Scilab Textbook Companion for Applied Physics by K. Kumar, T. Sreekanth[1]

Created by
Lalit Kumar
B. Tech
Electronics Engineering
COER
College Teacher
None
Cross-Checked by
None

October 16, 2016

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

Book Description

Title: Applied Physics

Author: K. Kumar, T. Sreekanth

Publisher: S. Chand & Company Ltd, New Delhi

Edition: 1

Year: 2005

ISBN: 8121926904

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

BONDING IN SOLIDS

Scilab code Exa 1.1 Bond energy

```
1 //Example number 1.1, Page number 10
2 clc;clear;
3 close;
4
5 //Variable declaration
6 e=1.6*10**-19;       //charge(coulomb)
7 epsilon0=8.85*10**-12;
8 r0=23.6*10**-10;     //equilibrium distance(m)
9 I=5.14;     //ionisation energy(eV)
10 EA=3.65;    //electron affinity(eV)
11 N=8;      //born constant
12
13 //Calculation
14 x=1-(1/N);
15 V=(e**2)*x/(4*e*%pi*epsilon0*r0);    //potential(V)
16 E=I-EA;          //net energy(eV)
17 BE=(V*10)-E;          //bond energy(eV)
18
19 //Result
20 printf( "bond energy = %.2f eV",BE)
```

Scilab code Exa 1.2 Compressibility

```
1 //Example number 1.2, Page number 10
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;       //charge(coulomb)
8 epsilon0=8.85*10**-12;
9 r0=0.41*10**-3;     //equilibrium distance(m)
10 A=1.76;        //madelung constant
11 n=0.5;         //repulsive exponent value
12
13 //Calculation
14 Beta=72*%pi*epsilon0*r0**4/(A*e**2*(n-1));     //
   compressibility
15
16 //Result
17 printf( "compressibility = %.4e",Beta)
18 //answer in the book is wrong
```

Scilab code Exa 1.3 cohesive energy

```
1 //Example number 1.3, Page number 10
2 clc;clear;
3 close;
4
5 //Variable declaration
6 e=1.6*10**-19;       //charge(coulomb)
7 epsilon0=8.85*10**-12;
8 r0=0.314*10**-9;     //equilibrium distance(m)
```

```
A=1.75;        //madelung constant
N=5.77;       //born constant
I=4.1;     //ionisation energy(eV)
EA=3.6;    //electron affinity(eV)

//Calculation
V=-A*e**2*((N-1)/N)/(4*e*%pi*epsilon0*r0);
PE=V/2;     //potential energy per ion(eV)
x=(I-EA)/2;
CE=PE+x;    //cohesive energy(eV)

//Result
printf( "cohesive energy is = %.3f eV",CE)
```

Scilab code Exa 1.4 binding energy

```
//Example number 1.4, Page number 11
clc;clear;
close;

//Variable declaration
N=6.02*10**26;              //Avagadro Number
e=1.6*10**-19;        //charge(coulomb)
epsilon0=8.85*10**-12;
r0=0.324*10**-9;     //equilibrium distance(m)
A=1.75;       //madelung constant
n=8.5;        //repulsive exponent value

//Calculations
U0=(A*e/(4*%pi*epsilon0*r0))*(1-1/n);
U=U0*N*e/10**3;       //binding energy(kJ/kmol)

//Result
printf( "binding energy is %.1e kJ/mol",U)
//answer in the book is wrong
```

Scilab code Exa 1.5 Density of CsCl

```
1 //Example number 1.5, Page number 11
2 clc;clear;
3 close;
4
5 //Variable declaration
6 rCs=0.165*10**-9;     //radius(m)
7 rCl=0.181*10**-9;     //radius(m)
8 MCs=133;     //atomic weight
9 MCl=35.5;    //atomic weight
10 N=6.02*10**26;            //Avagadro Number
11
12 //Calculation
13 a=2*(rCl+rCs)/sqrt(3);      //lattice constant(m)
14 M=(MCs+MCl)/N;        //mass of 1 molecule(kg)
15 V=a**3;     //volume of unit cell(m**3)
16 rho=M/V;     //density of CsCl(kg/m**3)
17
18 //Result
19 printf( "density of CsCl is %.3e kg/m**3",rho)
20 //answer in the book varies due to rounding off
   errors
```

Scilab code Exa 1.6 Effective charge

```
1 //Example number 1.6, Page number 12
2
3 clc;clear;
4 close;
5
```

```
//Variable declaration
dm=1.98*(10**-29)*(1/3);        //dipole moment
l=0.92*10**-10;          //bond length(m)

//Calculation
ec=dm/l;          //effective charge(coulomb)

//Result
printf( "effective charge is %.1e Coulomb",ec)
//answer given in the book is wrong
```

Scilab code Exa 1.7 Energy required

```
//Example number 1.7, Page number 12

clc;clear;
close;

//Variable declaration
e=1.6*10**-19;        //charge(coulomb)
epsilon0=8.85*10**-12;
r=0.5*10**-9;     //distance(m)
I=5;     //ionisation energy(eV)
E=4;    //electron affinity(eV)

//Calculation
C=e**2/(4*%pi*epsilon0*e*r);     //coulomb energy(eV)
Er=I-E-C;     //energy required(eV)

//Result
printf( "energy required is %.1f eV",Er)
```

Scilab code Exa 1.9 Youngs modulus

```
1  //Example number 1.9, Page number 13
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  a=7.68*10**-29;
8  r0=2.5*10**-10;     //radius(m)
9
10 //Calculation
11 b=a*(r0**8)/9;
12 y=((-2*a*r0**8)+(90*b))/r0**11;
13 E=y/r0/10**9;              //young's modulus(GPa)
14
15 //Result
16 printf( "young''s modulus is %d GPa",E)
```

Chapter 2

CRYSTAL STRUCTURE

Scilab code Exa 2.1 Density

```
1 //Example number 2.1, Page number 31
2
3 clc;clear;
4 close;
5
6 // Variable declaration
7 N=6.02*10**26;              // Avagadro Number
8 n=8;     // number of atoms
9 a=5.6*10**-10;     // lattice constant(m)
10 M=72.59;      // atomic weight(amu)
11
12 // Calculation
13 rho=n*M/(a**3*N);      // density(kg/m**3)
14
15 // Result
16 printf( "density is %.3f kg/m^3",rho)
```

Scilab code Exa 2.2 lattice constant

```
1 //Example number 2.2, Page number 32
2
3 clc;clear;
4 close;
5 // Variable declaration
6 N=6.02*10**23;            // Avagadro Number
7 n=2;
8 rho=7860;     // density(kg/m**3)
9 M=55.85;     // atomic weight(amu)
10
11 // Calculation
12 a=(n*M/(rho*N))**(1/3)*10**8;     // lattice constant
     (angstrom)
13
14 // Result
15 printf( "lattice constant is %.4f Angstrom",a)
```

Scilab code Exa 2.3 lattice constant

```
1 //Example number 2.3, Page number 32
2 clc;clear;
3 close;
4
5 // Variable declaration
6 N=6.02*10**26;            // Avagadro Number
7 n=2;
8 rho=530;     // density(kg/m**3)
9 M=6.94;     // atomic weight(amu)
10
11 // Calculation
12 a=(n*M/(rho*N))**(1/3)*10**10;     // lattice
     constant(angstrom)
13
14 // Result
15 printf( "lattice constant is %.3f Angstrom",a)
```

Scilab code Exa 2.4 number of atoms

```
1 //Example number 2.4, Page number 32
2
3 clc;clear;
4 close;
5
6 // Variable declaration
7 N=6.02*10**26;              // Avagadro Number
8 rho=7870;      // density(kg/m**3)
9 M=55.85;      // atomic weight(amu)
10 a=2.9*10**-10;      // lattice constant(m)
11
12 // Calculation
13 n=a**3*rho*N/M;        // number of atoms
14
15 // Result
16 printf( "number of atoms is %d",n)
```

Scilab code Exa 2.5 Density

```
1 //Example number 2.5, Page number 33
2 clc;clear;
3 close;
4
5 // Variable declaration
6 N=6.02*10**26;              // Avagadro Number
7 M=63.5;      // atomic weight(amu)
8 r=0.1278*10**-9;      // atomic radius(m)
9 n=4;
10
```

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

// Result
printf( "density is %.2f kg/m**3",rho)
//answer in the book is wrong
```

Scilab code Exa 2.6 percent volume change

```
//Example number 2.6, Page number 33

clc;clear;
close;

// Variable declaration
r1=1.258*10**-10;      // radius(m)
r2=1.292*10**-10;      // radius(m)

// Calculation
a_bcc=4*r1/sqrt(3);
v=a_bcc**3;
V1=v/2;
a_fcc=2*sqrt(2)*r2;
V2=a_fcc**3/4;
V=(V1-V2)*100/V1;            // percent volume change
    is",V,"%"

// Result
printf( "percent volume change is %.1f %%",V)
```

Scilab code Exa 2.7 maximum radius of sphere

