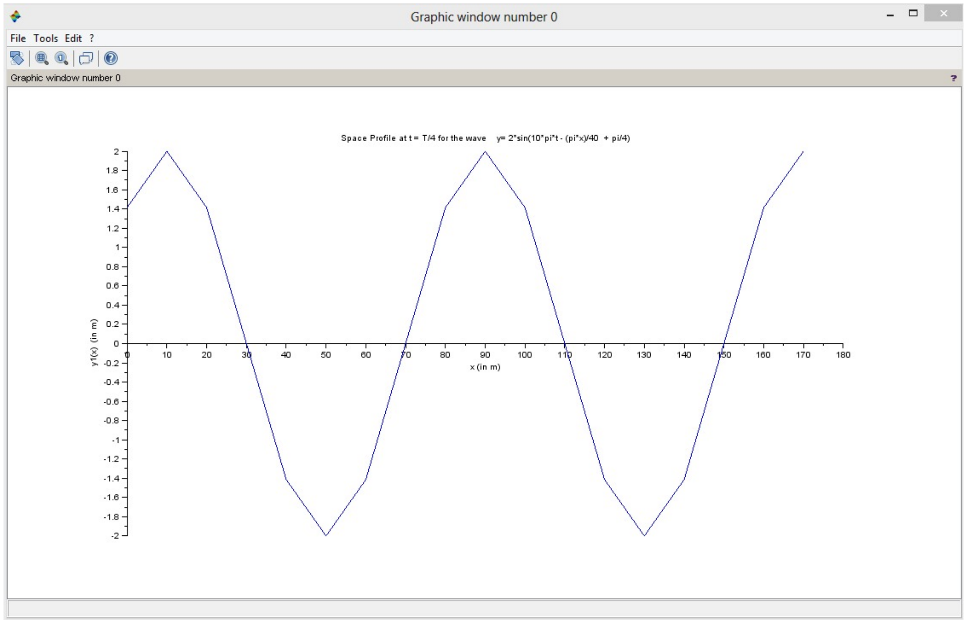

Figure 2.1: Space and Time profile

```
axis1.x_location = "origin";
axis1.y_location = "origin";
plot(x_2,y_2,style= 4);
xtitle("Time Profile at x = lambda/8 for the wave
      y= 2*sin(10*pi*t - (pi*x)/40  + pi/4)","t (in
   s)","y2(t)  (in m)");
```

Scilab code Exa 2.3 Wave Parameters

```
clc();
clear;
```

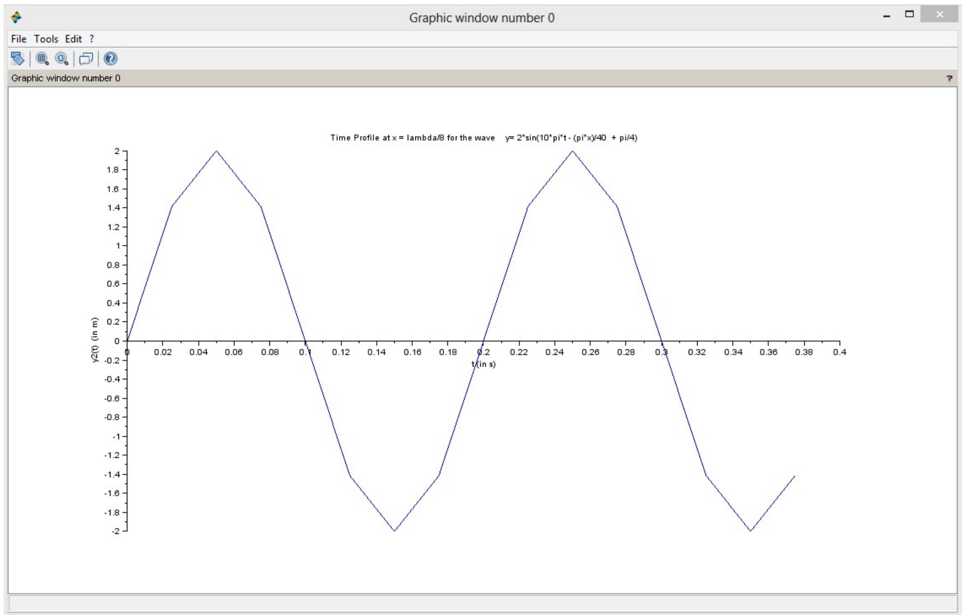

Figure 2.2: Space and Time profile

```
//Let us consider, wave function  y = A*sin(omega*t
    - K*x + phi)
A= 0.02;// Amplitude in m
lambda = 6; // Wavelength (lambda) = Crest Distance
    = 6 m
T= 2;// Time period is s
nu =  1/T; // Frequency in Hz
omega = 2*%pi*nu ; //Angular Frequency in rad/s
k = 2*%pi/lambda; //wave number in rad/m
//from Space profile , when x=1.5 m, t= 0
y = 0.02; //in m
x=1.5;//in m
t= 0; // in s
phi = (asin(y/A) +(k*x) - (omega*t)); // Initial
    phase in radians
printf(" Wave parameters from the space profile and
    time profile\n")
printf(" (1)Amplitude is %.2f m \n (2)Wavelength  is
```

```
%d m \n (3)Time period is %d s \n (4)Frequency
is %.1f Hz \n (5)Angular Frequency is %.3f rad/s\
n (6)Wave number is %.3f rad/m \n (7)Initial
phase is %.3f radians\n",A,lambda,T,nu,omega,k,
phi);
// y(x,t=0) : -0.02 = 0.02*sin(0-(pi*x)/3 + pi)
//Thus (-pi*x)/3 + pi = -pi/2,-5*pi/2, giving x=
9/2 m,21/2m
V= omega/k; // Velocity of wave in m/s
// I is proportional to A^2
I = A^2; // Intensity in m^2 (Proportional)
printf(" (8)The velocity of wave is %d m/s \n (9)
Intensity is proportional to : %.1f x 10^-4 m^2."
,V,I*10^4);
```

Scilab code Exa 2.6 Sound and Light waves

```
clc();
clear;
//(a)Tunning fork
nu= 440; // Frequency in Hz
V=340; // velocity of sound in air in m/s
lambda= V/nu ;// Wavelength of sound wave in m
k= 2*%pi/lambda; // Wave number in m
//(b) Red Light
nu1 = 5*10^14;// Frequency of Red light in Hz
V1 = 3*10^8;//Velocity of light in m/s
lambda1= V1/nu1; //Wavelength of light wave in m
k1= 2*%pi/lambda1; // Wave number in m
printf("For Sound wave : \n\n Frequency: %d Hz \n
Velocity: %d m/s \n Wavelegth: %.3f m\n Wave
number : %.2f m \n Wave Equation for Sound wave:
y = A*sin((%.2f*x)-(%.3f*t)) \n\n",nu,V,lambda,k
```

```
,k,(2*%pi*nu));
printf("For Light wave : \n\n Frequency: %.0f x
10^14 Hz \n Velocity: %d x 10^8 m/s \n Wavelegth:
 %.1f x 10^-7 m\n Wave number : %.2f x 10^7 m \n
Wave Equation for Sound wave: y =  A*sin((%.2f
*10^7*x)-(%.1f*10^15*t)) \n\n",nu1*10^-14,V1
*10^-8,lambda1*10^7,k1*10^-7,k1*10^-7,(2*%pi*nu1
*10^-15));
```

Chapter 3

Interference

Scilab code Exa 3.3 Incoherence

```
clc();
clear;
//Given :
lambda1 = 5890 ; // Wavelength in angstroms
lambda2 = 5896 ; // Wavelength in angstroms
//For sodium doublet
nu1 = 5.0934*10^14; //Frequency in Hz
nu2 = 5.0882*10^14; //Frequency in Hz

deltanu = nu1-nu2; // Differnece in Frequencies in Hz
Tc = 1/deltanu ; // Coherence time in s

n1 = Tc*nu1; // Number of Cycles of wavelength 5890 angstroms
n2 = Tc*nu2;// Number of cycles of wavelegth 5896 angstrom
//in this coherence time , we have:
printf("Cycles : %d , Wavelength %d A \n",round(n1), lambda1);
printf("Cycles : %d , Wavelength %d A",round(n2),
```

```
lambda2);
```

Scilab code Exa 3.4 Ordinary and Laser source

```
clc();
clear;
//Given:
deltalambda1 = 0.01; // The line width of the orange line of krypton,Kr^86  in A
lambda = 6058; // Wavelength in angstroms = 6058*10^-10 m
deltalambda2 = 0.00015; // The line width of a laser source in A
c = 3*10^8 ;// Velocity of light  in vacuum in m/s
nu0 = c/(lambda*10^-10);// lambda in m , 1 A = 1.0   10^-10 m
//1 A = 1.0    10^-10 m
//For orange line of Krypton
Lc1= (lambda^2/deltalambda1)*10^-10; // coherence length in m
deltanu1 = c/Lc1 ;//  bandwidth in Hz
Tc1 = (Lc1/c);// Coherence time in s
//Xi = deltanu/nu0 , where nu0 = c/lambda which equals to (deltanu*lambda)/c, lambda in A
Xi1 = deltanu1/nu0 ; //degree of monochromaticity
//For Laser Source
Lc2= (lambda^2/deltalambda2)*10^-10;// coherence length in m
deltanu2 = c/Lc2 ;//  in Hz
Tc2 = (Lc2/c);//Calculating Coherence time in s
//Xi = deltanu/nu0 , where nu0 = c/lambda which equals to (deltanu*lambda)/c, lambda in A
Xi2 = deltanu2/nu0 ;// degree of monochromaticity
```

```
printf("For Orange line of Krypton : \n\n Coherence Length : \t %.4f m \n Bandwidth : \t\t %.2f x 10^8 Hz \n Coherence : \t\t %.2f x 10^-8 s \n Degree of Monochromaticity :  %.2f x 10^-6  \n\n",Lc1,deltanu1*10^-8,Tc1*10^8,Xi1*10^6);
printf("For Laser Source : \n\n Coherence Length : \t %.2f m \n Bandwidth : \t\t %.2f x 10^7 Hz \n Coherence : \t\t %.2f x 10^-8 s \n Degree of Monochromaticity :  %.2f x 10^-8  \n\n",Lc2,deltanu2*10^-7,Tc2*10^8,Xi2*10^8);
```

Scilab code Exa 3.5 Optical path

```
clc();
clear;
//(a)
//Given:
lambda = 5890;// Wavelength in A
l = 5.89; //thickness of the film in mu m
mu = 1.35; //refractive index
delta = mu*l;// optical path in the medium  in m
//(b) (i)Number of waves in the medium
//1 angstrom = 1.0*10^-10 m and 1 mu m = 1*10^-6 m
N= (l*10^-6)/(lambda*10^-10/mu);
//the distance in vaccum for those waves :
delta1 =N*lambda*10^-10; // optical path in m
//(b) (ii)Phase difference in the medium
//1 angstrom = 1.0*10^-10 m and 1 mu m = 1*10^-6 m
phi = ((2*%pi)/(lambda*10^-10/mu))*(l*10^-6) ;
printf("Optical path = %.4f mu m\n",delta);
printf("Number of waves : %.1f\n",N);
printf("The distance in vaccum for those waves is :  %.4f mu m  \n",delta1*10^6);
```

```
printf("Phase difference = %.3f\n",phi);
```

