Scilab Textbook Companion for Engineering Physics by P. K. Palanisamy[1]

Created by
Ebby George
M.tech
Mechanical Engineering
NITK
College Teacher
None
Cross-Checked by
None

November 10, 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: Engineering Physics

Author: P. K. Palanisamy

Publisher: Scitech, Chennai

Edition: 1

Year: 2010

ISBN: 9788183714631

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 since the net change in energy

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 e=1.602*10^-19;    //charge of electron(c)
7 epsilon0=8.85*10^-12;     //permittivity(C/Nm)
8 r=3*10^-10;      //seperation(m)
9 N=6.022*10^20;
10 Ea=502;      //ionisation energy of A(kJ/mol)
11 Eb=-335;     //electron affinity for B(kJ/mol)
12
13 //Calculation
14 E=-e^2*N/(4*%pi*epsilon0*r);     //electrostatic
      attraction(kJ/mol)
15 nE=Ea+Eb+E;     //net change in energy per mole(kJ/
      mol)
16
17 //Result
18 printf("\n net change in energy per mole is %0.0f kJ
      /mol",nE)
```

```
19 printf("\n answer varies due to rounding off errors"
   )
20 printf("\n since the net change in energy is
   negative, the A+B- molecule will be stable")
```

Scilab code Exa 1.2 seperation

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  IPk=4.1;     //IP of K(eV)
7  EACl=3.6;     //EA of Cl(eV)
8  e=1.602*10^-19;    //charge of electron(c)
9  onebyepsilon0=9*10^9;
10
11 //Calculation
12 deltaE=IPk-EACl;
13 Ec=deltaE;       //energy required(eV)
14 R=e*onebyepsilon0/deltaE;      //seperation(m)
15
16 //Result
17 printf("\n energy required is %0.3f eV",Ec)
18 printf("\n seperation is %0.2f nm",R*10^9)
```

Scilab code Exa 1.3 bond energy

```
1  clc//
2  //
3  //
4
5  //Variable declaration
```

```
e=1.602*10^-19;    //charge of electron(c)
epsilon0=8.85*10^-12;     //permittivity(C/Nm)
r0=236*10^-12;      //seperation(m)
N=6.022*10^20;
IP=5.14;      //ionisation energy of A(kJ/mol)
EA=3.65;     //electron affinity for B(kJ/mol)

//Calculation
Ue=-e^2/(4*%pi*epsilon0*r0*e);     //potential energy
   (eV)
BE=-Ue-IP+EA;       //bond energy(eV)

//Result
printf("\n bond energy is %0.2f eV",BE)
```

Scilab code Exa 1.4 cohesive energy

```
clc//
//
//

//Variable declaration
A=1.748;        //madelung constant
n=9;      //born repulsive exponent
e=1.602*10^-19;    //charge of electron(c)
epsilon0=8.85*10^-12;     //permittivity(C/Nm)
r0=0.281*10^-9;      //seperation(m)
IE=5.14;      //ionisation energy of A(kJ/mol)
EA=3.61;     //electron affinity for B(kJ/mol)

//Calculation
CE=A*e^2*(1-(1/n))/(4*%pi*epsilon0*r0*e);       //
   cohesive energy(eV)

//Result
```

```
18 printf("\n cohesive energy is %0.3f eV",CE)
```

Chapter 2

Crystallography and Crystal Structures

Scilab code Exa 2.4 interplanar spacing for 212

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  r=0.1278;      //atomic radius(m)
7  h1=1;
8  k1=1;
9  l1=0;
10 h2=2;
11 k2=1;
12 l2=2;
13
14 //Calculation
15 a=(4*r/sqrt(2));
16
17 d110=a/sqrt(h1^2+k1^2+l1^2);     //interplanar
     spacing for (110)(nm)
18 d212=a/sqrt(h2^2+k2^2+l2^2);     //interplanar
```

```
    spacing for (212)(nm)
19
20 //Result
21 printf("\n interplanar spacing for (110) is %0.4f nm
    ",d110)
22 printf("\n interplanar spacing for (212) is %0.3f nm
    ",d212)
```

Scilab code Exa 2.5 seperation between successive lattice planes

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  h1=1;
7  k1=0;
8  l1=0;
9  h2=1;
10 k2=1;
11 l2=0;
12 h3=1;
13 k3=1;
14 l3=1;
15
16 //Calculation
17 d100=1/sqrt(h1^2+k1^2+l1^2);     //interplanar
    spacing for (110)
18 d110=1/sqrt(h2^2+k2^2+l2^2);     //interplanar
    spacing for (110)
19 d111=1/sqrt(h3^2+k3^2+l3^2);     //interplanar
    spacing for (111)
20
21 //Result
22 printf("\n seperation between successive lattice
```

```
planes is %0.3f : %0.2f : %0.2f ",d100,d110,d111)
```