```
1 //Example number 2.7, Page number 34
2
3 clc;clear;
4 close;
5
6 // Variable declaration
7 r=poly([0],'r')
8
9 // Calculation
10 a=4*r/sqrt(2);
11 R=(4*r/(2*sqrt(2)))-r
12
13 // Result
14 printf( "maximum radius of sphere is ")
15 disp(R)
```

Scilab code Exa 2.8 distance between atoms

```
1 //Example number 2.8, Page number 34
2 clc;clear;
3 close;
4
5 // Variable declaration
6 N=6.023*10**23;              // Avagadro Number
7 Mw=23+35.5;          // molecular weight of NaCl
8 rho=2.18;     // density(gm/cm**3)
9
10 // Calculation
11 M=Mw/N;          // mass of 1 molecule(gm)
12 Nv=rho/M;        // number of molecules per unit
      volume(mole/cm**3)
13 Na=2*Nv;       // number of atoms
14 a=(1/Na)**(1/3)*10**8;    // distance between atoms(
      angstrom)
15
```

```
16 // Result
17 printf( "distance between atoms is %.2f Angstrom",a)
```

Chapter 3

CRYSTAL PLANES AND POINT DEFECTS

Scilab code Exa 3.1 Miller Indices

```
1 //Example number 3.1, Page number 45
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=1;
8 b=1/2;
9 c=3;       //intercepts
10 //Calculation
11 h=int(c/a);
12 k=int(c/b);
13 l=int(c/c);      //smiller indices
14 //Result
15 printf("miller indices are (%d,%d,%d)",h,k,l)
```

Scilab code Exa 3.5 Miller Indices

```
1 //Example number 3.5, Page number 48
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=1;
8 b=2;
9 c=3;       //intercepts
10 //Calculation
11 h=int(c/a);
12 k=int(b);
13 l=int(c*b);      //miller indices
14 //Result
15 printf("miller indices are (%d,%d,%d)",h,k,l)
```

Scilab code Exa 3.7 Miller Indices

```
1 //Example number 3.7, Page number 48
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=poly([0],'a')
8 b=poly([0],'b')
9 X=3;
10 Y=4;
11 Z=0;        //intercepts
12 //Calculation
13 x=a/X;
14 y=b/Y;
15 z=%inf ;       //miller indices
```

```
16 //Result
17 printf("miller indices are : \n")
18 disp (z,y,x)
```

Scilab code Exa 3.8 spacing between planes

```
1 //Example number 3.8, Page number 49
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=0.25;
8 b=0.25;
9 c=0.18;
10 h=1;
11 k=1;
12 l=1;
13 //Calculation
14 d_hkl=1/sqrt((a**2/h**2)+(b**2/k**2)+(c**2/l**2));
       //spacing between planes(nm)
15 //Result
16 printf("spacing between planes is %.3f nm",d_hkl)
17 //answer in the book is wrong
```

Scilab code Exa 3.9 number of atoms

```
1 //Example number 3.9, Page number 49
2 clc;clear;
3 close;
4
5 //Variable declaration
6 h1=1;
```

```
k1=0;
l1=0;     //miller indices of (100)
h2=1;
k2=1;
l2=0;     //miller indices of (110)
a=0.287;     //lattice constant(nm)
//Calculation
d100=a/sqrt(h1**2+k1**2+l1**2);     //spacing(nm)
d110=a/sqrt(h2**2+k2**2+l2**2);     //spacing(nm)
rho=2/(sqrt(2)*(d100*10**-9)**2);     //number of
   atoms(per mm**2)
//Result
printf("number of atoms is %.3E atoms/mm^2",rho)
//answer in the book is wrong
```

Scilab code Exa 3.10 interplanar spacing

```
//Example number 3.10, Page number 49
clc;clear;
close;

//Variable declaration
r=0.1278*10**-9;     //atomic radius(m)
h1=1;
k1=1;
l1=1;
h2=3;
k2=2;
l2=1;
//Calculation
a=2*sqrt(2)*r;
d111=a*10**10/sqrt(h1**2+k1**2+l1**2);     //
   interplanar spacing for (111)
d321=a*10**10/sqrt(h2**2+k2**2+l2**2);     //
   interplanar spacing for (321)
```

```
17 //Result
18 printf("interplanar spacing for (111) is %.3f
      Angstrom",d111)
19 printf("\n interplanar spacing for (321) is %.3f
      Angstrom",d321)
```

Scilab code Exa 3.11 percent volume change

```
1  //Example number 3.11, Page number 50
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  r1=1.258*10**-10;      //radius(m)
8  r2=1.292*10**-10;      //radius(m)
9  //Calculation
10 a_bcc=4*r1/sqrt(3);
11 v=a_bcc**3;
12 V1=v/2;
13 a_fcc=2*sqrt(2)*r2;
14 V2=a_fcc**3/4;
15 V=(V1-V2)*100/V1;            //percent volume change
      is",V,"%"
16 //Result
17 printf("percent volume change is %.1f %%",V)
```

Scilab code Exa 3.12 volume of cell

```
1  //Example number 3.12, Page number 50
2  clc;clear;
3  close;
4
```

```
//Variable declaration
C=0.494*10**-9;       //height(m)
a=0.27*10**-9;       //distance(m)
M=65.37;       //atomic weight
N=6.02*10**26;     //avagadro number
//Calculation
V=3*sqrt(3)*a**2*C/2;       //volume of cell(m**3)
m=6*M/N;
rho=m/V;       //density of Zn(kg/m**3)
//Result
printf("volume of cell is %.3e m**3",V)
printf("\n density of Zn is %.1f kg/m**3",rho)
//answer in the book is wrong
```

Scilab code Exa 3.13 fraction of vacancy

```
//Example number 3.13, Page number 51

clc;clear;
close;

//Variable declaration
T1=773;     //temperature(K)
T2=1273;    //temperature(K)
n=1*10**-10;     //fraction of vacancy sites
//Calculation
logx=T1*log(n)/T2
x=%e**(logx);           //fraction of vacancy sites
//Result
printf("fraction of vacancy sites is %.3e",x)
//answer in the book varies due to rounding off
   errors
```

Scilab code Exa 3.14 Ratio of number of vacancies

```
1 //Example number 3.14, Page number 51
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 Ev=68*10**3;     //enthalpy(j/mol)
8 R=8.314;
9 T1=300;     //temperature(K)
10 T2=800;    //temperature(K)
11 //Calculation
12 x1=-Ev/(R*T1);
13 x2=-Ev/(R*T2);
14 n=%e**(x1)/%e**(x2);       //ratio of number of
      vacancies
15 //Result
16 printf("ratio of number of vacancies is %.2e",n)
17 //answer in the book varies due to rounding off
      errors
```

Scilab code Exa 3.15 value of concentration

```
1 //Example number 3.15, Page number 52
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 KbT=0.025;
8 nbyN=1/10**10;     //concentration
9 N=10**29;
10 //Calculation
11 x=2*KbT;
```

```
Ev=x*log(1/nbyN);      //value of concentration(eV)
n=1/((N*nbyN)**(1/3));      //average seperation(m)
//Result
printf("value of concentration is %.1f eV",Ev)
printf("\n average seperation is %.2e m",n)
```

Scilab code Exa 3.16 Energy required

```
//Example number 3.16, Page number 52

clc;clear;
close;

//Variable declaration
N=2.303*16.65;
T=298;     //temperature(K)
Kb=8.625*10**-5;
//Calculation
E=2*N*Kb*T;       //energy required(eV)
//Result
printf("energy required is %.2f eV",E)
```

Chapter 4

DISLOCATIONS AND CRYSTAL STRUCTURE DETERMINATION

Scilab code Exa 4.1 maximum order of diffraction

```
1 //Example number 4.1, Page number 66
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 d=0.282*10**-9;     //lattice spacing(m)
8 theta=8+(35/60);    //glancing angle(degree)
9 n=1;    //order
10 Theta=90;     //angle(degree)
11 //Calculation
12 theta=theta*%pi/180;     //angle(radian)
13 Theta=Theta*%pi/180;     //angle(radian)
14 lamda=2*d*sin(theta)/n;     //wavelength(m)
15 nmax=2*d*sin(Theta)/lamda;     //maximum order of
    diffraction
16 //Result
```

```
17 printf("wavelength is %.3f Angstrom",lamda*10**10)
18 //answer varies due to rounding off errors
19 printf("\n maximum order of diffraction is %d",round
   (nmax))
```

Scilab code Exa 4.2 glancing angle

```
1 //Example number 4.2, Page number 66
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 d=3.04*10**-10;     //lattice spacing(m)
8 n=3;    //order
9 lamda=0.79*10**-10;     //wavelength(m)
10 //Calculation
11 theta=asin(n*lamda/(2*d));      //glancing angle(
   radian)
12 theta=theta*180/%pi;           //glancing angle(
   degrees)
13 //Result
14 printf("glancing angle is %.3f degree",theta)
```

Scilab code Exa 4.3 glancing angle