Scilab code Exa 3.6 Total optical path

```
clc();
clear;
//Given:
lambda = 5890; // Wavelength of a beam of sodium
    light in A
l = 100 ; // thickness in cm
mu1 = 1.00;//refractive index of air
mu2 = 1.33;// refractive index of water
mu3 = 1.39; // refractive index of oil
mu4 = 1.64; // refractive index of glass
c = 3*10^8 ;// Velocity of light in vacuum in m/s
//For Air :
lambda1 = lambda/mu1; // wavelength of light in A
v1 = c/mu1;// Velocity of light in air in m/s
// 1cm = 1*10^-2 m
t1 = (l*10^-2/v1); //time of travel in s
// 1 A = 1*10^-10 m
N1 = (l*10^-2)/(lambda1*10^-10);// Number of waves
delta1 = mu1*l; //Optical path in cm
//For Water :
lambda2 = lambda/mu2; // wavelength of light in A
v2 = c/mu2;// Velocity of light in water in m/s
//1cm = 1*10^-2 m
t2 = (l*10^-2/v2); //time of travel in s
//1 A = 1*10^-10 m
N2 = (l*10^-2)/(lambda2*10^-10);// Number of waves
delta2 = mu2*l; //Optical path in cm
//For Oil :
lambda3 = lambda/mu3; // wavelength of light in A
```

```
v3 = c/mu3;// Velocity of light in Oil in m/s
//1cm = 1*10^-2 m
t3 = (l*10^-2/v3); //time of travel in s
//1 A = 1*10^-10 m
N3 = (l*10^-2)/(lambda3*10^-10);// Number of waves
delta3 = mu3*l; //Optical path in cm
//For Glass:
lambda4 = lambda/mu4; // wavelength of light in A
v4 = c/mu4;// Velocity of light in Glass in m/s
// 1cm = 1*10^-2 m
t4 = (l*10^-2/v4); //time of travel in s
//1 A = 1*10^-10 m
N4 = (l*10^-2)/(lambda4*10^-10);// Number of waves
delta4 = mu4*l; //Optical path in cm
delta = delta1+delta2+delta3+delta4; // total optical path in cm
printf("Parameters \t\t\t Air \t\t\t Water \t\t\t Oil \t\t\tGlass \n\n");
printf("Wavelength : \t\t %.0f A \t\t %.1f A \t\t %.1f A \t\t %.1f A \n",lambda1,lambda2,lambda3,lambda4);
printf("Velocity : \t\t %.0f x 10^8 m/s \t\t %.2f x 10^8m/s \t %.2f x 10^8 m/s \t %.2f x 10^8 m/s \n",v1*10^-8,v2*10^-8,v3*10^-8,v4*10^-8);
printf("Time of travel : \t %2.1f x 10^-10 s\t %2.1f x 10^-10 s\t %2.1f x 10^-10 s\t %2.1f x 10^-10 s \n",t1*10^10,t2*10^10,t3*10^10,t4*10^10);
printf("Number of waves: \t %.1f x 10^6 \t\t %.1f x 10^6 \t\t %.1f x 10^6 \t\t %.1f x10^6 \n",N1*10^-6,N2*10^-6,N3*10^-6,N4*10^-6);
printf("Optical path : \t\t %d cm \t\t %d cm \t\t %d cm \t\t %d cm \n\n",delta1,delta2,delta3,delta4);
printf(" The total optical path = %d cm\n\n",delta);
```

Scilab code Exa 3.8 Maximum observable fringes

```
clc();
clear;
//Given :
lambda = 6058;// Wavelength of light in A
deltalambda1 = 0.01; // line width for a krypton
    source in A
deltalambda2 = 0.00015; // line width for a laser
    source in A
// The maximum number of fringes is given by n_max =
    lambda/deltalambda
// (a) For a krypton source :
n_max1 = lambda/deltalambda1 ;
// (b) For a laser source :
n_max2 = lambda/deltalambda2;
printf("The maximum number of fringes observable are
    :\n\n");
printf("(a) For a krypton source : %d \n\n",n_max1);
printf("(b) For a laser source : %d \n\n",n_max2);
```

Scilab code Exa 3.9 Interference pattern

```
clc();
clear;
//Given :
mu = 1.4;// refractive index of a thin film
lambda = 5890; // Wavelength of sodium light in A
```

```
deltalambda = 20; //line width in A
// For observing interference pattern , t < lambda^2/(2*mu*deltalambda)
t_max = lambda^2/(2*mu*deltalambda); //thickness of the film in A
printf(" t_max : %1.3f x 10^5 A \n\n",t_max*10^-5);
```

Scilab code Exa 3.10 Wedge angle

```
clc();
clear;
//Given:
lambda = 6000; // wavelength in A
mu = 1; //refractive index for air
// Fringe pattern having 100 fringes per cm
betaa = 0.01; // fringe width in cm
// And,We know betaa = lambda/(2*mu*alpha) , so
 // 1 A = 1.0*10^-8 cm
 alpha = lambda*10^-8/(2*mu*betaa); // wedge angle in rad
 printf("Wedge angle = %.3f rad",alpha);
```

Scilab code Exa 3.13 Interference for different waves

```
clc();
clear;
//Given :
angle = 4*10^-2 ; // angle in rad
//1 radian = 57.2957795 degrees
```

```
theta = angle*57.2957795 ;// in degrees
//  d*sin(theta) = lambda , so  d = lambda/(sin(
    theta)) :
//(a)For Sound waves
lambda1 = 0.75; // Wavelength in m
d1 = lambda1/sind(theta); // distance in m
//(b)For Ultrasonic waves
lambda2 = 0.1; // Wwavelength in m
d2 =  lambda2/sind(theta); // distance in m
//(c)For microwaves
lambda3 = 2.9 ; // Wavelength in cm
//1cm = 1.0*10^-2 m
d3 = lambda3*10^-2/sind(theta); // distance in m
//(d)For IR waves
lambda4 = 10; // Wavelength in mu_m
// 1 mu_m = 1.0*10^-6 m
d4 = lambda4*10^-6/sind(theta);// distance in m
//(e)For light waves
lambda5 = 5890;// in angstroms
//1 A = 1.0*10^-10 m
d5 = lambda5*10^-10/sind(theta); // distance in m
printf(" (a)For Sound waves : %.2f m \n",d1);
printf(" (b)For Ultrasonic waves : %.2f m \n",d2);
printf(" (c)For Microwaves  : %.2f m \n",d3);
printf(" (d)For IR waves : %.1f mu m \n",d4*10^6);
printf(" (e)For Light waves : %.2f mu m \n",d5*10^6)
    ;
```

Scilab code Exa 3.14 Intensity distribution

```
clc();
clear;
//Given :
```

```
// Now, the intensity  distribution  is given by :
// I = I_1 + I_2 + 2*(I_1*I_2)^0.5 *cos(alpha1-alpha2) , Using alpha = alpha1 - alpha2  and I_1 = I_2 = I_0
// I = 2*I_0*(1+ cos(alpha))
nu = 1.2 * 10^6 ; // frequency in Hz
c = 3*10^8 ; // velocity of light in m/s
lambda = c/nu ; // wavelength in m
d = 500; // two identical vertical dipole antenna spaced 500 m apart
// Directions along which  the intensity is maximum :
printf("Maximum Intensity \n\n");
for n= 0 :2
theta = asind((n*lambda)/d);// in degrees
printf("---> theta = %d degrees\n",theta);
end
// Directions for which intensity is minimum :
n1 =0;
theta1 = asind(((n1 + (1/2))*lambda)/d);//in degrees
printf("Minimum Intensity \n\n");
printf("--> theta = %.1f degrees\n",theta1);
```

Scilab code Exa 3.15 Linear expansivity

```
clc();
clear;
//Given :
lambda = 5900 ; //Wavelength in A
delta_T = 150; // Temperature of the metal cylinder is now raised by 150 K
p = 20 ; // p is the number of rings shifted due to increase in t_n (t_n  is the thickness of the air
```

```
    film)
l = 5 ; // length of the metal cyclinder in  mm
mu = 1; //refractive index for air
//Increase in length  = (p*lambda)/2*mu
// 1 A = 1.0*10^-7 mm
delta_l = (p*lambda*10^-7)/2*mu; // increase in
    length in mm
//Linear expansivity of the metal of the cyclinder
alpha = (delta_l)/(l*delta_T); // in 1/K
printf("The linear expansivity of the metal of the
    cylinder using Newtons rings apparatus is : %.1f
    x 10^-6/K ", alpha*10^6);
```

Scilab code Exa 3.16 Michelsons interferometer

```
clc();
clear;
//Given :
d = 0.065; //distance in mm
p = 200 ;// 200 fringes cross the field of view
//Michelson's interferometer arrangement : 2*d = p*
    lambda
lambda = 2*d/p;// wavelength in mm

printf(" Wavelength : %.1f x 10^-4 mm ",lambda*10^4)
    ;
```

Scilab code Exa 3.17 Newtons ring apparatus

```
clc();
clear;
//Given :
D10_air = 1.75 ;//diameter of the 10th bright ring
    in Newton's ring apparatus in cm
D10_liquid = 1.59 ; // diameter of the 10th bright
    ring in Newton's ring apparatus in cm
// The diameter of the nth bright ring in Newton's
    ring apparatus :  D_n = 2*(R*(n + 1/2)*(lambda/mu
    ))^0.5
mu = (D10_air/D10_liquid)^2;
printf("The refractive index of the liquid is %.3f",
    mu);
```

Scilab code Exa 3.18 Anti Reflection Coating

```
clc();
clear;
//Given :
lambda = 5500; // Wavelength in A
mu_f = 1.38; // refractive index for MgF2
mu_f1 = 1.48; // refractive index for lucite
//The minimum thickness
t = lambda/(4*mu_f) ; // thickness in A
printf("The minimum thickness = %.1f A\n\n",t);
// Resultant reflected intensity = I = 2*I_0*(1 +
    cos(alpha))
// alpha = (2*pi/lambda)*(path difference)
alpha1 = (2*%pi/lambda)*(2*mu_f*t); // angle in
    radians
alpha2 = (2*%pi/lambda)*(2*mu_f1*t); // angle in
    radians
printf(" alpha = %.3f for MgF2 and %.3f for lucite\n
```

```
\n",alpha1,alpha2);
printf(" For MgF2 : I = (%f)*I_0\n\n",2*(1+cos(alpha1)));
printf(" For lucite : I = (%.3f)*I_0\n\n",2*(1+cos(alpha2)));
printf("For Lucite : (%.3f)*I_0 , indicates %.1f percentage of the incident light is reflected , so it is less suitable for coating.",2*(1+cos(alpha2)), 100*2*(1+cos(alpha2)));
```

Chapter 4

Diffraction

Scilab code Exa 4.4 Interference minima

```
clc();
clear;
//Given :
d = 8.8*10^-2 ; // slit width in mm
b = 0.7;// seperation between slits in mm
lambda = 6328 ; //Wavelength in A
//First diffraction minima is possible , when  d*sin(theta) = lambda
// 1 A = 1.0*10^-7 mm
theta = asind((lambda*10^-7)/d); // angle in degrees
printf("theta = %.3f degrees .\n\n",theta);
//interference minima is possible , when sin(theta) = ((p + 1/2)*lambda)/b
for p = 0 : 10
    //1 A = 1.0*10^-7 mm
    theta1 = asind((p + 1/2)*(lambda*10^-7/b)); // angle in degrees
    printf("When p = %d \n",p);
    printf("theta = %.3f degrees . \n\n",theta1);
 if(theta1 > theta)
     printf(" When p >= %d ,  theta > %.3f degrees
```

```
        .\n\nBetween the first two diffraction
        minima , %d interference minima are possible
        .",p,theta,2*p);
    break;
 end
end
```

Scilab code Exa 4.6 Angles of Diffraction

```
clc();
clear;
//Given :
// a+b = (2.54/N)cm
N = 15000;//grating has 15000 lines
a_plus_b = 2.54/N ; // grating element in cm
//Grating equation , (a+b)*sin(theta_n) = n*lambda ,
   we get : theta_n = asind((n*lamba)/(a+b))
printf("For line D1  and Wavelength 5890 A:\n\n");
printf(" Angles at which first order and second
   order maxima will be observed are :\n");
lambda1 = 5890; //Wavelength in A
for n = 1:2 // First and second order maxima
// 1 A = 1.0*10^-7 mm
theta1_n = asind((n*lambda1*10^-8)/a_plus_b);//
   angle in degrees
printf(" Order :%d ,%.3f degrees \n",n,theta1_n);
end
printf("For line D2 and Wavelength 5895.9 A :\n\n");
printf(" Angles at which first order and second
   order maxima will be observed are :\n");
lambda2 = 5895.9 ; //Wavelength in A
for n1 = 1:2 //First and second order maxima
// 1 A = 1.0*10^-7 mm
```

```
 theta2_n = asind((n1*lambda2*10^-8)/a_plus_b);//
    angle in degrees
 printf("Order : %d, %.3f degrees \n",n1,theta2_n);
end
printf(" When n = 3,  sin(theta)= ((n*lambda*10^-8)/
   a_plus_b)>1 , which falls outside the sine range,
    hence third order maximum is not visible");
```