Scilab code Exa 2.6 miller indices of plane

```
clc//
//
//

//Variable declaration
a=1;
b=1/2;
c=3;

//Calculation
A=1/a;
B=1/b;
C=1/c;
h=A*c;
k=B*c;
l=C*c;        //miller indices of plane

//Result
printf("\n miller indices of plane is ( %0.3f %0.3f
   %0.3f)",h,k,l)
```

Scilab code Exa 2.7 radius of interstitial sphere

```
clc//
//
//

//Variable declaration
r=1;     //assume
```

```
7
8  //Calculation
9  a=4/sqrt(3);
10 R=(a-(2*r))/2;        //radius of interstitial sphere(
       r)
11
12 //Result
13 printf("\n radius of interstitial sphere is %0.3f r"
       ,R)
```

Scilab code Exa 2.8 decrease of volume

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  r1=1.258;       //atomic radius(angstrom)
7  r2=1.292;       //atomic radius(angstrom)
8
9  //Calculation
10 a1=4*r1/sqrt(3);          //spacing(angstrom)
11 n1=((1/8)*8)+1;          //number of atoms per unit cell
12 v1=a1^3/n1;               //volume occupied by 1 atom(m
       ^3)
13 n2=(1/2*6)+(1/8*8);       //number of atoms per unit
       cell
14 a2=2*sqrt(2)*r2;      //spacing(angstrom)
15 v2=a2^3/n2;               //volume occupied by 1 atom(m
       ^3)
16 dc=(v1-v2)*100/v1;        //change in volume(%)
17
18 //Result
19 printf("\n decrease of volume is %0.1f percentage"
                                              ,dc)
```

Scilab code Exa 2.9 answer varies due to rounding off errors

```
clc//
//
//

//Variable declaration
a=0.27*10^-9;      //spacing(m)
c=0.494*10^-9;
n=6;     //number of atoms
M=65.37;     //atomic weight
N=6.023*10^26;     //avagadro number

//Calculation
V=3*sqrt(3)*a^2*c/2;      //volume of unit cell(m^3)
rho=n*M/(N*V);      //density of zinc(kg/m^3)

//Result
printf("\n volume of unit cell is %0.3f *10^-29 m^3"
   ,V*10^29)
printf("\n density of zinc is %0.0f kg/m^3",rho)
printf("\n answer varies due to rounding off errors"
   )
```

Scilab code Exa 2.10 radius of interstitial sphere

```
clc//
//
//

//Variable declaration
```

```
r=1;     //assume

//Calculation
a=4*r/sqrt(2);
R=(a/2)-r;       //radius of interstitial sphere(r)

//Result
printf("\n radius of interstitial sphere is %0.3f r"
   ,R)
```

Scilab code Exa 2.11 answer varies due to rounding off errors

```
clc//
//
//

//Variable declaration
a=0.356*10^-9;        //cube edge(m)
m=12.01;     //atomic weight of carbon
N=6.023*10^26;     //avagadro number

//Calculation
n=8/a^3;      //number of atoms per m^3
M=m/N;
d=M*n;        //density of diamond(kg/m^3)

//Result
printf("\n number of atoms per m^3 is %0.2f *10^29",
   n/10^29)
printf("\n density of diamond is %0.1f kg/m^3",d)
printf("\n answer varies due to rounding off errors"
   )
```

Scilab code Exa 2.12 distance between 2 adjacent atoms

```
clc//
//
//

//Variable declaration
mw=23+35.5;      //molecular weight of NaCl(gm/mol)
N=6.023*10^23;     //avagadro number(per mol)
d=2.18;         //mass of unit volume

//Calculation
M=mw/N;       //mass of NaCl molecule(gm)
n=2*d/M;        //number of atoms per unit volume(
   atoms/cm^3)
a=(1/n)^(1/3);       //distance between 2 adjacent
   atoms(cm)

//Result
printf("\n distance between 2 adjacent atoms is %e
   cm = %0.2f angstrom ",a,a*10^8)
```

Scilab code Exa 2.13 density of copper crystal