```
1 //Example number 4.3, Page number 66
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=0.28*10**-9;     //lattice spacing(m)
```

```
n=2;    //order
lamda=0.071*10**-9;     //wavelength(m)
h=1;
k=1;
l=0;
//Calculation
d110=a/sqrt(h**2+k**2+l**2);      //spacing(m)
theta=asin(n*lamda/(2*d110));     //glancing angle(
   radian)
theta=theta*180/%pi;           //glancing angle(
   degrees)
//Result
printf("glancing angle is %.2f degree",theta)
//answer in the book is wrong
```

Scilab code Exa 4.4 space of plane

```
//Example number 4.4, Page number 67

clc;clear;
close;

//Variable declaration
n=1;    //order
lamda=3*10**-10;     //wavelength(m)
h=1;
k=0;
l=0;
theta=40;     //angle(degree)
//Calculation
theta=theta*%pi/180;     //angle(radian)
d=n*lamda/(2*sin(theta));      //space of plane(m)
a=d*sqrt(h**2+k**2+l**2);
V=a**3;       //volume of unit cell(m**3)
//Result
```

```
19 printf("space of plane is %.4f Angstrom",d*10**10)
20 printf("\n volume of unit cell is %.3e m**3",V)
```

Scilab code Exa 4.5 Spacing

```
1 //Example number 4.5, Page number 67
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=3;     //lattice spacing(m)
8 n=1;    //order
9 lamda=0.82*10**-9;     //wavelength(m)
10 theta=75.86;     //angle(degree)
11 //Calculation
12 theta=theta*%pi/180;     //angle(radian)
13 d=n*10**10*lamda/(2*sin(theta));     //spacing(
   angstrom)
14 //Result
15 printf("spacing is %.2f Angstrom",d)
16 //answer in the book is wrong. hence the miller
   indices given in the book are also wrong.
```

Scilab code Exa 4.6 interplanar spacing

```
1 //Example number 4.6, Page number 68
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;     //charge(c)
```

```
m=9.1*10**-31;     //mass(kg)
h=6.625*10**-34;    //plank constant
n=1;    //order
theta=9+(12/60)+(25/(60*60));     //angle(degree)
V=235.2;     //kinetic energy of electron(eV)
//Calculation
theta=theta*%pi/180;     //angle(radian)
lamda=h*10**10/sqrt(2*m*e*V);
d=n*lamda/(2*sin(theta));        //interplanar
   spacing(angstrom)
//Result
printf("interplanar spacing is %.3f Angstrom",d)
//answer in the book is wrong
```

Scilab code Exa 4.7 wavelength of X ray

```
//Example number 4.7, Page number 68

clc;clear;
close;

//Variable declaration
n=1;    //order
h=1;
k=1;
l=1;
e=1.6*10**-19;     //charge(c)
theta=27.5;     //angle(degree)
H=6.625*10**-34;     //plancks constant
c=3*10**10;     //velocity of light(m)
a=5.63*10**-10;     //lattice constant(m)
//Calculation
theta=theta*%pi/180;     //angle(radian)
d=a/sqrt(h**2+k**2+l**2);
lamda=2*d*sin(theta)/n;       //wavelength of Xray
```

```
    beam(m)
E=H*c/(e*lamda);              //energy of Xray beam(eV)
//Result
printf("wavelength of X-ray beam is %.f Angstrom",
    int32(lamda*10**10))
printf("\n energy of Xray beam is %.2e eV",E)
//answer in the book is wrong
```

Scilab code Exa 4.8 spacing of crystal

```
//Example number 4.8, Page number 69

clc;clear;
close;

//Variable declaration
e=1.6*10**-19;     //charge(c)
theta=56;     //angle(degree)
V=854;     //voltage(V)
n=1;       //order of diffraction
m=9.1*10**-31;     //mass(kg)
h=6.625*10**-34;    //plank constant
//Calculation
theta=theta*%pi/180;     //angle(radian)
lamda=h/sqrt(2*m*e*V);     //wavelength(m)
d=n*lamda/(2*sin(theta))*10**10;      //spacing of
    crystal(Angstrom)
//Result
printf("spacing of crystal is %.3f Angstrom",d)
```

Scilab code Exa 4.9 lattice parameter

```
//Example number 4.9, Page number 69
```

```
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n=1;    //order
8 h=2;
9 k=0;
10 l=2;
11 theta=34;     //angle(degree)
12 lamda=1.5;    //wavelength(angstrom)
13 //Calculation
14 theta=theta*%pi/180;     //angle(radian)
15 d=n*lamda/(2*sin(theta));      //spacing of crystal(
      angstrom)
16 a=d*sqrt(h**2+k**2+l**2);      //lattice parameter(
      angstrom)
17 //Result
18 printf("lattice parameter is %.3f Anstrom",a)
19 //answer in the book is wrong
```

Scilab code Exa 4.10 braggs angle

```
1 //Example number 4.10, Page number 70
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n=1;    //order
8 h=1;
9 k=1;
10 l=1;
11 e=1.6*10**-19;     //charge(c)
12 V=5000;     //voltage(V)
```

```
m=9.1*10**-31;     //mass(kg)
H=6.625*10**-34;    //plank constant
d=0.204*10**-9;     //interplanar spacing(m)
//Calculation
lamda=H/sqrt(2*m*e*V);     //wavelength(m)
theta=asin(n*lamda/(2*d));     //bragg's angle(radian
   )
theta=theta*180/%pi;     //bragg's angle(degree)
//Result
printf("bragg''s angle is %.4f degree",theta)
```

Chapter 5

PRINCIPLES OF QUANTUM MECHANICS

Scilab code Exa 5.1 wavelength

```
1 //Example number 5.1, Page number 85
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;     //charge(c)
8 m=9.1*10**-31;     //mass(kg)
9 h=6.626*10**-34;    //plank constant
10 E=2000;            //energy(eV)
11 //Calculation
12 lamda=h/sqrt(2*m*E*e)*10**9;     //wavelength(nm)
13 //Result
14 printf("wavelength is %.4f nm",lamda)
```

Scilab code Exa 5.2 Wavelength

```
1 //Example number 5.2, Page number 85
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 V=1600;       //potential energy of electron(V)
8 //Calculation
9 lamda=12.27/sqrt(V);       //wavelength(m)
10 //Result
11 printf("wavelength is %f Angstrom",lamda)
12 //answer given in the book is wrong
```

Scilab code Exa 5.3 de broglie wavelength

```
1 //Example number 5.3, Page number 85
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 me=9.1*10**-31;     //mass(kg)
8 h=6.62*10**-34;    //plank constant
9 mn=1.676*10**-27;     //mass(kg)
10 c=3*10**8;      //velocity of light(m/s)
11 //Calculation
12 lamda=h*10**10/sqrt(4*mn*me*c**2);      //de broglie
   wavelength(angstrom)
13 //Result
14 printf("de broglie wavelength is %.1e Angstrom",
   lamda)
```

Scilab code Exa 5.4 energy of second state

```
1 //Example number 5.4, Page number 85
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=2*10**-10;     //length(m)
8 n1=2;
9 n2=4;
10 m=9.1*10**-31;     //mass(kg)
11 e=1.6*10**-19;     //charge(c)
12 h=6.626*10**-34;    //plank constant
13 //Calculation
14 E2=n1**2*h/(8*m*e*a);       //energy of second state(
    eV)
15 E4=n2**2*h/(8*m*e*a);       //energy of fourth state(
    eV)
16 //Result
17 printf("energy of second state is %.5e eV",E2)
18 printf("\n energy of second state is %.5e eV",E4)
```

Scilab code Exa 5.5 spacing of crystal

```
1 //Example number 5.5, Page number 86
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 V=344;     //accelerated voltage(V)
8 n=1;
9 theta=60;    //glancing angle(degrees)
10 //Calculation
11 theta=theta*%pi/180;     //glancing angle(radian)
12 lamda=12.27/sqrt(V);
```

```
13 d=n*lamda/(2*sin(theta));     //spacing of crystal(
      angstrom)
14 //Result
15 printf("spacing of crystal is %.4f Angstrom",d)
```

Scilab code Exa 5.6 kinetic energy

```
1 //Example number 5.6, Page number 86
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 lamda=1.66*10**-10;     //wavelength(m)
8 m=9.1*10**-32;     //mass(kg)
9 e=1.6*10**-19;     //charge(c)
10 h=6.626*10**-34;    //plank constant
11 //Calculation
12 E=h**2/(4*m*e*lamda**2);    //kinetic energy(eV)
13 v=h/(m*lamda);       //velocity(m/s)
14 //Result
15 printf("kinetic energy is %.2f eV",E)
16 printf("\n velocity is %.2e m/s",v)
```

Scilab code Exa 5.7 ground state energy

```
1 //Example number 5.7, Page number 87
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 a=1*10**-10;     //length(m)
```