Scilab code Exa 4.8 Dispersion and resolving power

```
clc();
clear;
// Given :
//(a) 15000 lines per inch
N1 = 15000; //15000 lines per inch
a1_plus_b1 = (2.54/N1)*10^8 ; //grating element in A
lambda1 = 5890; //Wavelength in A
lambda2 = 5895.9 ; // Wavelength in A
deltalambda1 = lambda2-lambda1; //in A
//For first order
n =1;
theta1 = 20.355; // in degrees
deltatheta1 = ((n*deltalambda1)/((a1_plus_b1)*cosd(
   theta1)));// dispersion in degrees/A
rp1 = n*N1; // resolving power


//(b)15000 lines per cm
// 1 cm = 0.393701 inches, so We have 15000 lines
   per 0.393701 inches.
//Therefore, For 1 inch we have 15000/0.393701 =
   38099.979 or 38100 lines
N2 = 38100 ; //38100 lines per inch
```

```
a2_plus_b2 = (2.54/N2)*10^8 ; //grating element in A
//For first order
theta_1 = asind((n*lambda1)/(a2_plus_b2));// in degrees
deltatheta_1 = ((n*deltalambda1)/((a2_plus_b2)*cosd(theta_1)));//  dispersion in degrees/A
rp2 = n*15000; // resolving power


//(c)5906 lines per cm
// 1 cm = 0.393701 inches, so We have 5906 lines per 0.393701 inches.
//Therefore, For 1 inch we have 5906/0.393701 = 15001.232  or 15001 lines
N3 = 15001; //15001 lines per inch
a3_plus_b3 = (2.54/N3)*10^8; //grating element in A
//For first order
theta__1 = asind((n*lambda1)/(a3_plus_b3)); // in degrees
deltatheta__1 = ((n*deltalambda1)/((a3_plus_b3)*cosd(theta__1))); //  dispersion in degrees/A
rp3 = n*5906; // resolving power

printf(" Number of lines \tGrating element (in A)\t Angle of diffraction(degrees)\t Dispersion (degrees/A) \t Resolving Power\n");
printf("%d /inch\t\t\t %.0f\t\t %.2f \t\t\t\t %.2f x 10^-3\t\t\t %d\n",N1,a1_plus_b1,theta1,deltatheta1*10^3,rp1);
printf("%d /cm\t\t\t %.0f\t\t %.2f \t\t\t\t %.2f x 10^-3\t\t\t %d\n",15000,a2_plus_b2,theta_1,deltatheta_1*10^3,rp2);
printf("%d /cm\t\t\t %.0f\t\t %.2f \t\t\t\t %.2f x 10^-3\t\t\t %d\n",5906,a3_plus_b3,theta__1,deltatheta__1*10^3,rp3);
// Error in textbook for dispersion values . Error in decimal point placement .
```

Scilab code Exa 4.9 Determination of separation of lines

```
clc();
clear;
//Given:
//Wavelength
n=1; // first order diffraction
lambda1 = 4680 ;// Wavelength in A
lambda2 = 4800; //Wavelength in A
lambda3 = 5770 ; // Wave;ength in A
// First order diffraction angle
theta1 = 28.0; // angle in degrees
theta2 = 28.7; // angle in degrees
theta3 = 35.5; //angle in degrees
//Grating equation : (a+b) = n*lambda/sin(theta)
a1_plus_b1 = (n*lambda1)/sind(theta1); //spacing in
    A
a2_plus_b2 = (n*lambda2)/sind(theta2); //spacing in
    A
a3_plus_b3 = (n*lambda3)/sind(theta3); //spacing  in
     A
mean_spacing = (a1_plus_b1 + a2_plus_b2 + a3_plus_b3
    )/3; // mean spacing in A
printf("(a)Wavelength :%d  A \n Angle of 1st order
    Diffraction : %.1f degrees \n Spacing = %.1f A\n\
    n",lambda1,theta1,a1_plus_b1);
printf("(b)Wavelength :%d  A \n Angle of 1st order
    Diffraction : %.1f degrees \n Spacing = %.1f A\n\
    n",lambda2,theta2,a2_plus_b2);
printf("(c)Wavelength :%d  A \n Angle of 1st order
    Diffraction : %.1f degrees \n Spacing = %.1f A\n\
    n",lambda3,theta3,a3_plus_b3);
```

```
printf("Mean Spacing = %.1f A",mean_spacing);
```

Scilab code Exa 4.10 Diffraction of Xrays

```
clc();
clear;
//Given:
N = 15000;//Number of lines per inch
a_plus_b = (2.54/N)*10^8 ;//Grating period in A
lambda = 1 ; //Wavelength in A
//Grating equation :(a+b)*sin(theta_n) = n*lambda
//First order maximum
theta1 = asind(lambda/a_plus_b); // angle in degrees
printf("The first order maximum will be obtained at
    : %.4f degrees .\n\n",theta1);
```

Scilab code Exa 4.11 Resolution of human eye

```
clc();
clear;
//Given:
lambda = 6000; //Wwavelength in A
mu = 1.33; //Refractive index for cornea
D = 2; //Diameter of pupil in mm
//Yellow light wavelength in eye:
lambda1 = lambda/mu ; //Wavelength in A
//The angular resolution
//1 A = 1.0*10^-7 mm
theta_c = (1.22*lambda1*10^-7)/D; // angle in rad
```

```
//Maximum value for L
L = 1/tan(theta_c); // in mm
printf("Maximum value for L should be : %.1f mm",L);
```

Chapter 5

Polarisation

Scilab code Exa 5.4 Brewster Law

```
clc();
clear;
//Given:
mu = 1.33; //Refractive index of water
//Brewster's angle, theta_p = atand(mu) ;
theta_p = atand(mu); // in degrees
theta_s = 90-theta_p ; // in degrees
printf("Angle = %.1f degrees",theta_s);
```

Scilab code Exa 5.5 Critical angle for TIR

```
clc();
clear;
//Given:
r = 90; // in degrees
mu_o= 1.658 ;// Refractive index for ordinary array
```

```
mu =1.55; // Refractive index for a canada balsam
    material
//Snell's Law,mu1*sin(i) = mu2*sin(r), we have :
i = asind((mu*sind(90))/mu_o); // angle in degrees
printf("Critical angle =  %d degrees",i);
```

Scilab code Exa 5.6 Minimum thickness of wave plate

```
clc();
clear;
//Given :
mu_o = 1.544; //Refractive index for ordinary ray
mu_e = 1.553;//Refractive index for extraordinary
    ray
lambda = 5890;//Wavelength in A
//(a)Plane polarised light :
//lambda is converted from A to cm , 1 A = 1.0*10^-8
    cm
t1 = (lambda*10^-8)/(2*(mu_e-mu_o));//Minimum
    thickness in cm
//(b)Circularly polarised light :
t2 = (lambda*10^-8)/(4*(mu_e-mu_o));// Minimum
    thickness in cm
printf("Minimum thickness :\n\n");
printf("(a)Plane polarised light : %.2f x 10^-3 cm \
    n\n",t1*10^3);
printf("(b)Circularly polarised light : %.2f x 10^-3
    cm ",t2*10^3);
```

Scilab code Exa 5.7 Birefringent crystal

```
clc();
clear;
//Given :
lambda = 5890; //Wavelength in A
//(a)Calcite crystal
mu1_o = 1.658;//refractive index for ordinary ray
mu1_e = 1.486;//refractive index for extraordinary
    ray
t1 = 0.0052 ; //thickness in mm
// 1 A = 1.0*10^-7 mm
alpha1 = ((2*%pi*(mu1_o-mu1_e)*t1)/(lambda*10^-7));
    // phase difference in radians
//(b) Quartz crystal
mu2_o = 1.544; //refractive index for ordinary ray
mu2_e = 1.553; //refractive index for extraordinary
    ray
t2 = 0.0234;//thickness in mm
alpha2 = ((2*%pi*(mu2_e-mu2_o)*t2)/(lambda*10^-7));
    // phase difference in radians
printf("(a)Calcite crystal : \n Phase difference is
    %.3f radians \n",alpha1);
printf("(a)Quartz crystal : \n Phase difference is
    %.3f radians",alpha2);
```

Scilab code Exa 5.9 Application of Optical Activity

```
clc();
clear;
//Given :
rho = 6.6; // Specific rotation of sugar in degrees
    g^-1 cm^2
```

```
l = 20; //length in cm
deltad = 1*10^-3;//difference in sugar concentration in g/cm^3
lc = 0.1; // least count in degrees
//Rotation due to optical activity  = rho*l*d
deltatheta = rho*l*deltad; // in degrees
printf("Change in theta :%1.3f degrees.\n\n", deltatheta);

if(deltatheta > lc)
    printf("The concentration of 1 mg/cm^3 will be detected by the given urinalysis tube.");
  else
      printf("The concentration of 1 mg/cm^3 will not be detected.");
end
```

Chapter 6

Quantum Physics

Scilab code Exa 6.1 Quantised energy levels

```
//Quantised energy levels for microscopic and macroscopic systems
clc();
clear;
//Given :
// (a) For a 1s simple pendulum :
T = 1; // time period in s
nu = 1/T; //Frequency in Hz
//Planck's quantisation princple : E_n = n*h*nu
h = 6.625*10^-34 ; //Planck's constant in Js
printf("Energy at First three levels for a 1s simple pendulum :\n\n");
for n1 = 1:3
    E1 = n1*h*nu ; // Energry  in J
    printf("E_%d : %1.3f x 10^-34 J\n",n1,E1*10^34);
end
// (b) For a hydrogen electron
// E_n = (-13.6/n^2)eV
 printf("Energy at First three levels  for a hydrogen electron :\n\n");
for n2 = 1:3
```

```
    E2 = (-13.6/n2^2);//Energy in eV
    printf("E_%d : %.2f J\n",n2,E2);
end

//Now, for a simple pendulum
m = 10; // mass in g
a = 1; // amplitude in cm
omega = 2*%pi*nu; // angular frequency in rad/s
// 1 g = 1.0*10^-3 Kg and 1 cm = 1.0*10^-2 m
E = 1/2*((m*10^-3)*(omega^2)*(a*10^-2)^2); // Energy in J
//Thus,quantum number n = E/h*nu
n = E/(h*nu);
printf("Quantum number n is : %.2f x 10^28 \n\n",n*10^-28);
//(i)Pendulum :
//percentage change in energy = (E_n+1 - E_n)*100/E_n which is equal to [(n+1)*h*nu - n*h*nu]*100/(n*h*nu )
//Therefore , it is (1/n) * 100
pc = (1/n)*100; //percentage change in energy
printf("Percentage change in energy ( pendulum ) is %1.3f x 10^-27 \n\n",pc*10^27);
//(ii)Hyderogen electron :
n_1 = 1; //ground state
n_2 = 2; // next quantum state
E_1 = (-13.6/n_1^2); // Energy in eV
E_2 = (-13.6/n_2^2);//Energy in eV
//percentage change : |((E_2-E_1)*100)|/ |E_1|
pc1 =((E_2-E_1)*100)/(-E_1);//percentage change
printf("Percentage change in energy (hydrogen electron) is %.1f",abs(pc1));
```

Scilab code Exa 6.2 Finding Photon Energy

```
clc();
clear;
//Given :
h = 6.625*10^-34;//Planck's constant in Js
c = 3*10^8 ; //velocity of light in m/s
// 1A = 1.0*10^-10 m
//(a)Energy of a photon :
// E = h*nu or E = h*c/lambda
printf("Energy of a photon is %2.4f x 10^-16 /lambda
    (in A)  J\n",((h*c)*10^10)*10^16);
//1eV = 1.6*10^-19 J
printf("Energy of a photon is %.0f/lambda(in A)  eV\
    n\n",round(((h*c)/(1.6*10^-19))*10^10));
//(b)Visible light Range is 4000-7000 A
lambda1 = 4000;//Wavelength in A
lambda2 = 7000;//Wavelength in A
// 1eV = 1.6*10^-19 J ,
E1 = (h*c)/(lambda1*10^-10*1.6*10^-19); //Energy in
    eV
E2 = (h*c)/(lambda2*10^-10*1.6*10^-19);//Energy in
    eV
printf("Hence the range of energies for visible
    photos is  %.1f eV  to  %.1f eV",E2,E1);
```

Scilab code Exa 6.3 Failure of wave theory

```
clc();
clear;
//Given :
//Power of the source = 10^-5 W = 10^-5 J/s
P = 10^-5 ; //Power in J/s
```

```
r = 10^-9; //radius in m
r1 = 5; // metal plate 5 m away from the source
WF = 5;//Work function in eV
area = %pi*(10^-9)^2 ; //area in m^2
area1 = 4*%pi*r1^2; // area in m^2
P1 = P*(area/area1); // in J/s
// 1eV = 1.6*10^-19 J
t = (WF*1.6*10^-19)/P1 ;// in s
//1 day = 24 hours * 60 minutes * 60 seconds
N = t/(24*60*60); //in days
printf(" It will take %.0f  days \n",round(N));
```

Scilab code Exa 6.4 Determination of h and phi

```
clc();
clear;
//Given :
nu1 = 10*10^14;// Frequency in Hz
nu2 = 6*10^14;// Frequency in Hz
V_01 = 2.37; //Stopping potential in volts
V_02 = 0.72; //Stopping potential in volts
//Einstein's photoeletric equation : h*nu = phi + e*
   V_0
e = 1.6*10^-19 ;// Charge of an electron in C
h = (e*(V_02 - V_01))/(nu2 - nu1);//Planck's
   constant in Js
phi = ((h*nu1)-(e*V_01))/e ; // work function in eV
printf("Plancks constant h is %.1f x 10^-34 Js  and
   Work function phi is %.2f eV ",h*10^34,phi);
```