```
clc//
//
//

//Variable declaration
M=63.5;     //atomic weight
N=6.023*10^23;     //avagadro number
r=1.278*10^-8;     //radius(m)
n=4;

//Calculation
```

```
12 m=M/N;      //mass of copper atom(gm)
13 a=4*r/sqrt(2);
14 Mu=n*m;      //mass of unit cell
15 d=Mu/a^3;     //density of copper crystal(gm/cm^3)
16
17 //Result
18 printf("\n density of copper crystal is %0.3f gm/cm
      ^3",d)
```

Scilab code Exa 2.14 answer varies due to rounding off errors

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 r=0.1249*10^-9;      //radius(m)
7 pf=0.68;       //packing factor
8
9 //Calculation
10 a=4*r/sqrt(3);     //lattice constant(m)
11 v=a^3;      //volume of unit cell(m^3)
12 Fv=(1-pf)*v;      //free volume per unit cell(m^3)
13
14 //Result
15 printf("\n free volume per unit cell is %0.4f
      *10^-30 m^3",Fv*10^30)
16 printf("\n answer varies due to rounding off errors"
      )
```

Chapter 3

X ray Diffraction

Scilab code Exa 3.1 when thetais90 maximum order of diffraction possible

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 d=0.313;      //lattice spacing(m)
7 theta=7+(48/60);     //angle(degrees)
8 n=1;
9
10 //Calculation
11 theta=theta*%pi/180;     //angle(radian)
12 lamda=2*d*sin(theta)/n;     //wavelength of X-rays(nm
   )
13 //when theta=90
14 n=2*d/lamda;        //maximum order of diffraction
   possible
15
16 //Result
17 printf("\n wavelength of X-rays is %0.5f nm",lamda)
18 printf("\n answer varies due to rounding off errors"
   )
```

```
19 printf("\n when theta=90, maximum order of
      diffraction possible is %0.3f ",n)
```

Scilab code Exa 3.2 interatomic spacing

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  lamda=1.5418;       //wavelength(angstrom)
7  theta=30;       //angle(degrees)
8  n=1;     //first order
9  h=1;
10 k=1;
11 l=1;
12
13 //Calculation
14 theta=theta*%pi/180;     //angle(radian)
15 d=n*lamda/(2*sin(theta));
16 a=d*sqrt(h^2+k^2+l^2);     //interatomic spacing(
      angstrom)
17
18 //Result
19 printf("\n interatomic spacing is %0.2f angstrom",a)
```

Scilab code Exa 3.3 glancing angle

```
1  clc//
2  //
3  //
4
5  //Variable declaration
```

```
d100=0.28;     //spacing(nm)
lamda=0.071;     //wavelength of X rays(nm)
n=2;     //second order

//Calculation
d110=(d100/sqrt(2));      //spacing(nm)

x=n*lamda/(2*d110);
theta=asin(x);     //glancing angle(radian)
theta=theta*180/%pi ;      //glancing angle(degrees)

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

Scilab code Exa 3.4 distance between planes

```
clc//
//
//

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

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

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

Scilab code Exa 3.5 glancing angle

```
clc//
//
//

//Variable declaration
a=0.19;      //lattice constant(nm)
h=1;
k=1;
l=1;
lamda=0.058;     //wavelength of X rays(nm)
n=2;     //second order

//Calculation
d=a/sqrt(h^2+k^2+l^2);      //distance between planes
   (nm)
x=n*lamda/(2*d);
theta=asin(x);     //glancing angle(radian)
theta=theta*180/%pi ;      //glancing angle(degrees)

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

Chapter 4

Defects in Crystals

Scilab code Exa 4.1 ratio of vacancies

```
clc//
//
//

//Variable declaration
Ev=1;
k=1.38*10^-23;      //boltzmann constant(J/K)
e=1.6*10^-19;       //charge(eV)

//Calculation
r=Ev/(2.303*1000*k/e);
n=10^r;                 //ratio of n1000/n500

//Result
printf("\n ratio of vacancies is %0.3f *10^5",n
   /10^5)
```

Scilab code Exa 4.2 number of vacancies per atom at 500K

```
clc//
//
//

//Variable declaration
Ev=1.2;
k=1.38*10^-23;      //boltzmann constant(J/K)
e=1.6*10^-19;       //charge(eV)
T1=350;     //temperature(K)
T2=500;     //temperature(K)

//Calculation
x1=Ev/(2.303*k*T1/e);
n1=1/(10^x1);      //number of vacancies per atom at
   350K
x2=Ev/(2.303*k*T2/e);
n2=1/(10^x2);      //number of vacancies per atom at
   500K