```
n2=2;
n3=3;
m=9.1*10**-31;      //mass(kg)
e=1.6*10**-19;      //charge(c)
h=6.626*10**-34;    //plank constant
//Calculation
E1=h**2/(8*m*e*a**2);
E2=n2**2*E1;        //energy of 1st excited state(eV)
E3=n3**2*E1;        //energy of 2nd excited state(eV)
//Result
printf("ground state energy is %.2f eV",E1)
printf("\n energy of 1st excited state is %.2f eV",
   E2)
printf("\n energy of 2nd excited state is %.2f eV",
   E3)
//answer in the book varies due to rounding off
   errors
```

Scilab code Exa 5.8 maximum energy

```
//Example number 5.8, Page number 88

clc;clear;
close;

//Variable declaration
n=poly([0],'n');
a=4*10**-10;     //width of potential well(m)
m=9.1*10**-31;      //mass(kg)
e=1.6*10**-19;      //charge(c)
h=6.626*10**-34;    //plank constant
//Calculation
E1=n**2*h**2/(8*m*e*a**2);      //maximum energy(eV)
//Result
disp(E1,"maximum energy in eV is")
```

Scilab code Exa 5.10 uncertainity in velocity

```
1 //Example number 5.10, Page number 88
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 delta_x=10**-8;      //length of box(m)
8 m=9.1*10**-31;     //mass(kg)
9 h=6.626*10**-34;    //plank constant
10 //Calculation
11 delta_v=h/(m*delta_x)/10**3;      //uncertainity in
      velocity(km/s)
12 //Result
13 printf("uncertainity in velocity is %.1f km/s",
      delta_v)
```

Scilab code Exa 5.11 de broglie wavelength

```
1 //Example number 5.11, Page number 89
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 me=9.1*10**-31;     //mass(kg)
8 mp=1.6*10**-27;     //mass(kg)
9 h=6.626*10**-34;    //plank constant
10 c=3*10**10;     //velocity of light(m/s)
11 //Calculation
```

```
12 lamda=h/sqrt(2*mp*me*c**2)*10**10;     //de broglie
   wavelength(m)
13 //Result
14 printf("de broglie wavelength is %.5e Angstrom",
   lamda)
```

Scilab code Exa 5.12 glancing angle

```
1 //Example number 5.12, Page number 89
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 m=1.675*10**-27;     //mass(kg)
8 h=6.626*10**-34;    //plank constant
9 E=0.04;      //kinetic energy(eV)
10 e=1.6*10**-19;     //charge(c)
11 n=1;
12 d110=0.314*10**-9;    //spacing(m)
13 //Calculation
14 E=E*e;        //energy(J)
15 lamda=h/sqrt(2*m*E);
16 theta=asin(n*lamda/(2*d110));      //glancing angle(
   radian)
17 theta=theta*180/%pi;          //glancing angle(degrees
   )
18 theta_m=60*(theta-int(theta));
19 //Result
20 printf("glancing angle is %d degree and %d minutes",
   theta,theta_m)
21 //answer given in the book is wrong
```

Chapter 6

ELECTRON THEORY OF METALS

Scilab code Exa 6.1 relaxation time

```
1 //Example number 6.1, Page number 116
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 rho=1.54*10**-8;     //resistivity(ohm m)
8 n=5.8*10**28;        //conduction electrons(per m**3)
9 e=1.6*10**-19;       //charge(c)
10 m=9.1*10**-31;       //mass(kg)
11 //Calculation
12 towr=m/(n*e**2*rho);      //relaxation time(sec)
13 //Result
14 printf("relaxation time is %.4e sec",towr)
```

Scilab code Exa 6.2 mean free path

```
1 //Example number 6.2, Page number 116
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 T=300;     //temperature(K)
8 n=8.5*10**28;     //density(per m**3)
9 rho=1.69*10**-8;    //resistivity(ohm/m**3)
10 e=1.6*10**-19;       //charge(c)
11 m=9.11*10**-31;       //mass(kg)
12 Kb=1.38*10**-23;       //boltzmann constant(J/k)
13 //Calculation
14 rho=sqrt(3*Kb*m*T)/(n*e**2*rho);      //mean free
    path(m)
15 //Result
16 printf("mean free path is %.2e m",rho)
17 //answer given in the book is wrong
```

Scilab code Exa 6.3 relaxation time

```
1 //Example number 6.3, Page number 117
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 rho=1.43*10**-8;     //resistivity(ohm m)
8 n=6.5*10**28;        //conduction electrons(per m**3)
9 e=1.6*10**-19;       //charge(c)
10 m=9.1*10**-34;       //mass(kg)
11 //Calculation
12 towr=m/(n*e**2*rho);      //relaxation time(sec)
13 //Result
14 printf("relaxation time is %.3e sec",towr)
```

```
15 //answer in the book varies due to rounding off
      errors
```

Scilab code Exa 6.4 temperature

```
1  //Example number 6.4, Page number 117
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  PE=1/100;       //probability
8  E_EF=0.5;       //energy difference
9  //Calculation
10 x=log((1/PE)-1);
11 T=E_EF/x;       //temperature(K)
12 //Result
13 printf("temperature is %.4f K",T)
14 //answer given in the book is wrong
```

Scilab code Exa 6.5 mobility

```
1  //Example number 6.5, Page number 117
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  d=8.92*10**3;       //density(kg/m**3)
8  rho=1.73*10**-8;     //resistivity(ohm m)
9  M=63.5;     //atomic weight
10 N=6.02*10**26;      //avagadro number
11 e=1.6*10**-19;       //charge(c)
```

```
m=9.1*10**-31;        //mass(kg)
//Calculation
n=d*N/M;
mew=1/(rho*n*e);        //mobility(m/Vs)
tow=m/(n*e**2*rho);     //average time(sec)
//Result
printf("mobility is %.3e m/Vs",mew)
printf("\n average time is %.2e sec",tow)
```

Scilab code Exa 6.6 temperature

```
//Example number 6.6, Page number 118

clc;clear;
close;

//Variable declaration
EF=5.5;       //energy(eV)
FE=10/100;    //probability
e=1.6*10**-19;        //charge(c)
Kb=1.38*10**-23;      //boltzmann constant(J/k)
//Calculation
E=EF+(EF/100);
x=(E-EF)*e;
y=x/Kb;
z=(1/FE)-1;
T=y/log(z);       //temperature(K)
//Result
printf("temperature is %.1f K",T)
```

Scilab code Exa 6.7 kinetic energy

```
//Example number 6.7, Page number 119
```

```
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 Kb=1.38*10**-23;      //boltzmann constant(J/k)
8 T=303;      //temperature(K)
9 e=1.6*10**-19;       //charge(c)
10 MH=2*1.008*1.67*10**-27;    //mass(kg)
11 //Calculation
12 KE=3*Kb*T/(2*e);      //kinetic energy(eV)
13 cbar=sqrt(3*Kb*T/MH);      //velocity(m/s)
14 //Result
15 printf("kinetic energy is %.1e eV",KE)
16 printf("\n velocity is %.2f m/s",cbar)
17 //answer given in the book is wrong
```

Scilab code Exa 6.8 density of electrons

```
1 //Example number 6.8, Page number 119
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 rho=10**4;       //density of silver(kg/m**3)
8 N=6.02*10**26;     //avagadro number
9 e=1.6*10**-19;       //charge(c)
10 m=9.1*10**-31;       //mass(kg)
11 MA=107.9;      //atomic weight(kg)
12 sigma=7*10**7;     //conductivity(per ohm m)
13 //Calculation
14 n=rho*N/MA;       //density of electrons(per m**3)
15 mew=sigma/(n*e*10**2);     //mobility of electrons(m
   **2/Vs)
```

```
16 tow=sigma*m*10**15/(n*e**2);     //collision time(n
     sec)
17 //Result
18 printf("density of electrons is %.1e m^3",n)
19 printf("\n mobility of electrons is %.4e m^2/Vs",mew
     )
20 printf("\n collision time is %.1f sec",tow)
```

Scilab code Exa 6.9 electron velocity

```
1 //Example number 6.9, Page number 120
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 Ee=10;      //electron kinetic energy(eV)
8 Ep=10;      //proton kinetic energy(eV)
9 e=1.6*10**-19;       //charge(c)
10 me=9.1*10**-31;       //mass(kg)
11 mp=1.67*10**-27;      //mass(kg)
12 //Calculation
13 cebar=sqrt(2*Ee*e/me);     //electron velocity(m/s)
14 cpbar=sqrt(2*Ep*e/mp);     //proton velocity(m/s)
15 //Result
16 printf("electron velocity is %.3e m/s",cebar)
17 printf("\n proton velocity is %.3e m/s",cpbar)
18 //answers given in the book are wrong
```

Scilab code Exa 6.10 drift velocity

```
1 //Example number 6.10, Page number 120
2
```

```
clc;clear;
close;