Scilab code Exa 6.5 Incident wavelength in Compton Scattering

```
clc();
clear;
//Given :
ME = 35*10^3 ; //Maximum energy in eV
theta = %pi; // photon is backscattered
h = 6.625*10^-34; //planck's constant in Js
m0 = 9.1*10^-31; //electron mass in Kg
c = 3*10^8; //Speed of light in m/s
deltalambda = (h*(1-cos(theta)))/(m0*c); // in A
// (h*c/lambda) - (h*c/lambda') = 35 KeV  or  (
   deltalambda/lambda*lambda1) = (35 KeV/h*c)
//Simplifying the above Equation , we will obtain :
   lambda^2 + 0.048 lambda - 0.017
//Roots of the quadratic equation are :
values = [-0.017,0.048,1]; // a,b,c values of the
   quadratic equation
equation = poly(values,'lamb','coeff'); //quadratic
    equation
r = roots(equation); //Roots of the final equation
printf("Incident photon wavelength in Compton
   scattering is %.2f A",r(2));
```

Scilab code Exa 6.6 Observability of Compton effect

```
clc();
clear;
```

```
//Given :
theta = 90; //angle in degrees
m0 = 9.1*10^-31; //electron mass in kg
c = 3*10^8; //Speed of light in m/s
h = 6.625*10^-34; //planck's constant in Js
deltalambda = ((h*(1-cosd(theta)))/(m0*c))*10^10; // in A
//(a) Microwave range
lambda1 = 3.0 ;// wavelength in cm
//lambda1 = 3.0*10^8 A , 1 cm = 1*10^8 A
pc1 = ((deltalambda)*100)/((lambda1*10^8) + deltalambda) ;//percent change in photon energy
printf("Percentage change in energy for radiation in microwave range is : %.0f x 10^-9 \n",pc1*10^9);
//(b) Visible range
lambda2 = 5000 ;// wavelength in A
pc2 = ((deltalambda)/(lambda2 + deltalambda))*100 ; //percent change in photon energy
printf("Percentage change in energy for radiation in visible range is : %.0f x 10^-4 \n",pc2*10^4);
//(c) X-ray range
lambda3 = 1 ; //wavelength in A
pc3 = ((deltalambda)/(lambda3 + deltalambda))*100 ; //percent change in photon energy
printf("Percentage change in energy for radiation in X-ray range is : %.1f\n",pc3);
//(d)Gamma ray range
lambda4 = 0.012 ;// wavelength in A
pc4 = ((deltalambda)/(lambda4 + deltalambda))*100 ; //percent change in photon energy
printf("Percentage change in energy for radiation in Gamma range is : %.1f\n",pc4);
```

Scilab code Exa 6.7 Finding pe by pp ratio

```
clc();
clear;
//Given:
//Photoelectric effect
lambda1 = 2000; //wavelength in A
phi1 = 2.3;// Work function in eV
m = 9.1*10^-31; //electron mass in kg
E1 = 12422/lambda1; // Energy of photon in eV
c = 3*10^8; //Speed of light in m/s
Ee1 = (12422/lambda1)- phi1; // energy of an electron in eV
pe1 = sqrt(2*m*Ee1*1.6*10^-19); //electron momentum in kg m/s
pp1 = (E1*1.6*10^-19)/c ; // Momentum of incident photon in kg m/s
ratio1 = pe1/pp1 ; // (pe/pp)
//Compton effect
lambda2 = 1; // wavelength in A
deltalambda = 0.048; // Compton shift in A
E2 = 12422/lambda2; // Energy of photon in eV
Ee2 = (12422/lambda2)- (12422/(lambda2+deltalambda));//energy of an electron in eV
pe2 = sqrt(2*m*Ee2*1.6*10^-19); //electron momentum in kg m/s
pp2 = (E2*1.6*10^-19)/c ; // Momentum of incident photon in kg m/s
ratio2 = pe2/pp2 ; // (pe/pp)
printf("Photoelectric effect :\n\n");
printf("Electron energy : %.1f eV \n Electron momentum  : %.2f x 10^-24 kg m/s \n Momentum of incident photon : %.2f x 10^-27 kg m/s \n pe/pp : %.0f \n\n",Ee1,pe1*10^24,pp1*10^27,ratio1);
printf("Compton  effect :\n\n");
printf("Electron energy : %.1f eV \n Electron momentum  : %.1f x 10^-23 kg m/s \n Momentum of incident photon : %.2f x 10^-24 kg m/s \n pe/pp :
```

```
%.2f \n\n",Ee2,pe2*10^23,pp2*10^24,ratio2);
```

Scilab code Exa 6.8 Wave particle characteristics

```
clc();
clear;
//Given:
//Gamma-rays,X-rays
lambda1 = 0.01;//Wavelength in A
c = 3*10^8; //Speed of light in m/s
E1 =  12422/lambda1; // Energy in A
p1 = (E1*1.6*10^-19)/c ; //Momentum in kg m/s
//UV
lambda2 = 100;//Wavelength in A
c = 3*10^8; //Speed of light in m/s
E2 =  12422/lambda2; // Energy in A
p2 = (E2*1.6*10^-19)/c ; //Momentum in kg m/s
//IR
lambda3 = 1*10^-4;//Wavelength in m
c = 3*10^8; //Speed of light in m/s
//lambda3 = 1*10^-4*10^10 A , 1 m = 1*10^10 A
E3 =  12422/(lambda3*10^10); // Energy in A
p3 = (E3*1.6*10^-19)/c ; //Momentum in kg m/s
//Microwave
lambda4 = 1;//Wavelength in m
c = 3*10^8; //Speed of light in m/s
//lambda4 = 1*10^10 A , 1 m = 1*10^10 A
E4 =  12422/(lambda4*10^10); // Energy in A
p4 = (E4*1.6*10^-19)/c ; //Momentum in kg m/s
//Radio waves
lambda5 = 100;//Wavelength in m
c = 3*10^8; //Speed of light in m/s
//lambda5 = 100*10^10 A , 1 m = 1*10^10 A
```

```
E5 =  12422/(lambda5*10^10); // Energy in A
p5 = (E5*1.6*10^-19)/c ; //Momentum in kg m/s
printf("Gamma-rays,X-rays : \n  Energy : %.2f x 10^6
    eV \n Momentum : %.1f x 10^-22 kg m/s \n\n",E1
   *10^-6,p1*10^22);
printf(" UV : \n  Energy : %.2f eV \n Momentum : %.1
   f x 10^-26 kg m/s\n\n",E2,p2*10^26);
printf(" IR : \n  Energy : %.4f eV \n Momentum : %.1
   f x 10^-30 kg m/s\n\n",E3,p3*10^30);
printf(" Microwave : \n  Energy : %.2f x 10^-6 eV \n
    Momentum : %.1f x 10^-34 kg m/s\n\n",E4*10^6,p4
   *10^34);
printf(" Radio waves : \n  Energy : %.2f x 10^-8 eV
   \n Momentum : %.1f x 10^-36 kg m/s",E5*10^8,p5
   *10^36);
```

Scilab code Exa 6.9 deBroglie wavelength of an electron

```
clc();
clear;
//Given :
h = 6.625*10^-34; //planck's constant in Js
m = 9.109*10^-31;// electron mass in kg
e = 1.6*10^-19; // charge of an electron in C
//Lambda = h/sqrt(2*m*eV)   here we dont have V , so
    let us caluclate the remaining part.
lambda = h/sqrt(2*m*e);// wavelength in A
// 1 A = 1.0*10^-10 m
printf("Lambda(A) = %.2f/sqrt(V) ",lambda
   /(1.0*10^-10));
```

Scilab code Exa 6.10 de Broglie wavelength

```
clc();
clear;
//Given :
//(a)Rock
h = 6.625*10^-34; //planck's constant in Js
m = 50 ; // mass in g
v = 40; // Speed in m/s
// m = 50*10^-3 kg , 1g = 1.0*10^-3 kg
lambda1 = h/(m*10^-3*v); // Wavelength in m
//(b)For an electron
V = 50; // in volts
lambda2 = 12.28/sqrt(V); // Wavelength in A
printf("De Broglie wavelength \n\n (a)Rock : %.2f x
   10^-34 m \n (b)For an electron : %.2f A",lambda1
   *10^34,lambda2);
```

Scilab code Exa 6.14 Application of uncertainty principle

```
clc();
clear;
//Given:
//(a) Ball
h = 6.625*10^-34; //planck's constant in Js
m1 = 45; //mass in g
v1 = 40; //Speed in m/s
prec1 = 1.5/100 ;//precision
```

```
// m1 = 45*10^-3 kg , 1 g = 1.0*10^-3 kg
p1 =m1*10^-3*v1 ; // momentum in kg m/s
//(deltap/p)*100 = 1.5
deltap1 = prec1*p1 ;
deltax1 = h/deltap1; // uncertainty in position in m
printf("Uncertainty in position for a ball : %.2f x
    10^-32 m \n",deltax1*10^32);
//(b) Electron
m2 = 9.1*10^-31; //electron mass in kg
v2 = 2*10^6 ; // Speed in m/s
prec2 = 1.5/100 ; // precision
p2 = m2*v2; // momentum in kg m/s
//(deltap/p)*100 = 1.5
deltap2 = prec2*p2 ;
deltax2 = h/deltap2; // uncertainty in position in m
// 1 A = 1.0*10^-10 m
printf("Uncertainty in position for an electron : %
    .0f A \n",deltax2/(1.0*10^-10));
```

Scilab code Exa 6.17 Application of Schrodinger equation

```
clc();
clear;
//Given:
//(a) Marble
h = 6.625*10^-34; //planck's constant in Js
m1 = 10; // mass in g
L1 = 10; // width in cm
// m1 = 10*10^-3 kg , 1 g = 1.0*10^-3 kg and L1 =
    10*10^-2 m , 1 cm = 1.0*10^-2 m
printf("(a)Marble \n\n");
for n1 = 1:3
    En1 = (n1^2*h^2)/(8*m1*10^-3*(L1*10^-2)^2); //
```

```
Energry in J
    printf("E_%d : %.1f x 10^-64 J\n",n1,En1*10^64);
end
//(b) For an electron
m2 = 9.1*10^-31; //electron mass in kg
L2 = 1 ; // width in A
//L2 = 1*10^-10 m , 1 A = 1.0*10^-10 m
printf("(b)For an electron \n\n");
for n2 = 1:3
    En2 = (n2^2*h^2)/(8*m2*(L2*10^-10)^2); // Energry in J
    printf("E_%d : %.1f eV\n",n2,(En2*6.24150934*10^18)); // 1J = 6.24150934*10^18 eV
end
```

Chapter 7

Atomic Physics

Scilab code Exa 7.1 Hydrogen atom

```
clc();
clear;
//Given :
n =1 ; // ground state
m = 9.109382*10^-31; //electron mass in kg
h = 6.625*10^-34; //planck's constant in Js
e = 1.602176*10^-19; // Charge of an electron in C
e0 = 8.854188*10^-12; // Vacuum permittivity in F/m
r1 = (n^2*h^2*e0)/(%pi*m*e^2);// Radius in A
v1 = e^2/(2*h*e0*n); // Velocity in m/s
E1 = -((m*e^4)/(8*n^2*h^2*e0^2)); // Energy of an
    electron in eV
// 1 A = 1.0*10^-10 m , 1 eV = 1.6*10^-19 J
printf("For hydrogen atom : \n Radius  = %.2f A \n
    Velocity = %.1f x 10^6 m/s \n Energy of an
    electron = %.1f eV",r1*10^10,v1*10^-6,E1
    /(1.6*10^-19));
```

Scilab code Exa 7.2 Bohr Theory

```
clc();
clear;
//Given :
//(a)
m = 9.109382*10^-31; //electron mass in kg
c = 2.997925*10^8; //Speed of light  in m/s
h = 6.626069*10^-34; //planck's constant in Js
e = 1.602176*10^-19; // Charge of an electron in C
e0 = 8.854188*10^-12; // Vacuum permittivity in F/m
R = (m*e^4)/(8*h^3*e0^2*c);// Rydberg constant in m^-1
printf("Rydberg constant for hydrogen : %.2f cm^-1\n\n",R*10^-2);
//(b)
M = 1.672622*10^-27; // proton mass in kg
R1 = ((m*e^4)/(8*h^3*e0^2*c))*(1/(1 + (m/M))); // Rydberg Constant in m^-1
//1 m^-1 = 1.0*10^-2 cm^-1
printf("Rydberg Constant is %.2f cm^-1",R1*10^-2);
```

Scilab code Exa 7.3 Bohrs theory for helium

```
clc();
clear;
//Given :
```

```
RH= 109677.58; //Rydberg constant for Hydrogen in cm
    ^-1
RHe = 109722.269; //Rydberg constant for Helium in
    cm^-1
//Ratio = M/m
Ratio = ((4*RH)- (RHe))/(4*(RHe-RH));
printf("M/m value is : %.1f ",Ratio);
```