//Result
printf("\n number of vacancies per atom at 350K is
   %0.4f *10^-17",n1*10^17)
printf("\n number of vacancies per atom at 500K is
   %0.3f *10^-12",n2*10^12)
```

Scilab code Exa 4.3 average energy required

```
clc//
//
//

//Variable declaration
d=2.82*10^-10;     //distance(m)
k=1.38*10^-23;      //boltzmann constant(J/K)
e=1.6*10^-19;       //charge(eV)
```

```
T=273+25;     //temperature(K)
sd=5*10^11;     //schotky defects(per m^3)

//Calculation
V=(2*d)^3;     //volume of unit cell(m^3)
N=4/V;     //density of ion pairs
x=(log10(N/sd));

Es=2*(k/e)*T*2.303*x;         //average energy
   required(eV)

//Result
printf("\n average energy required is %0.3f eV",Es)
```

Scilab code Exa 4.4 ratio of Frenkel defects

```
clc//
//
//

//Variable declaration
T1=273+25;       //temperature(K)
T2=273+350;      //temperature(K)
Ef=1.35;      //energy(eV)
k=8.625*10^-5;

//Calculation
x=(Ef/k)*((1/(2*T1))-(1/(2*T2)))/2.303;
r=1/(10^(x));       //ratio of Frenkel defects


//Result
printf("\n ratio of Frenkel defects is %0.3f *10^-6"
   ,r*10^6)
```

Chapter 5

Elements of statistical mechanics

Scilab code Exa 5.1 average thermal energy

```
clc//
//
//

//Variable declaration
k=1.38*10^-23;      //boltzmann constant(J)
T=300;          //temperature(K)
e=1.6*10^-19;       //charge(c)

//Calculation
E=3*k*T/(2*e);       //average thermal energy(eV)

//Result
printf("\n average thermal energy is %0.3f eV",E)
```

Scilab code Exa 5.3 fermi function

```
clc//
//
//

//Variable declaration
kT=1;     //assume
E_Ef=kT;

//Calculation
FE=1/(1+exp(1));      //fermi function

//Result
printf("\n fermi function is %0.3f ",FE)
```

Scilab code Exa 5.4 temperature

```
clc//
//
//

//Variable declaration
FE=10/100;      //fermi function
EF=5.5;      //energy function(eV)
e=1.6*10^-19;      //charge(c)
k=1.38*10^-23;     //boltzmann constant(J)

//Calculation
E=EF+(EF/100);     //energy(eV)
x=log((1/FE)-1);
T=(E-EF)*e/(k*x);      //temperature(K)

//Result
printf("\n temperature is %0.1f K",T)
```

Scilab code Exa 5.5 fermi velocity

```
clc//
//
//

//Variable declaration
k=1.38*10^-23;      //boltzmann constant(J)
T=24600;       //temperature(K)
m=9.108*10^-31;      //mass(kg)

//Calculation
vF=sqrt(2*k*T/m);       //fermi velocity(m s-1)

//Result
printf("\n fermi velocity is %0.2f *10^6 m s-1",vF
   /10^6)
```

Chapter 6

Principles of quantum mechanics

Scilab code Exa 6.1 de broglie wavelength

```
clc//
//
//

//Variable declaration
c=3*10^8;       //velocity of light(m/s)
m=1.67*10^-27;      //mass of proton(kg)
h=6.626*10^-34;     //planck's constant

//Calculation
lamda=h*10/(m*c);       //de broglie wavelength(m)

//Result
printf("\n de broglie wavelength is %0.3f *10^-14 m"
   ,lamda*10^14)
```

Scilab code Exa 6.2 de broglie wavelength

```
clc//
//
//

//Variable declaration
V=400;     //voltage(V)

//Calculation
lamda=12.26/sqrt(V);        //de broglie wavelength(
    angstrom)

//Result
printf("\n de broglie wavelength is %0.3f angstrom",
    lamda)
```

Scilab code Exa 6.3 de broglie wavelength

```
clc//
//
//

//Variable declaration
m=1.674*10^-27;       //mass of proton(kg)
h=6.626*10^-34;     //planck's constant
E=0.025*1.6*10^-19;     //energy(J)

//Calculation
lamda=h/sqrt(2*m*E);      //de broglie wavelength(m)

//Result
printf("\n de broglie wavelength is %0.3f nm",lamda
    *10^9)
```

Scilab code Exa 6.4 de broglie wavelength