//Variable declaration
A=10*10**-6;      //area(m**2)
i=100;     //current(amp)
n=8.5*10**28;     //number of electrons
e=1.6*10**-19;       //charge(c)
//Calculation
vd=i/(n*A*e);      //drift velocity(m/s)
//Result
printf("drift velocity is %.4e m/s",vd)
```

Scilab code Exa 6.11 thermal conductivity

```
//Example number 6.11, Page number 121

clc;clear;
close;

//Variable declaration
Kb=1.38*10**-23;      //boltzmann constant(J/k)
m=9.1*10**-31;       //mass(kg)
tow=3*10**-14;     //relaxation time(sec)
n=8*10**28;       //density of electrons(per m**3)
T=273;      //temperature(K)
//Calculation
sigma_T=3*n*tow*T*Kb**2/(2*m);       //thermal
   conductivity(W/mK)
//Result
printf("thermal conductivity is %.3f W/mK",sigma_T)
```

Chapter 7

DIELECTRICS

Scilab code Exa 7.1 electronic polarisability

```
1 //Example number 7.1, Page number 146
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 epsilonr=3.75;        //relative dielectric constant
8 T=27;     //temperature(C)
9 gama=1/3;       //internal field constant
10 rho=2050;       //density(kg/m**3)
11 Ma=32;          //atomic weight(amu)
12 Na=6.022*10**23;       //avagadro number
13 epsilon0=8.85*10**-12;
14 //Calculation
15 x=(epsilonr-1)/(epsilonr+2);
16 alpha_e=x*Ma*3*epsilon0/(rho*Na);      //electronic
    polarisability(Fm**2)
17 //Result
18 printf("electronic polarisability is %.3e Fm^2",
    alpha_e)
19 //answer varies due to rounding off errors
```

Scilab code Exa 7.2 capacitance

```
1 //Example number 7.2, Page number 146
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 A=100*10**-4;       //area(m**2)
8 epsilon0=8.85*10**-12;
9 d=1*10**-2;       //separation(m)
10 V=100;       //potential(V)
11 //Calculation
12 C=A*epsilon0/d*10**12;       //capacitance(PF)
13 Q=(C/10**12)*V;          //charge on plates(C)
14 //Result
15 printf("capacitance is %.2f pF",C)
16 printf("\n charge on plates is %.2e C",Q)
```

Scilab code Exa 7.3 polarisability

```
1 //Example number 7.3, Page number 147
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 epsilonr=1.0000684;      //dielectric constant
8 N=2.7*10**25;    //number of atoms
9 epsilon0=8.85*10**-12;
10 //Calculation
```

```
11 alpha_e=epsilon0*(epsilonr-1)/N;     //polarisability
   (Fm**2)
12 //Result
13 printf("polarisability is %.3e Fm^2",alpha_e)
14 //answer varies due to rounding off errors
```

Scilab code Exa 7.4 dielectric constant

```
1 //Example number 7.4, Page number 147
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 alpha_e=10**-40;       //polarisability(Fm**2)
8 N=3*10**28;     //density of atoms
9 epsilon0=8.85*10**-12;
10 //Calculation
11 x=N*alpha_e/epsilon0;
12 epsilonr=(1+(2*x))/(1-x);      //dielectric constant(
   F/m)
13 //Result
14 printf("dielectric constant is %.3f F/m",epsilonr)
15 //answer in the book is wrong
```

Scilab code Exa 7.5 voltage

```
1 //Example number 7.5, Page number 147
2
3 clc;clear;
4 close;
5
6 //Variable declaration
```

```
A=650*10**-4;        //area(m**2)
epsilon0=8.85*10**-12;
d=4*10**-2;        //seperation(m)
Q=2*10**-10;       //charge(C)
epsilonr=3.5;      //dielectric constant
//Calculation
C=A*epsilon0/d;
V=Q/C;      //voltage(V)
//Result
printf("voltage is %.1f V",V)
```

Scilab code Exa 7.6 polarisability

```
//Example number 7.6, Page number 148

clc;clear;
close;

//Variable declaration
A=6.45*10**-4;        //area(m**2)
epsilon0=8.85*10**-12;
d=2*10**-3;        //seperation(m)
epsilonr=5;     //dielectric constant
N=6.023*10**23;     //avagadro number
//Calculation
alpha_e=epsilon0*(epsilonr-1)/N;      //polarisability
    (Fm**2)
//Result
printf("polarisability is %.3e Fm^2",alpha_e)
//answer in the book is wrong
```

Scilab code Exa 7.7 radius of electron

```
1 //Example number 7.7, Page number 148
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 epsilonr=1.0000684;      //dielectric constant
8 Na=2.7*10**25;    //number of atoms
9 x=1/(9*10**9);
10 E=10**6;    //electric field(V/m)
11 e=1.6*10**-19;       //charge(c)
12 Z=2;     //atomic number
13 //Calculation
14 r0=((epsilonr-1)/(4*%pi*Na))**(1/3);      //radius of
       electron cloud(m)
15 X=x*E*r0**3/(Z*e);          //displacement(m)
16 //Result
17 printf("radius of electron cloud is %.2e m",r0)
18 printf("\n displacement is %.4e m",X)
```

Scilab code Exa 7.8 electronic polarisability

```
1 //Example number 7.8, Page number 149
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 epsilonr=4;        //relative dielectric constant
8 Na=2.08*10**23;      //avagadro number
9 epsilon0=8.85*10**-12;
10 //Calculation
11 x=(epsilonr-1)/(epsilonr+2);
12 alpha_e=x*3*epsilon0/Na;      //electronic
      polarisability(Fm**2)
```

```
13 //Result
14 printf("electronic polarisability is %.3e Fm^2",
      alpha_e)
15 //answer in the book is wrong
```

Scilab code Exa 7.9 energy in condenser

```
1 //Example number 7.9, Page number 149
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 C=4*10**-6;     //capacitance(F)
8 epsilonr=200;        //relative dielectric constant
9 V=2000;        //voltage(V)
10 //Calculation
11 C0=C/epsilonr;     //energy in condenser(F)
12 E=C0*V/2;       //energy in dielectric(J)
13 //Result
14 printf("energy in condenser is %.e F",C0)
15 printf("\n energy in dielectric is %.1e J",E)
16 //answer in the book is wrong
```

Scilab code Exa 7.10 relative permittivity

```
1 //Example number 7.10, Page number 149
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 epsilon0=8.85*10**-12;
```

```
8 N=2.7*10**25;    //density of atoms
9 R=0.55*10**-10;     //radius(m)
10 //Calculation
11 alpha_e=4*%pi*epsilon0*R**3;      //polarisability(Fm
   **2)
12 epsilonr=(N*alpha_e/epsilon0)+1;      //relative
   permittivity
13 //Result
14 printf("polarisability is %.3e Fm^2",alpha_e)
15 printf("\n relative permittivity is %.7f Fm^2",
   epsilonr)
```

Scilab code Exa 7.11 field strength

```
1 //Example number 7.11, Page number 150
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 A=180*10**-4;      //area(m**2)
8 epsilonr=8;      //relative permittivity
9 C=3*10**-6;      //capacitance(F)
10 V=10;           //potential(V)
11 epsilon0=8.85*10**-12;
12 //Calculation
13 E=V*C/(epsilon0*epsilonr);     //field strength(V/m)
14 dm=epsilon0*(epsilonr-1)*A*E;      //total dipole
   moment(coul m)
15 //Result
16 printf("field strength is %.4e V/m",E)
17 printf("\n total dipole moment is %.4e Coul.m",dm)
18 //answer in the book is wrong"
```

Chapter 8

MAGNETIC MATERIALS

Scilab code Exa 8.1 change in magnetic moment

```
1 //Example number 8.1, Page number 170
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 r=0.05*10**-9;      //radius(m)
8 B=1;      //magnetic induction(web/m**2)
9 e=1.6*10**-19;      //charge(c)
10 m=9.1*10**-31;      //mass(kg)
11 //Calculation
12 d_mew=e**2*r**2*B/(4*m);        //change in magnetic
    moment(Am**2)
13 //Result
14 printf("change in magnetic moment is %.2e Amˆ2",
    d_mew)
15 //answer in the book is wrong
```

Scilab code Exa 8.2 magnetic flux density

```
//Example number 8.2, Page number 170

clc;clear;
close;

//Variable declaration
chi=-0.5*10**-5;        //magnetic susceptibility
H=9.9*10**4;          //magnetic field intensity(amp/m
    )
mew0=4*%pi*10**-7;
//Calculation
I=chi*H;       //intensity of magnetisation(amp/m)
B=mew0*H*(1+chi);        //magnetic flux density(wb/m
    **2)
//Result
printf("intensity of magnetisation is %.3f amp/m",I)
printf("\n magnetic flux density is %.3f Wb/m^2",B)
```

Scilab code Exa 8.3 relative permeability

```
//Example number 8.3, Page number 170

clc;clear;
close;

//Variable declaration
H=220;          //magnetic field intensity(amp/m)
I=3300;       //magnetisation(amp/m)
//Calculation
mewr=1+(I/H);       //relative permeability
//Result
printf("relative permeability is %d",mewr)
```

Scilab code Exa 8.4 magnetic induction