Scilab code Exa 7.4 Bohrs radius

```
clc();
clear;
//Given
h = 6.625*10^-34; //planck's constant in Js
m = 9.1*10^-31; //electron mass in kg
E1 = 13.6; //Energy of electron in eV
//1 eV = 1.6*10^-19 J
p = sqrt(2*m*E1*1.6*10^-19); //momentum in kg m/s
deltax = h/(2*%pi*p);
// 1 A = 1.0*10^-10 m
printf("Uncertainty in position : %.2f A",deltax
    /(1.0*10^-10));
```

Chapter 8

Nuclear Physics

Scilab code Exa 8.1 Nuclear and atomic density

```
clc();
clear;
//Given:
mp = 1.67*10^-27 ; // proton mass in kg
r0 = 1.2*10^-15; // constant in m
a0 = 0.5*10^-10; // atomic dimensions in m
//rho_nucleus = nuclear mass/ nuclear volume
rho_nucleus = (3*mp)/(4*%pi*r0^3); // nuclear
   density in kg/m^3
//ratio = rho_nucleus/rho_atom = (a0/r0)^3
ratio = a0^3/r0^3;
printf("Nuclear density is %.1f x 10^17 kg/m^3 \n",
   rho_nucleus*10^-17);
printf("Nuclear density is %.1f x 10^13 times Atomic
    density.",ratio*10^-13);
```

Scilab code Exa 8.2 Rest mass of a pion

```
clc();
clear;
//Given :
h = 1.05*10^-34; //planck's constant in Js
m = 9.1*10^-31; //electron rest mass in kg
c = 3*10^8; //Speed of light in m/s
b = 1.7*10^-15; // range of nuclear force in m
m_pi = h/(b*c); // rest mass of a pion in kg
t = m_pi/m; // times the rest mass of an electron
printf("Rest mass of a pion is %d times the rest
    mass of an electron",t);
// textbook answer is 220 , because  approximate
    value for m_pi was considered.
```

Scilab code Exa 8.3 Nuclear and Electronic Binding Energy

```
clc();
clear;
//Given :
mp = 1.007276470 ; // proton mass in u
mn = 1.008665012; // neutron mass in u
md = 2.013553215; // deuteron mass in u
//E = ( mp + mn - md)*c^2
// 1 u * c^2 = 931.5 MeV , where 1 u = 1.66*10^-27
    kg and c = 3*10^8 m/s
E = (mp + mn - md)*931.5; // Binding energy in MeV
printf("Binding energy : %.3f MeV",E);
```

Scilab code Exa 8.4 Average Binding Energy

```
clc();
clear;
//Given :
m_alpha = 4.001506106; // mass of an alpha particle in u
mp = 1.007276470 ; // proton mass in u
mn = 1.008665012; // neutron mass in u
//E = ( 2*mp + 2*mn - m_alpha)*c^2
// 1 u * c^2 = 931.5 MeV , where 1 u = 1.66*10^-27 kg and c = 3*10^8 m/s
E = (2*mp + 2*mn - m_alpha)*931.5; // Binding energy in MeV
printf("Average binding energy per nucleon : %.3f MeV",E/4);
```

Scilab code Exa 8.5 Q value of a nuclear reaction

```
clc();
clear;
//Given :
Mn = 14.00753; //mass of Nitrogen 14 in u
Mo = 17.0045; // mass of Oxygen 17 in u
m_alpha = 4.00387; // mass of alpha particle in u
mp = 1.00184; // mass of proton in u
//Q = (m_alpha + Mn - Mo - mp)*c^2
//// 1 u * c^2 = 931.5 MeV , where 1 u = 1.66*10^-27 kg and c = 3*10^8 m/s
Q = (m_alpha + Mn - Mo - mp)*931.5 ;// Q value in MeV
printf("Q value is %.1f  MeV",Q);
```

Scilab code Exa 8.7 Angle of ejection

```
clc();
clear;
//Given :
Q = 4 ;// in MeV
Ex = 2; // in MeV
Ey = 5 ; // in MeV
mx = 4; // in u
my = 1 ; // in u
My =13; // in u
theta = acosd(( (Ey*(1 + (my/My))) - (Ex*(1 - (mx/My
    ))) - Q )/((2/My)*sqrt(mx*Ex*my*Ey))); // angle
    of ejection in degrees
printf("Angle of ejection is %.0f degrees",theta);
```

Scilab code Exa 8.8 Electronic and nuclear energy levels

```
clc();
clear;
//Given :
h = 6.625*10^-34 ; //planck's constant in Js
me = 9.1*10^-31 ; //electron mass in kg
mn = 1.67*10^-27;// a nucleon mass in kg
//(a)For electron
L1 = 1; // in A
//E = (n^2*h^2)/(8*m*L^2) , here n value is not
    given , so let us calculate the remaining part (
```

```
neglecting n^2 in the formula)
//L1 = 1*10^-10 m , 1A = 1.0*10^-10 m
E1 = h^2/(8*me*(L1*10^-10)^2); // energy in J
//(b)For nucleon
L2 = 1; // in fm
//E = (n^2*h^2)/(8*m*L^2) , here n value is not given , so let us calculate the remaining part (neglecting n^2 in the formula)
//L2 = 1*10^-15 m , 1 fm = 1.0*10^-15 m
E2 = h^2/(8*mn*(L2*10^-15)^2);//energy in J
printf("Energy for an electron : %.1f x 10^-17 x n^2 J \n",E1*10^17);
printf("Energy for a nucleon : %.2f x 10^-11 x n^2 J",E2*10^11);
```

Scilab code Exa 8.9 Energy released in Fission

```
clc();
clear;
//Given :
Na = 6.023*10^23 ; // Avogadro constant in atoms/mole
LE = 200 ; // liberated energy in MeV
mm = 235; // molar mass  of U 235 in gm/mole
// 1 eV = 1.6*10^-19 J , 1 MeV = 1.0*10^6 eV
RE = (Na*LE*1.6*10^-19*10^6)/mm ; //released energy in J
// 1 cal = 4.187 J
EC = RE/4.187 ; // energy in cal
//Burning 1 kg of coal releases 7000 K cal of energy
Q1 = EC/(7000*10^3); // Quantity of Coal in Kg
//Exploding 1 kg of TNT releases 1000 cal of energy
Q2 = EC/1000; // Quantity of TNT in kg
```

```
printf("Energy released : %.0f x 10^10 cal \n",EC
    *10^-10);
printf(" %.1f tonnes of Coal\n",Q1*10^-3);
printf(" %.0f tonnes of TNT\n",Q2*10^-3);
// Results obtained differ from those in textbook ,
    because approximate values were considered in
    textbook.
```

Scilab code Exa 8.10 Power output

```
clc();
clear;
//Given :
Na = 6.023*10^23 ; // Avogadro constant atoms/mole
LE = 200 ; // liberated energy in MeV
mm = 235*10^-3; // molar mass of U 235 in gm/mole
p = 30/100 ; // conversion efficiency
// 1 eV = 1.6*10^-19 J , 1 MeV = 1.0*10^6 eV
RE = (Na*LE*1.6*10^-19*10^6)/mm ; //released energy
    in J per day
// 1 day = 24 hrs * 60 mins * 60 sec
P = RE/(24*60*60); // Power output in W per day
// 1 cal = 4.187 J
EC = RE/4.187 ; // energy in cal
//Burning 1 kg of coal releases 7000 K cal of energy
Q1 = EC/(7000*10^3); // Quantity of Coal in Kg per
    day
EP = p*P ; // electric power in W
printf(" %.0f tonnes of Coal\n",Q1*10^-3);
printf(" Electric power for 30 percent conversion
    efficiency : %.1f kW",EP*10^-3);
// Results obtained differ from those in textbook ,
    because approximate values were considered in
```

```
textbook.
```

Scilab code Exa 8.11 Radioactive dating of a tree

```
clc();
clear;
//Given :
T_half = 5730; // carbon 14 half life in years
Na = 6.023*10^23; // Avogadro constant in nuclei/mole
M = 25;// charcoal mass in gm
mm = 12;// molar mass of carbon 12 in gm/mole
a = 250 ; // disinitegrations per minute  (Carbon 14 activity)
// 1 year = 525949 minutes
lambda = 0.693/(T_half*525949);// disinitegrations per minute per nucleus
N0_1 = (Na/mm)*M ; // Number of nuclei (Carbon 12)
// Carbon 14 to Carbon 12  ratio  = 1.3*10^-12
N0_2 = 1.3*10^-12*N0_1 ; // Number of nuclei (Carbon 14)
R0 = N0_2*lambda ; // disinitegrations per minute per nucleus
a0 = R0 ; // initial activity
t = log(a0/a)/lambda ;
// 1 year = 525949 minutes
printf("The tree died %d  years ago",t/525949 );
// Result obtained differs from the textbook, because R0 value obtained here is 375.1025, where as in textbook it is 374.
```

Scilab code Exa 8.12 Radioactivity of iodine 131

```
clc();
clear;
//Given :
T_half = 8 ; // iodine 131 haf life in days
lambda = 0.693/T_half ; // decay constant in decays/
    day
N0 = 20 ; // mass in mg
t = 48; // time in days
N = N0*exp(-lambda*t); // in mg
printf("Original amount : %d mg \n",N0);
printf("Remaining amount after 48 days : %.3f mg",N)
    ;
```

Scilab code Exa 8.13 Co 60 gamma rays

```
clc();
clear;
//Given :
RBE = 0.7 ; //RBE factor for cobalt 60 gamma rays
dose = 1000 ; // dose in rad
e = RBE*dose; // equivalent dose in rem
printf("Equivalent dose is %d  rem",e);
```

Chapter 9

Structure and Properties of Matter

Scilab code Exa 9.3 Miller indices of planes

```
clc();
clear;
//Given :
//Intercepts
ix = 1/3 ; //along x-axis
iy = 2/3; // along y-axis
iz =1; // along z-axis
//Reciprocals
rx = 1/ix;
ry = 1/iy;
rz = 1/iz;
//Conversion
x = rx*2;
y = ry*2;
z = rz*2;
printf("Miller indices of the plane are : ( %d %d %d
    )",x,y,z);
```

Scilab code Exa 9.7 Determination of crystal structure

```
clc();
clear;
//Given:
n = 1;
theta = 30; // angle in degrees
lambda = 1.67; // wavelength in A
r = 1.25; // atomic radius in A
//Bragg's Law : 2*d*sin(theta) = n*lambda , d= d111
d111 = (n*lambda)/(2*sind(theta));
//plane (111)
h =1;k=1;l=1;
//dhkl = a/sqrt(h^2 + k^2 + l^2)
a = d111*sqrt(h^2 + k^2 + l^2); // in A
ratio = r/a;
printf(" Since, r/a = %.4f and r = %f*a  Crystal
   Structure : BCC",ratio,ratio);
```

Scilab code Exa 9.8 Determination of density

```
clc();
clear;
//Given:
n = 1;
theta = 30; //angle in degrees
lambda = 2.88 ; // wavelength in A
M = 108; // atomic weight in kg
```

```
Z = 4; // unit cell of silver is FCC
Na = 6.023*10^26 ;// Avogadro constant in kmole
//Bragg's Law : 2*d*sin(theta) = n*lambda , d = d110
d110 = (n*lambda)/(2*sind(theta)); // in A
//plane (110)
h =1;k=1;l=0;
//dhkl = a/sqrt(h^2 + k^2 + l^2)
a = d110*sqrt(h^2 + k^2 + l^2); // in A
//1 A = 1.0*10^-10 m
rho = (Z*M)/(Na*(a*10^-10)^3); // density in kg/m^3
printf(" Density of silver : %.1f kg/m^3",rho);
```

Chapter 10

Dielectric and Magnetic Materials

Scilab code Exa 10.1 Electronic polarisation

```
clc();
clear;
//Given:
er = 1.0000684; // relative dielectric constant
N = 2.7*10^25; // atoms/m^3
//We know, er - 1 = 4*pi*N*R^3
R = ((er-1)/(4*%pi*N))^(1/3) ; // in m
printf("R : %.1f x 10^-10 m",R*10^10);
```

Scilab code Exa 10.2 Diamagnetism

```
clc();
clear;
//Given :
```

```
R = 1; // radius in A
N = 5*10^28 ; //  atoms/m^3
mu_0 = 4*%pi*10^-7; // permiability of free space in
    H/m
mu_r = 1;//relative permiability
m = 9.1*10^-31 // electron mass in kg
e = 1.6*10^-19 ; // charge of an electron in C
// R = 1*10^-10 m  because 1 A = 1.0*10^-10 m
chi = -(N*e^2*(R*10^-10)^2*mu_0*mu_r)/(4*m); //
    Susceptibility of diamagnetic material
printf("Susceptibility of diamagnetic materials  is
    %.2f x 10^-5",chi*10^5);
//Result obtained differs from that in textbook,
    because in textbook only the order of 10 is
    considered .
```