```
1 //Example number 8.4, Page number 171
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 r=6.1*10**-11;      //radius of atom(m)
8 new=8.8*10**15;     //frequency(revolution/sec)
9 mew0=4*%pi*10**-7;
10 e=1.6*10**-19;     //charge(c)
11 //Calculation
12 i=e*new;       //current(amp)
13 B=mew0*i/(2*r);      //magnetic induction(web/m**2)
14 mew=i*%pi*r**2;      //dipole moment(amp m**2)
15 //Result
16 printf("magnetic induction is %.3f Wb/m^2",B)
17 printf("\n dipole moment is %.3e Amp-m^2",mew)
18 //answers in the book are wrong
```

Scilab code Exa 8.5 average number of bohr magnetons

```
1 //Example number 8.5, Page number 171
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 Is=1.96*10**6;         //saturation magnetisation(amp
    /m)
8 a=3*10**-10;          //cube edge(m)
9 mewB=9.27*10**-24;    //bohr magneton(amp/m**2)
10 n=2;         //number of atoms
11 //Calculation
```

```
12 N=n/(a**3);
13 mew_bar=Is/(N*mewB);      //average number of bohr
      magnetons(bohr magneton/atom)
14 //Result
15 printf("average number of bohr magnetons is %.3f
      bohr magneton/atom",mew_bar)
```

Scilab code Exa 8.6 magnetizing force

```
1 //Example number 8.6, Page number 172
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 I=3000;        //magnetisation(amp/m)
8 mew0=4*%pi*10**-7;
9 B=0.005;       //flux density(weber/m**2)
10 //Calculation
11 H=(B/mew0)-I;        //magnetizing force(amp/m)
12 mewr=(I/H)+1;        //relative permeability
13 //Result
14 printf("magnetizing force is %.3f Amp/m",H)
15 printf("\n relative permeability is %.3f",mewr)
16 //answer in the book varies due to rounding off
      errors
```

Scilab code Exa 8.7 permeability

```
1 //Example number 8.7, Page number 172
2
3 clc;clear;
4 close;
```

```
5
6 //Variable declaration
7 H=1800;        //magnetizing force(amp/m)
8 chi=3*10**-5;      //magnetic flux(wb)
9 A=0.2*10**-4;      //area(m**2)
10 //Calculation
11 B=chi/A;
12 mew=B/H;       //permeability(henry/m)
13 //Result
14 printf("permeability is %.3e H/m^2",mew)
15 //answer in the book is wrong
```

Scilab code Exa 8.8 magnetic dipole moment

```
1 //Example number 8.8, Page number 172
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 r=0.04;     //radius(m)
8 i=1000*10**-3;      //current(mA)
9 B=10**-3;    //magnetic flux density(wb/m**2)
10 theta=45;    //angle(degrees)
11 //Calculation
12 A=%pi*r**2;       //area(m**2)
13 mew=i*A;          //magnetic dipole moment(amp m**2)
14 theta=theta*%pi/180;
15 tow=i*B*cos(theta);      //torque(Nm)
16 //Result
17 printf("magnetic dipole moment is %.4e Amp-m^2",mew)
18 printf("\n torque is %.4e Nm",tow)
19
20 //answer in the book varies due to rounding off
   errors
```

Scilab code Exa 8.9 hysteresis loss

```
1 //Example number 8.9, Page number 173
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 A=100;      //area(m**2)
8 B=0.01;     //flux density(wb/m**2)
9 H=40;       //magnetic field(amp/m)
10 M=7650;    //atomic weight(kg/m**3)
11 //Calculation
12 h=A*B*H;     //hysteresis loss per cycle(J/m**3)
13 //Result
14 printf("hysteresis loss per cycle is %.f J/m^3",h)
```

Scilab code Exa 8.10 power loss

```
1 //Example number 8.10, Page number 173
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 h=200;      //hysteresis loss per cycle(J/m**3)
8 M=7650;     //atomic weight(kg/m**3)
9 n=100;      //magnetisation cycles per second
10 //Calculation
11 hpl=h*n;      //hysteresis power loss per second(watt
   /m**3)
```

```
12 pl=hpl/M;     //power loss(watt/kg)
13 //Result
14 printf("hysteresis power loss per second is %.f W/m
      ^3",hpl)
15 printf("\n power loss is %.3f W/kg",pl)
```

Chapter 9

BONDING IN SOLIDS

Scilab code Exa 9.1 resistivity

```
1 //Example number 9.1, Page number 202
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;      //charge(c)
8 ni=2.4*10**19;     //particle density(per m**3)
9 mew_e=0.39;         //electron mobility(m**2/Vs)
10 mew_h=0.19;        //hole mobility(m**2/Vs)
11 //Calculation
12 rho=1/(ni*e*(mew_e+mew_h));        //resistivity(ohm
   m)
13 //Result
14 printf("resistivity is %.5f ohm-m",rho)
```

Scilab code Exa 9.2 hole concentration

```
1 //Example number 9.2, Page number 203
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;      //charge(c)
8 ni=1.5*10**16;     //particle density(per m**3)
9 mew_e=0.13;        //electron mobility(m**2/Vs)
10 mew_h=0.048;      //hole mobility(m**2/Vs)
11 ND=10**23;      //density(per m**3)
12 //Calculation
13 sigma_i=ni*e*(mew_e+mew_h);          //conductivity(
   s)
14 sigma=ND*mew_e*e;       //conductivity(s)
15 P=ni**2/ND;             //equilibrium hole
   concentration(per m**3)
16 //Result
17 printf("conductivity is %.2e s",sigma_i)
18 printf("\n conductivity is %.3e s",sigma)
19 printf("\n equilibrium hole concentration is %.2e
   per m^3",P)
20 //answer in the book varies due to rounding off
   errors
```

Scilab code Exa 9.3 intrinsic conductivity

```
1 //Example number 9.3, Page number 203
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;      //charge(c)
8 ni=1.5*10**16;     //particle density(per m**3)
```

```
mew_e=0.13;          //electron mobility(m**2/Vs)
mew_h=0.05;          //hole mobility(m**2/Vs)
ND=5*10**20;         //density(per m**3)
//Calculation
sigma=ni*e*(mew_e+mew_h);              //intrinsic
   conductivity(s)
sigma_d=ND*e*mew_e;          //conductivity during
   donor impurity(ohm-1 m-1)
sigma_a=ND*e*mew_h;          //conductivity during
   acceptor impurity(ohm-1 m-1)
//Result
printf("intrinsic conductivity is %.3e (ohm-m)^-1",
   sigma)
printf("\n conductivity during donor impurity is %.1
   f (ohm-m)^-1",sigma_d)
printf("\n conductivity during donor impurity is %.f
    (ohm-m)^-1",sigma_a)
```

Scilab code Exa 9.4 density of atoms

```
//Example number 9.4, Page number 204

clc;clear;
close;

//Variable declaration
RH=3.66*10**-4;       //hall coefficient(m**3/c)
rho=8.93*10**-3;      //resistivity(m)
e=1.6*10**-19;        //charge(c)
//Calculation
mew=RH/rho;           //mobility(m**2/Vs)
n=1/(RH*e);           //density of atoms(per m**3)
//Result
printf("mobility is %.5f m^2/Vs",mew)
printf("\n density of atoms is %.1e per m^3",n)
```

```
16
17 //answer in the book varies due to rounding off
      errors
```

Scilab code Exa 9.5 carrier density

```
1  //Example number 9.5, Page number 204
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  w=72.6;          //atomic weight
8  e=1.6*10**-19;       //charge(c)
9  mew_e=0.4;          //electron mobility(m**2/Vs)
10 mew_h=0.2;         //hole mobility(m**2/Vs)
11 T=300;            //temperature(K)
12 x=4.83*10**21;
13 Eg=0.7;         //band gap(eV)
14 y=0.052;
15 //Calculation
16 ni=x*(T**(3/2))*exp(-Eg/y);       //carrier density(
      per m**3)
17 sigma=ni*e*(mew_e+mew_h);           //conductivity(ohm
      -1 m-1)
18 //Result
19 printf("carrier density is %.2e per m^3",ni)
20 printf("\n conductivity is %.2f (ohm-m)^-1",sigma)
21 //answer in the book varies due to rounding off
      errors
```

Scilab code Exa 9.6 energy band gap

```
1  //Example number 9.6, Page number 205
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  T1=293;      //temperature(K)
8  T2=305;      //temperature(K)
9  e=1.6*10**-19;      //charge(c)
10 sigma1=2;
11 sigma2=4.5;
12 KB=1.38*10**-23;    //boltzmann constant
13 //Calculation
14 x=((1/T1)-(1/T2));
15 y=log(sigma2/sigma1);
16 z=3*log(T2/T1)/2;
17 Eg=2*KB*(y+z)/(e*x);     //energy band gap(eV)
18 //Result
19 printf("energy band gap is %.2f eV",Eg)
20 //answer in the book is wrong
```

Scilab code Exa 9.7 diffusion coefficient