Scilab code Exa 10.3 Dipole moment and polarisability

```
clc();
clear;
//Given :
e0 = 8.85*10^-12 ; // dielectric constant in farad/m
er1 = 1.006715 ; //relative dielectric constant
er2 = 1.005970;// relative dielectric constant
T1 = 300 ; // Temperature in K  (273+27 = 300 K)
T2 = 450; // Temperature in K  (273 + 177 = 450 K)
k = 1.38*10^-23; // in J/K
N = 2.44*10^25 ; // molecules/m^3
//e0*(er1 - er2)= ((N*mu_p^2)/(3*k))*((1/T1)- (1/T2)
    )
mu_p = sqrt((e0*(er1 - er2)*3*k)/(((1/T1)-(1/T2))* N
    )); //dipole moment in C m
D = 3.3*10^-30; // dipole of 1 Debye is equal to
```

```
3.33 x 10^-30 C m
printf("Dipole moment = %.2f debye \n",mu_p/D);
//e0*(er1 - 1) = N*(alpha_e + alpha_i + (mu_p^2/3*k*T1))
Sum = ((e0*(er1 - 1))/N) - ((mu_p)^2/(3*k*T1)); // alpha_e + alpha_i  in farad m^2
printf("Sum =  %.1f x 10^-39 farad m^2",Sum*10^39);
```

Scilab code Exa 10.4 Orientational polarisation

```
clc();
clear;
//Given :
mu_p = 1.2 ;// dipole moment in debye units
T = 300 ; // Temperature in Kelvin ( 273+27 = 300 K)
k = 1.38*10^-23 ; //  in J/K
per = 0.5/100 ; // percentage of saturated polarisation
// 0.05*N*mu_p = (N*(mu_p)^2*E/(3*k*T))
E = (3*k*T*per)/(mu_p*3.33*10^-30); // External field in V/m
printf(" E = %.2f x 10^7 V/m",E*10^-7);
```

Scilab code Exa 10.5 Susceptibility of paramagnetic materials

```
clc();
clear;
//Given :
N = 5*10^28 ;// number of dipoles per m^3
```

```
betaa = 1;// Bohr magneton
T = 300 ; // Room temperature in k
k = 1.38*10^-23 ; //  in J/K
mu_0 = 4*%pi*10^-7 ; //Magnetic permeability in H/m
//1 Bohr magneton = 9.27     10^-24 Am^2.
chi = (N*mu_0*betaa*(1*9.27*10^-24)^2)/(k*T);
printf("Susceptibility = %.2f x 10^-3",chi*10^3);
//Result obtained differs from that in textbook, because in textbook only the order is considered.
```

Scilab code Exa 10.6 Relative dielectric constant

```
clc();
clear;
//Given :
M = 32; // Atomic weight in kg/kmole
Na =6.023*10^26 ; // Avogadro constant in atoms/kmole
alpha_e = 3.28*10^-40; // electronic polarisability in farad/m^2
rho = 2.08; //density in gm/cm^3
e0 = 8.85*10^-12 ; // dielectric constant in farad/m
// (er - 1)/(er + 2)  = (N*alpha_e/3*e0)
//1 gm = 1.0*10^-3 kg , 1 cm^3 = 1.0*10^-6 m^3
N = (Na*(rho*10^3))/M; //  atoms/m^3
er =( 2*((N*alpha_e)/(3*e0)) + 1 )/(1 - ((N*alpha_e)/(3*e0)));
printf("Relative dielectric constant = %.2f ",er);
```

Scilab code Exa 10.7 Power loss due to hysteresis

```
clc();
clear;
//Given :
area = 50000; // area of hysteresis on a graph
axis1 = 10^-4 ; // units of scale in Wb/m^2
axis2 = 10^2; // units of scale in A/m
vol = 0.01; // volume in m^3
F = 50; //frequency in Hz
E1 = area*axis1*axis2; // Energy lost per cycle in J
    /m^3
E2 = E1*vol ; // Energy lost in core per cycle in J
P = E2*F; // Power loss in W
printf("Power loss = %d W ",P);
```

Scilab code Exa 10.8 Classical model of internal field

```
clc();
clear;
//Given :
mu_d = 9.27*10^-24; // Bhor magneton in Am^2
mu_0 = 4*%pi*10^-7; // Magnetic permiability in H/m
r = 2; // dipoles distance in A
//U = mu_d*B = -( mu_0*mu_d^2)/(2*pi*r)
//r = 2*10^-10 m , 1 A = 1.0*10^-10 m
U = ( mu_0*mu_d^2)/(2*%pi*(r*10^-10)^3); // Energy
printf("U = %.1f x 10^-25 ",U*10^25);
```

Scilab code Exa 10.9 Saturation Magnetisation

```
clc();
clear;
//Given :
a = 2.87; // lattice constant in A
mu = 4; // 4 Bohr magnetons/atom
// BCC = 2 atoms/unit cell , 1 A = 1.0*10^-10 m
N = 2/(2.87*10^-10)^3; // atoms/m^3
//1 Bohr magneton = 9.27*10^-24 Am^2
Msat = N*mu*9.27*10^-24;// Saturation in
    magnetisation in A/m
printf(" Saturation Magnetisation = %.2f x 10^6 A/m"
    ,Msat*10^-6);
```

Scilab code Exa 10.10 Electronic and Ionic polarisability

```
clc();
clear;
//Given :
er = 6.75 ; // relative dielectric constant for
    glass
f = 10^9 ;// frequency in Hz
n = 1.5;// refractive index of glass
e0 = 8.85*10^-12; // dielectric constant in farad/m
//Pe = e0*(n^2 - 1)*E , Pi = e0*(er - n^2)*E , P =
    Pi + Pe = e0*(er - 1)*E
//Percentage = [(e0*(er - n^2)*E)/(e0*(er -1)*E)
    ]*100 , both the E's cancel each other
per = [(e0*(er - n^2))/(e0*(er -1))]*100;//
    percentage
printf("Percentage = %.1f",per);
```

Chapter 11

Conductors Semiconductors and Superconductors

Scilab code Exa 11.3 Fermi energy in metals

```
clc();
clear;
//Given :
n =8.48*10^28; // number of conduction electrons / m
    ^3
Ef = 3.65*10^-19*(n^0.6667);//Fermi energy in eV
printf("Fermi energy : %.2f eV ",Ef);
```

Scilab code Exa 11.4 Fraction of electrons

```
clc();
clear;
//Given :
Ef = 7.04 ; // Ef for copper in eV
```

```
kT = 0.026; // kT value at room temperature in eV
F = (3/2)*(0.026/7.04); // Fraction of electrons
printf("Fraction of electrons which are excited  are %.4f  or %.2f percentage.",F, F*100);
```

Scilab code Exa 11.6 Intrinsic resistivity

```
clc();
clear;
//Given :
ni1 = 2.5*10^19; // per m^3 for Ge
ni2 = 1.5*10^16; // per m^3 for Si
mu_e1 = 0.38; // mobility of free electrons for Ge in m^2/Vs
mu_h1 = 0.18; //mobility of holes for Ge in m^2/Vs
mu_e2 = 0.13;//mobility of free electrons for Si in m^2/Vs
mu_h2 = 0.05;//mobility of holes for Si in m^2/Vs
e = 1.6*10^-19; // charge of an electron in C
sigma1 = ni1*e*(mu_e1 + mu_h1); // intrinsic conductivity in mho m^-1 for Ge
sigma2 = ni2*e*(mu_e2 + mu_h2);// intrinsic conductivity in mho m^-1 for Si
rho1 = 1/sigma1; //intrinsic resistivity in ohm m for Ge
rho2 = 1/sigma2;//intrinsic resistivity in ohm m for Si
printf("Resistivity of Ge %.3f ohm m \n",rho1);
printf("Resistivity of Si %.3f x 10^3 ohm m",rho2*10^-3);
```

Scilab code Exa 11.7 Variation of n by N

```
clc();
clear;
//Given :
//Fraction F = n/N
Eg = 0.72; // Energy gap in eV
k = 0.026/300;// kT value at 300 K , so k = kT/T
T1 = 30; // Temperature in K
T2 = 300; //Temperature in K
T3 = 1210;//Temperature in K
//Fraction of electrons : n/N = exp(-Eg/2*k*T)
F1 = exp(-Eg/(2*k*T1));
F2 = exp(-Eg/(2*k*T2));
F3 = exp(-Eg/(2*k*T3));
printf(" For 30 K , n/N  = %.1f x 10^-61\n",F1
    *10^61);
printf(" For 300 K , n/N  = %.1f x 10^-7\n",F2*10^7)
    ;
printf(" For 1210 K , n/N  = %.3f \n",F3);
```

Scilab code Exa 11.8 Variation of n by N

```
clc();
clear;
//Given :
Eg1= 0.72; //Energy gap for Germanium in eV
Eg2= 1.10; //Energy gap for Silicon in eV
```

```
Eg3= 5.6; //Energy gap for diamond in eV
//Fraction of electron : n/N = exp(-Eg/(2*k*T)) , k*T = 0.026 eV
F1 = exp(-Eg1/(2*0.026)); // For Germanium
F2 = exp(-Eg2/(2*0.026)); // For Silicon
F3 = exp(-Eg3/(2*0.026)); // For diamond
printf("For Germanium , n/N = %.1f x 10^-7\n",F1*10^7);
printf("For Silicon , n/N = %.1f x 10^-10\n",F2*10^10);
printf("For diamond, n/N = %.1f x 10^-47",F3*10^47);
```

Scilab code Exa 11.9 Ef equals to Ec

```
clc();
clear;
//Given :
D = 5*10^28; // density of atoms in silicon per m^3
C = 2.0*10^8; //donor concentration
ND = D/C; // donor atoms density per m^3
// ND = 4.82*10^21*T^(3/2)
T = (ND/(4.82*10^21))^(2/3);
printf("Temperature = %.2f K",T);
```

Scilab code Exa 11.10 Si doped with phosphorus

```
clc();
clear;
//Given :
```

```
Ecd = 0.045; // Ec-Ed in eV
Ecf = 0.035; // Ec-Ef in eV
Efd = 0.01;// Ef-Ed in eV
Ev = 0; // in eV
Ef = 1.065; // in eV
me = 9.1*10^-31;// electron mass in kg
m_e = 0.31*me; // free electron mass
m_h = 0.38*me;// hole mass
kT = 0.026; // kT value at room temperature
h = 6.625*10^-34; // planck's constant in Js
Nc = 2*((2*%pi*m_e*kT*1.6*10^-19)/(h^2))^(3/2); //
    per m^3
Nv = 2*((2*%pi*m_h*kT*1.6*10^-19)/(h^2))^(3/2); //
    per m^3
//(a)
// Nc*exp[-(Ec-Ef)/kT] = Nd*[1 - 1/(1+ exp[(Ed-Ef)/
    kT])]
//Ed - Ef = -(Ef-Ed) = - Efd
Nd = (Nc*exp(-Ecf/kT))/(1 - (1/(1+exp(-Efd/kT))));
    // per m^3
//(b)
Nd_plus = Nd*(1 - (1/(1 + exp(-Efd/kT)))); // per m
    ^3
//(c)
n = Nc*exp(-Ecf/kT); // per m^3
//(d)
p = Nv*exp((Ev-Ef)/kT);// per m^3
printf("Nd = %.1f x 10^24 / m^3 \n",Nd*10^-24);
printf("Nd_plus = %.2f x 10^24 / m^3 \n",Nd_plus
    *10^-24);
printf("n = %.2f x 10^24 / m^3\n",n*10^-24);
printf("p = %.1f x 10^6 / m^3",p*10^-6);
```

Scilab code Exa 11.11 Silicon wafer doped with phosphorus

```
clc();
clear;
//Given :
ni = 1.5*10^16; // ni for Si in m^-3
mue = 0.135; // mobility of free electrons in m^2/Vs
muh = 0.048; // mobility of holes in m^2/Vs
Nd = 10^21; // phosphorus atoms/m^3
e = 1.6*10^-19;// charge of an electron in C
//(a)
n = Nd; // electrons/m^3
//(b)
p = ni^2/Nd; // holes/m^3
//(c)
sigma = e*(n*mue + p*muh); // conductivity in mho m
    ^-1
rho = 1/sigma; // resistivity in ohm m

printf("Major carrier concentration = %.1f x 10^21
    electrons/m^3 \n",n*10^-21);
printf("Minor carrier concentration = %.2f x 10^11
    holes/m^3\n",p*10^-11);
printf("Resistivity = %.3f ohm m",rho);
```