```
1  //Example number 9.7, Page number 205
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  e=1.6*10**-19;      //charge(c)
8  mew_e=0.19;         //electron mobility(m**2/Vs)
9  T=300;          //temperature(K)
10 KB=1.38*10**-23;    //boltzmann constant
11 //Calculation
12 Dn=mew_e*KB*T/e;     //diffusion coefficient(m**2/sec
```

```
    )
13 //Result
14 printf("diffusion coefficient is %.1e m^2/s",Dn)
```

Scilab code Exa 9.8 carrier density

```
1  //Example number 9.8, Page number 206
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  sigma=2.12;      //conductivity(ohm-1 m-1)
8  T=300;       //temperature(K)
9  e=1.6*10**-19;      //charge(c)
10 mew_e=0.36;         //electron mobility(m**2/Vs)
11 mew_h=0.7;        //hole mobility(m**2/Vs)
12 C=4.83*10**21;
13 KB=1.38*10**-23;    //boltzmann constant
14 //Calculation
15 ni=sigma/(e*(mew_e+mew_h));      //carrier density(
       per m**3)
16 x=C*T**(3/2)/ni;
17 Eg=2*KB*T*log(x)/e;       //energy gap(eV)
18 //Result
19 printf("carrier density is %.2e per m^3",ni)
20 printf("\n energy gap is %.2f eV",Eg)
21 //answer in the book is wrong
```

Scilab code Exa 9.9 probability of occupation

```
1  //Example number 9.9, Page number 206
2
```

```
clc;clear;
close;

//Variable declaration
Eg=6.408*10**-20;     //energy gap of semiconductor(J
    )
T1=273;     //temperature(K)
T2=323;     //temperature(K)
T3=373;     //temperature(K)
KB=1.38*10**-23;    //boltzmann constant
//Calculation
FE1=1/(1+exp(Eg/(2*KB*T1)));      //probability of
    occupation at 0C(eV)
FE2=1/(1+exp(Eg/(2*KB*T2)));      //probability of
    occupation at 50C(eV)
FE3=1/(1+exp(Eg/(2*KB*T3)));      //probability of
    occupation at 100C(eV)
//Result
printf("probability of occupation at 0C is %.3e eV",
    FE1)
printf("\n probability of occupation at 50C is %.2e
    eV",FE2)
printf("\n probability of occupation at 100C is %.2e
     eV",FE3)
```

Scilab code Exa 9.10 ratio between conductivity

```
//Example number 9.10, Page number 207

clc;clear;
close;

//Variable declaration
Eg=1.9224*10**-19;     //energy gap of semiconductor(
    J)
```

```
T1=600;     //temperature(K)
T2=300;     //temperature(K)
x=-1.666*10**-3;
KB=1.38*10**-23;    //boltzmann constant
//Calculation
T=(1/T1)-(1/T2);
r=exp(x*(-Eg/(2*KB)));      //ratio between
   conductivity
//Result
printf("ratio between conductivity is %.3e",r)
```

Scilab code Exa 9.11 resistivity of material

```
//Example number 9.11, Page number 207

clc;clear;
close;

//Variable declaration
ni=2.5*10**19;       //charge carriers(per m**3)
r=10**-6;     //ratio
e=1.6*10**-19;       //charge(c)
mew_e=0.36;          //electron mobility(m**2/Vs)
mew_h=0.18;         //hole mobility(m**2/Vs)
N=4.2*10**28;        //number of atoms(per m**3)
//Calculation
Ne=r*N;         //number of impurity atoms(per m**3)
Nh=ni**2/Ne;
sigma=(Ne*e*mew_e)+(Nh*e*mew_h);       //conductivity
   (ohm m)
rho=1/sigma;      //resistivity of material(per ohm m
   )
//Result
printf("resistivity of material is %.4e ohm-m",rho)
```

Scilab code Exa 9.12 conductivity

```
1 //Example number 9.12, Page number 208
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n=5*10**17;      //concentration(m**3)
8 vd=350;     //drift velocity(m/s)
9 E=1000;     //electric field(V/m)
10 e=1.6*10**-19;      //charge(c)
11 //Calculation
12 sigma=n*e*vd/E;     //conductivity(per ohm m)
13 //Result
14 printf("conductivity is %.3f per ohm-m",sigma)
```

Scilab code Exa 9.13 concentration

```
1 //Example number 9.13, Page number 208
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 sigmae=2.2*10**-4;        //conductivity(ohm/m)
8 mew_e=125*10**-3;        //electron mobility(m**2/Vs)
9 e=1.602*10**-19;      //charge(c)
10 //Calculation
11 ne=sigmae/(e*mew_e);      //concentration(per m**3)
12 //Result
13 printf("concentration is %.1e per m^3",ne)
```

Scilab code Exa 9.14 density of charge carriers

```
1 //Example number 9.14, Page number 209
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 RH=3.66*10**-4;        //hall coefficient(m*3/c)
8 rho_i=8.93*10**-3;      //resistivity(ohm m)
9 e=1.602*10**-19;        //charge(c)
10 //Calculation
11 nh=1/(RH*e);         //density of charge carriers(per m
    **3)
12 mewh=1/(rho_i*nh*e);       //mobility of charge
    carriers(m**2/Vs)
13 //Result
14 printf("density of charge carriers is %.4e per m^3",
    nh)
15 printf("\n mobility of charge carriers is %.3f m^2/
    Vs",mewh)
```

Scilab code Exa 9.15 hall voltage

```
1 //Example number 9.15, Page number 209
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 I=3*10**-3;      //current(A)
```

```
RH=3.66*10**-4;      //hall coefficient(m**3/C)
e=1.6*10**-19;       //charge(c)
d=2*10**-2;
z=1*10**-3;
B=1;           //magnetic field(wb/m**2)
//Calculation
w=d*z;       //width(m**2)
A=w;       //area(m**2)
EH=RH*I*B/A;
VH=EH*d*10**3;       //hall voltage(mV)
n=1/(RH*e);       //charge carrier concentration(per m
   **3)
//Result
printf("hall voltage is %.1f mH",VH)
printf("\n charge carrier concentration is %.2e per
   m^3",n)
```

Chapter 10

SUPERCONDUCTIVITY

Scilab code Exa 10.1 Critical field

```
1 //Example number 10.1, Page number 224
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 T=5;      //temperature(K)
8 Tc=7.2;    //critical temperature(K)
9 H0=6.5*10**3;    //critical magnetic field(A/m)
10 //Calculation
11 Hc=H0*(1-(T/Tc)**2);     //critical field(A/m)
12 //Result
13 printf("critical field is %.3e A/m",Hc)
```

Scilab code Exa 10.2 critical field

```
1 //Example number 10.2, Page number 225
2
```

```
clc;clear;
close;

//Variable declaration
T=2.5;      //temperature(K)
Tc=3.5;    //critical temperature(K)
H0=3.2*10**3;    //critical magnetic field(A/m)
//Calculation
Hc=H0*(1-(T/Tc)**2);     //critical field(A/m)
//Result
printf("critical field is %.3e A/m",Hc)
```

Scilab code Exa 10.3 critical temperature

```
//Example number 10.3, Page number 225

clc;clear;
close;

//Variable declaration
Hc=5*10**3;     //critical magnetic field(A/m)
T=6;      //temperature(K)
H0=2*10**4;    //critical magnetic field(A/m)
//Calculation
Tc=T/sqrt(1-(Hc/H0));      //critical temperature(K)
//Result
printf("critical temperature is %.3f K",Tc)
//answer given in the book is wrong
```

Scilab code Exa 10.4 critical current

```
//Example number 10.4, Page number 225

```

```
clc;clear;
close;

//Variable declaration
Hc=2*10**3;     //critical magnetic field(A/m)
r=0.02;     //radius(m)
//Calculation
Ic=2*%pi*r*Hc;      //critical current(amp)
//Result
printf("critical current is %.1f A",Ic)
//answer in the book varies due to rounding off
   errors
```

Scilab code Exa 10.5 isotopic mass

```
//Example number 10.5, Page number 225

clc;clear;
close;

//Variable declaration
T1=5;      //temperature(K)
T2=5.1;      //temperature(K)
M1=199.5;    //isotopic mass(amu)
//Calculation
M2=M1*(T1/T2)**2;     //isotopic mass(amu)
//Result
printf("isotopic mass is %.2f a.m.u.",M2)
```

Scilab code Exa 10.6 critical current

```
//Example number 10.6, Page number 226

```

```
clc;clear;
close;

//Variable declaration
T=5;      //temperature(K)
Tc=8;    //critical temperature(K)
H0=5*10**4;    //critical magnetic field(A/m)
r=1.5*10**-3;     //radius(m)
//Calculation
Hc=H0*(1-(T/Tc)**2);     //critical field(A/m)
Ic=2*%pi*r*Hc;      //critical current(amp)
//Result
printf("critical field is %.4e A/m",Hc)
printf("\n critical current is %.3f A",Ic)
//answer in the book varies due to rounding off
   errors
```

Scilab code Exa 10.7 critical temperature