Scilab code Exa 11.12 Increase in conductivity

```
clc();
clear;
//Given :
Eg = 1.1;// Energy gap in eV
T1 = 300 ;// Temperature in K
T2 = 473; // Temperature in K (273+ 200 = 473 K)
```

```
k = 8.62*10^-5 ; // in eV
// sigma = A*exp(-Eg/(2*k*T))
//Ratio = sigma_473/sigma_300
Ratio = exp((-Eg/(2*k))*((1/T2)-(1/T1)));
printf("Thus, sigma_473 is %d times sigma_300",
    Ratio);
```

Scilab code Exa 11.13 Photon energy

```
clc();
clear;
//Given :
Eg1 = 0.72; // Energy gap for Ge in eV
Eg2 = 1.1; // Energy gap for Si in eV
Eg3 = 1.32; // Energy gap for GaAs in eV
// lambda = c/v = (c*h)/Eg   or lambda(A) = 12422/Eg
    (eV)
lambda1 = 12422/Eg1; // wavelength in A (Ge)
lambda2 = 12422/Eg2; // wavelength in A (Si)
lambda3 = 12422/Eg3; // wavelength in A (GaAs)
printf("Wavelength for Ge = %.1f A \n",lambda1);
printf("Wavelength for Si = %.1f A \n",lambda2);
printf("Wavelength for GaAs = %.2f A",lambda3);
```

Scilab code Exa 11.14 Increase in conductivity

```
clc();
clear;
//Given :
```

```
sigma = 4*10^-4; // conductivity at room temperature in ohm^-1 m^-1
M = 28.1; // atomic weight in kg/kmole
d = 2330; // density in kg/m^3
dop = 10^8 ;// doping per 10^8 silicon atoms
e = 1.6*10^-19; // charge of an electron in C
mue = 0.135; // mobility of free electrons for silicon in m^2/Vs
Na = 6.023*10^26 ; // Avagadro's constant in atoms/kmole
N = (d*Na)/M; //atoms/m^3
Nd = N/dop; // per m^3
n = Nd; // electron concentration / m^3
sigma1 = n*e*mue; // conductivity in ohm^-1 m^-1
t = sigma1/sigma; // number of times the conductivity increased
printf("Conductivity increased %d times .",t);
//Result obtained differs from that in textbook, because approximate value for sigma1 was considered.
```

Chapter 12

Diodes and Transistors

Scilab code Exa 12.1 Determination of V0

```
clc();
clear;
//Given:
sigma_n = 10^4; //conductivity in mho/m
sigma_p = 10^2; // conductivity in mho/m
e = 1.6*10^-19;// charge of an electron in C
kT = 0.026 ;// k*T value at room temperature in eV
ni = 2.5*10^19; // per m^3
mue = 0.38; // mobility of free electrons in m^2/Vs
muh = 0.18;// mobility of free electrons in m^2/Vs
// sigma_n = e*n*mue  and sigma_p = e*p*muh
nn0 = sigma_n/(e*mue); // per m^3
pp0 = sigma_p/(e*muh);// per m^3
np0 =( ni^2)/pp0; // in m^-3
// V0 = (kT/e)*log(nn0/np0) , but we consider only
   kT because kT/e = 0.026 eV/e  , both the e's
   cancel each other.Finally we obtain the answer in
    Volts
V0 = (kT)*log(nn0/np0); // in V
printf("V0 = %.2f V",V0);
```

Scilab code Exa 12.2 Carrier concentration

```
clc();
clear;
//Given :
//(a)Forward bias of 0.1 V
// np = np0*exp[eV/kT] , here we dont have np0 value
   , so we will calculate the remaining part.
kT = 0.026; // in eV
np = exp(0.1/kT);
printf("(a) np = %.0f x np0 \n",np);
//(b)Reverse bias of 1 V
// np = np0*exp[-eV/kT] , here we dont have np0
   value, so we will calculate the remaining part.
np1 = exp(-1/kT);
printf("(b) np = %.2f x 10^-17 x np0 \n",np1*10^17);
```

Scilab code Exa 12.3 Current through pn junction diode

```
clc();
clear;
//Given :
I0 = 0.1; // muA
kT = 0.026; // kT value at room temperature
//Forward bias of 0.1 V
// I = I0[exp(eV/kT) - 1]
//  since I = I0*(exp(0.1 eV/kT (eV))), both the eV'
   s cancel each other , so it is only  I = I0*(exp
```

```
    (0.1/kT) - 1) while evaluating.
I = I0*(exp(0.1/kT) - 1) // in muA
printf("Current = %.2f muA ",I);
```

Scilab code Exa 12.4 Voltage regulation using Zener diode

```
clc();
clear;
//Given :
Vin = 36; // Input Voltage in V
Vb = 6; // Zerner Breakdown Voltage in V
Vr = Vin-Vb; // Volatge drop across resistor
R = 5*10^3; // resistance in ohm
Rl = 2*10^3; // load resistance in ohm
I = Vr/R; // current in A
Il = Vb/Rl; // current  in A
Iz = I - Il ;// current in A
//(a)
Vin1 = 41; // Input Voltage in V
I1 = (Vin1-Vb)/R; // current in A
Iz1 = I1-Iz; // current in A
//(b)
Rl1 = 4*10^3; //load  resistance in ohm
Il1 = Vb/Rl1; // current in A
Iz2 = I - Il1; // current in A
printf("Input volatge = 41 V , Iz = %.0f mA\n",Iz1
    *10^3);
printf("Load resistance = 4k ohm , Iz = %.1f mA",Iz2
    *10^3);
```

Scilab code Exa 12.5 Voltage gain

```
clc();
clear;
//Given :
deltaIE = 2; // in mA
deltaIB = 5; // in mA
Rl = 200*10^3; // load resistance in ohm
ri = 200; // input resistance in ohm
// IE= IB + IC   ,    1 muA = 1.0*10^-3 mA
deltaIC = deltaIE - deltaIB*10^-3 ;// in mA
alpha = deltaIC/deltaIE;
A = alpha*(Rl/ri);
printf("Voltage gain = %.1f ",A);
```

Chapter 13

Charged Particles in Electric and Magnetic Fields

Scilab code Exa 13.1 Electron in an electric field

```
clc();
clear;
//Given :
// E = 2*10^9*t  V/m
// a_x = e*E/m , where e = 1.6*10^-19 C , m = 9.12 10^-31 kg
// a_x = 3.52*10^20*t m/s^2
// v_x = integral of a_x dt
//(a)
function a_x = f(t),a_x = 3.530*10^20*t,endfunction
v_x = intg(0,50*10^-9,f); // electron speed in m/s at time = 50 ns
printf("v_x = %.1f x 10^5 m/s\n",v_x*10^-5);
//(b)
//v_x = 1.76*10^20*t^2 m/s
function vx = v(t),vx = 1.76*10^20*t^2 ,endfunction
x = intg(0,50*10^-9,v);// distance covered in m  in 50 ns
printf("x = %.2f mm\n",x*10^3);
```

```
//(c)
//x = 5.87*10^19*t^3 m
X = 5*10^-2; //distance between plates in m
t = (X/(5.87*10^19))^(1/3); // time required in s
printf("t = %.2f x 10^-7 s",t*10^7);
```

Scilab code Exa 13.5 Projected electron

```
clc();
clear;
//Given :
u = 5*10^5; //horizontal velocity in m/s
alpha = 35; // in degrees
E = 200 ;// Electric field in V/m
e = 1.6*10^-19; // electron charge in C
m = 9.12*10^-31; // electron mass in kg
a = (-e*E)/m; // horizontal range in m/s^2
//(a);
z_max = (-(u^2)*(sind(alpha))^2)/(2*a); // maximum
    penetration in m
//(b)
T = (-2*u*sind(alpha))/a; // Time of flight in s
//(c)
H = (-(u^2)*(sind(2*alpha)))/a; // horizontal range
    in m
printf("z_max = %.1f mm \n",z_max*10^3);
printf("T = %.2f x 10^-8 s  \n",T*10^8);
printf("H = %.1f  mm",H*10^3);
```

Scilab code Exa 13.7 Helical path of an electron

```
clc();
clear;
//Given :
m = 9.12*10^-31;// electron mass in kg
e = 1.6*10^-19;// electron charge in C
u = 5*10^7; // electron speed in m/s
alpha = 30; // angle in degrees
d = 0.5; // diameter in m
//(a)
//helix radius = (m*u*sin(alpha))/B*e
r = d/2; // radius in m
B = (m*u*sind(alpha))/(r*e); // magnetic flux density in Wb/m^2
//(b)
T = (2*%pi*m)/(B*e);// time in s
//(c)
p = T*u*cosd(alpha); // pitch in m
printf("B = %.2f x 10^-3 Wb/m^2 \n",B*10^3);
printf("T = %.2f x 10^-8 s  \n",T*10^8);
printf("p = %.2f m",p);
```

Scilab code Exa 13.9 Electrons orbit in magnetic field

```
clc();
clear;
//Given :
m = 9.109*10^-31;// eletcron mass in kg
e = 1.6*10^-19; // electron charge in C
//T = (2*pi*m)/(B*e) , here B is not given
T = (2*%pi*m)/e;// time in s
printf("T = %.2f x 10^-11 / B ",T*10^11);
```

Scilab code Exa 13.11 Angle of refraction

```
clc();
clear;
//Given :
V1 = 250; // potential in V
V2 = 500;// potential in V
theta1 = 45;// angle in degrees
//Law of electron refraction = sin(theta1)/sin(
   theta2) = (V2/V1)^0.5
theta2 = asind(((V1/V2)^(1/2))*sind(45));
printf("theta2 = %d degrees",theta2);
```

Scilab code Exa 13.12 Bainbridge mass spectograph

```
clc();
clear;
//Given :
M1 = 20; // neon isotope mass in amu
M2 = 22;//neon isotope mass in amu
E = 7*10^4; // Electric field in V/m
e = 1.6*10^-19;// electron charge in C
B = 0.5;// Magnetic field in Wb/m^2
B1 = 0.75; // Magnetic field in Wb/m^2
// Linear seperation = S2 - S1 = (2*E*(M2-M1))/(B*
   B1*e)
// 1 amu = 1.66*10^-27 kg
```

```
S2_S1 =  (2*E*(M2-M1)*1.66*10^-27)/(B*B1*e) ; //
   linear seperation in m
printf("S2-S1 =  %.0f mm",S2_S1*10^3);
```

Scilab code Exa 13.13 Deuteron motion in a cyclotron

```
clc();
clear;
//Given:
m = 2.01*1.66*10^-27; // deuteron mass in kg
q = 1.6*10^-19; // deuteron charge in C
//We know , 1/2(m*v^2) = q*V
//for a 5 MeV deuteron
// 1 MeV = 10^6*1.6*10^-19 J
v =((2*5*10^6*1.6*10^-19)/m)^(1/2) ; // velocity in
   m/s
//(a)
R = 15; // inches
//1 inch = 2.54*10^-2 m
B = (m*v)/(q*R*2.54*10^-2);// magnetic field
   intensity in Wb/m^2
//(b)
f = (q*B)/(2*%pi*m); // frequency in Hz
//(c)
t = 50/f; // time in s
printf("B = %.1f Wb/m^2  \n",B);
printf("f = %.2f MHz  \n",f*10^-6);
printf("t = %.2f  mu s ",t*10^6);
```

Chapter 14

Lasers

Scilab code Exa 14.2 Thermal pumping

```
clc();
clear;
//Given :
lambda = 6000; //wavelength in A
E2_E1 = 12422/lambda; // energy in eV
k = 8.62*10^-5; // in eV/K
T = 300; // Temperature in K
//Equilibrium ratio = N2/N1 =  exp[-(E2-E1)/k*T]
//(a)
Ratio = exp(-E2_E1/(k*T));
//(b)
T1 = (E2_E1)/(k*log(2)); // Temperature in K
printf("Ratio =  %.2f x 10^-35 \n",Ratio*10^35);
printf("T = %d K",T1);
//Resuts obtained differ from those in texbook, because approximate value of k*T was considered
```

Scilab code Exa 14.3 Calculating wavelength difference

```
clc();
clear;
//Given :
L =8;// in cm
lambda = 5330; //wavelength in A
// lambda = 2*L/n
// 1 A = 1.0*10^-8 cm
n= (2*L)/(lambda*10^-8); // allowed modes
//adjacent mode
n1 = round(n+1);
// 1 cm = 1.0*10^8 A
lambda1 = ((2*L)/n1)*10^8;// wavelength in A
D = lambda-lambda1; // difference in wavelengths in A
printf("Difference = %.3f A",D);
```

Scilab code Exa 14.5 deltalambda by lambda

```
clc();
clear;
//Given :
tau_c = 10^-5; // lifetime of lasing energy in s
tau_c1 = 10^-8; // coherence time in s
lambda = 5000; // wavelength in A
c = 3*10^8;// light speed in m/s
// Ratio = delta_lambda/lambda   = lambda/(c*tau_c)
// 1 A = 1.0*10^-10 m
//(a)Laser source
Ratio = (lambda*10^-10)/(c*tau_c);
//(b)Ordinary source
Ratio1 = (lambda*10^-10)/(c*tau_c1);
```

```
printf("Laser source = %.2f x 10^-10 \n",Ratio*10^10);
printf("Ordinary source = %.2f x 10^-7 \n",Ratio1*10^7);
//Results obtained differ from those in textbook, beacuse only order of 10 was considered in the result.
```

Scilab code Exa 14.6 Intensity of a laser beam

```
clc();
clear;
//Given :
P = 10; // Power in W
lambda =5000; // wavelength in A
SI = 7*10^3; // Sun's radiation intensity in W/cm^2
// 1 A = 1.0*10^-8 cm
I = P/(lambda*10^-8)^2; //Intensity in W/cm^2
Ratio = (I)/SI;
printf("Intensity = %.0f x 10^6 kW/cm^2 \n",I*10^-9);
printf("Intensity of this laser source is %.1f x 10^6 times the intensity of Sun radiation",Ratio*10^-6);
//Textbook : Only order of 10 is considered in the result
```

Scilab code Exa 14.7 Information capacity of laser

```
clc();
clear;
//Given :
c = 3*10^8;// light speed in m/s
//Visible range = 4000 A - 7000 A
lambda1 = 4000; // wavelength in A
lambda2 = 7000;// wavelength in A
// 1 A = 1.0*10^-10 m
nu1 = c/(lambda1*10^-10); // frequency in Hz
nu2 = c/(lambda2*10^-10);// frequency in Hz
deltanu = nu1-nu2; // in Hz
//(a)Telephone conversations
f1 = 10^3; // frequency in Hz
n1 = deltanu/f1;
//(b)Television programmes
f2 = 10^7; // frequency in Hz
n2 = deltanu/f2;
printf(" Number of Telephone conversations = %.1f x
    10^11 \n",n1*10^-11);
printf(" Number of Television programmes  = %.1f x
    10^7 \n",n2*10^-7);
```

Chapter 15

Fibre Optics

Scilab code Exa 15.1 Importance of cladding material

```
clc();
clear;
//Given :
n0 = 1;//refractive index of outer medium
n1 = 1.5025; // refractive index of core
n2 = 1.4975; // refractive index of cladding
NA = sqrt(n1^2 - n2^2); // Numerical aperture with
   cladding
alpha_c = asind(NA/n0); // acceptance angle in
   degrees
NA1 = sqrt(n1^2 - n0^2);// Numerical aperture
   without cladding
printf("With cladding , NA  and Acceptance angle = %
   .4f  and %.3f degrees \n ",NA,alpha_c);
printf("Without cladding , NA = %.4f ",NA1);
```

Scilab code Exa 15.2 Number of reflections

```
clc();
clear;
//Given :
n1 = 1.5025;// refractive index of core
delta = 0.0033; //
a = 50; // core radius in mu_m
Ls = a*sqrt(2/delta);// skip distance in mu_m
// 1 mu_m = 1.0*10^-6 m
R = 1/(Ls*10^-6);// reflections per m
printf("Ls = %.1f mu_m \n",Ls);
printf("Reflections per m = %d",R);
```

Scilab code Exa 15.3 Determining Limiting Diameter

```
clc();
clear;
//Given :
lambda = 1.25; // wavelength in mu_m
n1 = 1.462; // refractive index of core
n2 = 1.457; // refractive index of cladding
// Single mode propogation : (2*pi*a*sqrt(n1^2 - n2^2))/lambda < 2.405
a  = (2.405*lambda)/(2*%pi*sqrt(n1^2 - n2^2)); // radius in mu_m
d = a*2; // diameter in mu_m
printf("Limiting diameter = %.2f mu_m",d);
```

Scilab code Exa 15.4 Calculation of attenuation

```
clc();
clear;
//Given :
n1 = 1.525; // refractive index of core
n2 = 1.500; // refractive index of cladding
d = 30; // core diameter in mu_m
a = d/2; // core radius in mu_m
ab = 0.00001/100; // percentage absorbed
delta = (n1-n2)/n1;
Ls = a*sqrt(2/delta);// skip distance in mu_m
//1 mu_m = 1.0*10^-6 m
R = 1000/(Ls*10^-6); // reflections per km (1000 m)
red_p = 1 - ab; // reduced power for each reflection
//Power P1km = P0*red_p^(6*10^6)
// A = 10*log10[P0/P1km] , P0 in the numerator and denominator will cancel each other
A = 10*log10(1/(red_p)^(R));
printf("Attenuation = %.1f dB/km",A);
```

Scilab code Exa 15.5 Calculation of maximum delay

```
clc();
clear;
//Given :
n1 = 1.5025; // refractive index of core
n2 = 1.4975; // refractive index of cladding
L = 1; // length in m
F = 2*10^6; // frequency in Hz
c = 3*10^8;// light speed in m/s
delta_t = (n1*L/c)*((n1/n2)-1);// maximum delay in s;
f = 1/(2*delta_t); // bandwidth for 1 m propogation
L1 = 1/(2*F*delta_t); // distance for 2MHz bandwidth
```

```
printf("Maximum delay = %.1f ps \n",delta_t*10^12);
printf("Bandwidth of 2MHz can propogate a distance
    of %.1f  km ",L1*10^-3);
```

Chapter 16

Acoustics

Scilab code Exa 16.1 Increase in Sound velocity

```
clc();
clear;
//Given :
delta_t = 1; // temperature in degrees
t1 = 27; // temperature in degrees
//Ratio = v2/v1 = 1+ (delta_t/(t1+273))
Ratio = 1 + (delta_t /(2*(t1+273)));
v1 = 343;// speed of sound at room temperature in m/
    s
v2 = v1*Ratio; // speed of sound in air in m/s
delta_v = v2-v1; // speed in m/s
printf("Ratio = %.4f  \n",Ratio);
printf("delta_v = %.1f m/s",delta_v);
```

Scilab code Exa 16.2 Limits of displacement amplitudes

```
clc();
clear;
//Given :
p_rms = 0.0002; // in microbar
p_rms1 = 20; // in pascal
v = 343; // speed of sound in m/s
rho_0 = 1.21; // density of air in kg/m^3
f = 1000; // frequency in Hz
// p_rms = pm_min/(2)^0.5
//1 microbar = 0.1 N/m^2
pm_min = sqrt(2)*p_rms*0.1; //in N/m^2
// 1 pascal = 1 N/m^2
pm_max =sqrt(2)*p_rms1*1; // in N/m^2
// sm = pm/(v*rho_0*omega);
//omega = 2*pi*f
sm_min = pm_min/(v*rho_0*2*%pi*f); // displacement amplitude in m
sm_max = pm_max/(v*rho_0*2*%pi*f);// displacement amplitude in m
printf("Minimum displacement amplitude = %.2f  pm \n",sm_min*10^12);
printf("Maximum displacement amplitude = %.0f mu m",sm_max*10^6);
```

Scilab code Exa 16.3 Imax by Imin

```
clc();
clear;
//Given :
sm_min = 11*10^-12;// Minimum displacement amplitude in m
sm_max = 11*10^-6;// Maximum displacement amplitude in m
```

```
v = 343;// speed of sound in m/s
f = 1000; // frequency in Hz
rho_0 = 1.21; // density of air in kg/m^3
// Sound intensity = (rho_0*v*omega^2*sm^2)/2
//omega = 2*pi*f
I_max = (rho_0*v*((2*%pi*f)^2)*(sm_max^2))/2; //
   Maximum Intensity
I_min =  (rho_0*v*((2*%pi*f)^2)*(sm_min^2))/2; //
   Minimum Intensity
Ratio = I_max/I_min ;
printf("I_max/I_min = %.1f x 10^12 ", Ratio*10^-12);
```

Scilab code Exa 16.4 Sound intensities

```
clc();
clear;
//Given :
I0 = 10^-12; // in W/m^2
beta1 = 0; // in dB
beta2 = 60;// in dB
beta3 = 120; // in dB
// Intensity level = beta = 10*log10(I/I0)
I1 = 10^(beta1/10)*I0; // Intensity in W/m^2
I2 = 10^(beta2/10)*I0; // Intensity in W/m^2
I3 = 10^(beta3/10)*I0; // Intensity in W/m^2
printf("Hearing Threshold : %.1f x 10^-12 W/m^2 \n",
   I1*10^12);
printf("Speech Activity : %.1f x 10^-6 W/m^2 \n",I2
   *10^6);
printf("Pain Threshold : %.1f W/m^2",I3);
```

Scilab code Exa 16.5 Determination of reverberation time

```
clc();
clear;
//Given :
l = 200; // in ft
b = 50; // in ft
h = 30;// in ft
alpha = 0.25; //average absorption coefficient
V  = l*b*h; // Volume in ft^3
S = 2*((l*b)+(l*h)+(b*h)); //total surface area in
   ft^2
a = alpha*S;// in sabins
T = (0.049*V)/a; // reverberation time in s
//400 people present in the auditorium, 1 person is
   equivalent to 4.5 sabins
a1 = a+ 400*4.5; // in sabins
T1 = (0.049*V)/a1;// reverberation time in s
printf("For auditorium : %.2f s \n",T);
printf("When people are present %.2f s",T1);
```

Scilab code Exa 16.6 Determination of unknown absorption coefficient

```
clc();
clear;
//Given :
V = 9*10*11; // Volume in ft^3
T = 4; // reverberation time in s
```

```
S = 2*((9*10)+(10*11)+(11*9));// total surface area in ft^2
//T = (0.049*V)/(alpha*S)
alpha = (0.049*V)/(S*T);//average absorption coefficient
T1 = 1.3; // reverberation time in s
S1 = 50; // total surface area in ft^2
alpha_e =(((0.049*V)/S1)*((1/T1)-(1/T))) + alpha ; // effective absorption coefficient
printf("alpha = %.2f  \n",alpha);
printf("alpha_e = %.2f  ",alpha_e);
```

Scilab code Exa 16.7 Use of ultrasound by bats

```
clc();
clear;
//Given :
v = 343; // velocity of sound in m/s
lambda = 1; // wavelength in cm
// 1 cm = 1.0*10^-2 m
f = v/(lambda*10^-2); //frequency in Hz
printf("Frequency is %.1f kHz",f*10^-3);
```

Scilab code Exa 16.8 Ultrasonic generators

```
clc();
clear;
//Given :
E1 = 8.55*10^10; //Modulus of elasticity in N/m^2
```

```
E2 = 21*10^10; // Modulus of elasticity in N/m^2
rho1 = 2650; // density of Quartz in kg/m^3
rho2 = 8800;// density of Nickel in kg/m^3
t = 2; // thickness of crystal in mm
l = 50; // rod length in mm
//Piezoelectric generator
printf("Piezoelectric generator \n\n");
for n = 1:3
    // 1 mm = 1.0*10^-3 m
    nu1 = (n/(2*t*10^-3))*sqrt(E1/rho1);// frequency in Hz
    printf("For  n = %d  , Frequency  = %.2f  MHz\n",n,nu1*10^-6);
end
//Magnetostriction generator
printf("Magnetostriction generator\n\n");
for n1 = 1:3
     // 1 mm = 1.0*10^-3 m
    nu2 = (n1/(2*l*10^-3))*sqrt(E2/rho2);// frequency in Hz
    printf("For  n = %d  , Frequency  = %.1f  kHz\n",n1,nu2*10^-3);
end
//Results differ from those in textbook, because in the formulae (n/(2*t))*sqrt(E/rho) and (n/(2*l))*sqrt(E/rho) , 2 is not multiplied with either t or l.
```

Scilab code Exa 16.9 Noise pollution

```
clc();
clear;
//Given :
```

```
I0 = 10^-12; // in W/m^2
beta1 = 110; // in dB
beta2 = 150;// in dB
beta3 = 180; // in dB
// Intensity level = beta = 10*log10(I/I0)
I1 = 10^(beta1/10)*I0; // Intensity in W/m^2
I2 = 10^(beta2/10)*I0; // Intensity in W/m^2
I3 = 10^(beta3/10)*I0; // Intensity in W/m^2
printf("Amplified Rock Music : %.2f W/m^2 \n",I1);
printf("Jet plane  : %.1f x 10^3 W/m^2 \n",I2*10^-3);
printf("Rocket engine : %.1f x 10^6 W/m^2",I3*10^-6);
```

Scilab code Exa 16.10 Determination of sea depth

```
clc();
clear;
//Given :
v = 1500; // velocity of ultrasound in m/s
rt = 0.8; // recorded time in s
t = rt/2; // time in s
//Ultrasound velocity = D/t
D = v*t; // sea depth in m
printf("Depth = %d m",D);
```