```
//Example number 10.7, Page number 226

clc;clear;
close;

//Variable declaration
Tc1=4.185;      //critical temperature(K)
M1=199.5;    //isotopic mass(amu)
M2=203.4;    //isotopic mass(amu)
//Calculation
Tc2=Tc1*sqrt(M1/M2);       //critical temperature(K)
//Result
printf("critical temperature is %.4f K",Tc2)
```

Scilab code Exa 10.8 frequency

```
1 //Example number 10.8, Page number 226
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;     //charge(c)
8 h=6.626*10**-36;    //plank constant
9 V=8.5*10**-6;     //voltage(V)
10 //Calculation
11 new=2*e*V/h;     //frequency(Hz)
12 //Result
13 printf("frequency is %.3e Hz",new)
```

Scilab code Exa 10.9 critical temperature

```
1 //Example number 10.9, Page number 227
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 Tc1=5;      //critical temperature(K)
8 P1=1;    //pressure(mm)
9 P2=6;    //pressure(mm)
10 //Calculation
11 Tc2=Tc1*P2/P1;       //critical temperature(K)
12 //Result
13 printf("critical temperature is %.f K",Tc2)
```

Scilab code Exa 10.10 maximum critical temperature

```
1 //Example number 10.10, Page number 227
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 Hc=6*10**5;     //critical magnetic field(A/m)
8 Tc=8.7;      //critical temperature(K)
9 H0=3*10**6;    //critical magnetic field(A/m)
10 //Calculation
11 T=Tc*sqrt(1-(Hc/H0));      //maximum critical
      temperature(K)
12 //Result
13 printf("maximum critical temperature is %.3f K",T)
14 //answer given in the book is wrong
```

Chapter 11

LASER

Scilab code Exa 11.1 matter wave energy

```
1 //Example number 11.1, Page number 246
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;      //charge(coulomb)
8 v=3*10**3;    //velocity of matter wave(m/s)
9 h=6.6*10**-34;     //plank's constant(Js)
10 lamda=600*10**-9;     //wavelength(m)
11 //Calculation
12 Ej=h*v/lamda;     //matter wave energy(J)
13 E=Ej/e;       //matter wave energy(eV)
14 //Result
15 printf("matter wave energy is %.2e eV",E)
16 //answer given in the book is wrong
```

Scilab code Exa 11.2 wavelength of photon

```
1 //Example number 11.2, Page number 246
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;      //charge(coulomb)
8 c=3*10**10;    //velocity of light(m/s)
9 h=6.6*10**-34;     //plank's constant(Js)
10 Eg=3;     //energy gap(eV)
11 //Calculation
12 lamda=h*c*10**9/(Eg*e);      //wavelength of photon(
     nm)
13 //Result
14 printf("wavelength of photon is %.f nm",lamda)
15 //answer given in the book is wrong
```

Scilab code Exa 11.3 ratio in higher and lower energy

```
1 //Example number 11.3, Page number 246
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 e=1.6*10**-19;      //charge(coulomb)
8 E2_E1=3*e;     //energy gap(J)
9 Kb=1.38*10**-23;     //boltzmann constant(J/K)
10 T=323;     //temperature(K)
11 //Calculation
12 n=exp(-E2_E1/(Kb*T));     //ratio in higher and lower
     energy
13 //Result
14 printf("ratio in higher and lower energy is %.4e",n)
15 //answer given in the book is wrong
```

Scilab code Exa 11.4 ratio of emission

```
1 //Example number 11.4, Page number 247
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 c=2.998*10**8;      //velocity of light(m/s)
8 Kb=1.381*10**-23;     //boltzmann constant(J/K)
9 T=1000;     //temperature(K)
10 h=6.626*10**-34;     //plank's constant(Js)
11 lamda=0.5*10**-6;    //wavelength(m)
12 //Calculation
13 v=c/lamda;     //frequency(Hz)
14 BA=1/(exp(h*v/(Kb*T))-1);       //ratio of emission
15 //Result
16 printf("ratio of emission is %.1e",BA)
17 //answer varies due to rounding off errors
```

Scilab code Exa 11.5 wavelength

```
1 //Example number 11.5, Page number 247
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 c=2.998*10**8;      //velocity of light(m/s)
8 h=6.626*10**-34;     //plank's constant(Js)
9 e=1.602*10**-19;      //charge(coulomb)
```

```
10 Eg=1.43;     //energy gap(eV)
11 //Calculation
12 lamda=h*c*10**6/(Eg*e);          //wavelength(micro m)
13 //Result
14 printf("wavelength is %.2f micro-m",lamda)
```

Chapter 12

FIBRE OPTICS

Scilab code Exa 12.1 refractive index

```
1 //Example number 12.1, Page number 263
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 NA=0.39;     //numerical aperture
8 delta=0.05;      //refractive index of cladding
9 //Calculation
10 n1=NA/sqrt(2*delta);       //refractive index of core
11 //Result
12 printf("refractive index of core is %.3f",n1)
```

Scilab code Exa 12.2 fractional index change

```
1 //Example number 12.2, Page number 264
2
3 clc;clear;
```

```
close;

//Variable declaration
n1=1.563;           //Core refractive index
n2=1.498;           //Cladding refractive index
//Calculation
delta=(n1-n2)/n1;     //fractional index change
//Result
printf("fractional index change is %.5f",delta)
```

Scilab code Exa 12.3 numerical aperture

```
//Example number 12.3, Page number 264

clc;clear;
close;

//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("numerical aperture is %.2f",NA)
```

Scilab code Exa 12.4 acceptance angle

```
//Example number 12.4, Page number 264

clc;clear;
close;

//Variable declaration
```

```
7 n1=1.563;           //Core refractive index
8 n2=1.498;           //Cladding refractive index
9 //Calculation
10 NA=sqrt(n1**2-n2**2);     //numerical aperture
11 theta0=asin(NA);     //acceptance angle(radian)
12 theta0=theta0*180/%pi;     //acceptance angle(degrees
    )
13 //Resul"
14 printf("acceptance angle is %.2f degree",theta0)
```

Scilab code Exa 12.5 critical angle

```
1 //Example number 12.5, Page number 265
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n1=1.53;           //Core refractive index
8 n2=1.42;           //Cladding refractive index
9 //Calculation
10 thetac=asin(n2/n1);      //critical angle(radian)
11 thetac=thetac*180/%pi;     //critical angle(degrees)
12 //Resul"
13 printf("critical angle is %.2f degree",thetac)
```

Scilab code Exa 12.6 acceptance angle

```
1 //Example number 12.6, Page number 265
2
3 clc;clear;
4 close;
5
```

```
6 //Variable declaration
7 n1=1.6;           //Core refractive index
8 n0=1.33;          //refractive index of air
9 n2=1.4;           //Cladding refractive index
10 //Calculation
11 NA=sqrt(n1**2-n2**2)/n0;     //numerical aperture
12 theta0=asin(NA);     //acceptance angle(radian)
13 theta0=theta0*180/%pi;     //acceptance angle(degrees
   )
14 //Resul"
15 printf("acceptance angle is %.2f degree",theta0)
16 //answer in the book varies due to rounding off
   errors
```

Scilab code Exa 12.7 fractional index change

```
1 //Example number 12.7, Page number 265
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n1=1.5;           //Core refractive index
8 n2=1.3;           //Cladding refractive index
9 //Calculation
10 delta=(n1-n2)/n1;     //fractional index change
11 //Result
12 printf("fractional index change is %.3f delta",delta
   )
```

Scilab code Exa 12.8 refraction angle

```
1 //Example number 12.8, Page number 265
```

```
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n1=1.55;           //Core refractive index
8 n2=1.6;           //Cladding refractive index
9 theta1=60*%pi/180;      //incident angle(degrees)
10 //Calculation
11 x=n1*sin(theta1)/n2;
12 theta2=asin(x);     //refraction angle(radian)
13 theta2=theta2*180/%pi;     //refraction angle(degrees
   )
14 //Result
15 printf("refraction angle is %.2f degree",theta2)
```

Scilab code Exa 12.9 refractive index

```
1 //Example number 12.9, Page number 266
2
3 clc;clear;
4 close;
5
6 //Variable declaration
7 n2=1.3;           //Cladding refractive index
8 delta=0.140;     //fractional index change
9 //Calculation
10 n1=n2/(1-delta);      //Core refractive index
11 //Result
12 printf("refractive index of core is %.2f",n1)
```

Scilab code Exa 12.10 numerical aperture

```
1  //Example number 12.10, Page number 266
2
3  clc;clear;
4  close;
5
6  //Variable declaration
7  theta0=26.80*%pi/180;     //acceptance angle(radian)
8  //Calculation
9  NA=sin(theta0);          //numerical aperture
10 //Result
11 printf("numerical aperture is %.5f",NA)
```