```
clc//
//
//

//Variable declaration
V=1600;     //voltage(V)

//Calculation
lamda=12.26/sqrt(V);        //de broglie wavelength(
    angstrom)

//Result
printf("\n de broglie wavelength is %0.3f angstrom",
    lamda)
```

Scilab code Exa 6.5 uncertainity in momentum

```
clc//
//
//

//Variable declaration
deltax=0.2*10^-10;        //distance(m)
h=6.626*10^-34;     //planck's constant

//Calculation
deltap=h/(2*%pi*deltax);     //uncertainity in
    momentum(kg m/s)

//Result
```

```
13 printf("\n uncertainity in momentum is %0.2f *10^-24
       kg m/s",deltap*10^24)
```

Scilab code Exa 6.6 lowest energy of electron

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  n1=1;n2=1;n3=1;
7  h=6.62*10^-34;      //planck's constant
8  m=9.1*10^-31;       //mass(kg)
9  L=0.1*10^-9;        //side(m)
10
11 //Calculation
12 E1=h^2*(n1^2+n2^2+n3^2)/(8*m*1.6*10^-19*L^2);     //
       lowest energy of electron(eV)
13
14 //Result
15 printf("\n lowest energy of electron is %0.1f eV",E1
       )
```

Scilab code Exa 6.7 value of E122 E212 E221

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  n1=1;n2=1;n3=1;
7  h=6.62*10^-34;      //planck's constant
8  m=8.5*10^-31;       //mass(kg)
```

```
L=10^-11;       //side(m)

//Calculation
E111=h^2*(n1^2+n2^2+n3^2)/(8*m*1.6*10^-19*L^2);
   //lowest energy of electron(eV)
E112=6*h^2/(8*m*1.6*10^-19*L^2);       //value of
   E112(eV)
E121=E112;       //value of E121(eV)
E211=E112;       //value of E211(eV)
E122=9*h^2/(8*m*1.6*10^-19*L^2);      //value of E122
   (eV)
E212=E122;       //value of E212(eV)
E221=E122;       //value of E221(eV)

//Result
printf("\n lowest energy of electron is %0.3f *10^4
   eV",E111/10^4)
printf("\n value of E112, E121, E211 is %0.4f *10^4
   eV",E121/10^4)
printf("\n value of E122, E212, E221 is %0.3f *10^4
   eV",E122/10^4)
```

Scilab code Exa 6.8 de broglie wavelength

```
clc//
//
//

//Variable declaration
m=9.1*10^-31;      //mass of electron(kg)
h=6.626*10^-34;     //planck's constant
E=2000*1.6*10^-19;     //energy(J)

//Calculation
lamda=h/sqrt(2*m*E);      //de broglie wavelength(m)
```

```
12
13 //Result
14 printf("\n de broglie wavelength is %0.4f nm",lamda
   *10^9)
```

Scilab code Exa 6.9 lowest energy of electron

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 m=9.1*10^-31;      //mass of electron(kg)
7 h=6.626*10^-34;     //planck's constant
8 n=1;
9 L=4*10^-10;       //side(m)
10
11 //Calculation
12 E1=n^2*h^2/(8*m*L^2);     //lowest energy of electron
   (joule)
13
14
15 //Result
16 printf("\n lowest energy of electron is %0.3f
   *10^-18 joule",E1*10^18)
17 printf("\n answer varies due to rounding off errors"
   )
```

Scilab code Exa 6.10 energy of electron in 2nd state

```
1 clc//
2 //
3 //
```

```
4
5 //Variable declaration
6 m=9.1*10^-31;      //mass of electron(kg)
7 h=6.626*10^-34;     //planck's constant
8 n1=1;
9 n2=2;
10 n3=3;
11 L=1*10^-10;       //side(m)
12
13 //Calculation
14 E1=n1^2*h^2/(8*m*L^2);     //lowest energy of
      electron(joule)
15 E2=n2^2*h^2/(8*m*L^2);     //energy of electron in 1
      st state(joule)
16 E3=n3^2*h^2/(8*m*L^2);     //energy of electron in 2
      nd state(joule)
17
18 //Result
19 printf("\n lowest energy of electron is %0.4f
      *10^-17 joule",E1*10^17)
20 printf("\n energy of electron in 1st state is %0.3f
      *10^-17 joule",E2*10^17)
21 printf("\n energy of electron in 2nd state is %0.3f
      *10^-17 joule",E3*10^17)
```

Scilab code Exa 6.11 kinetic energy

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 m=9.1*10^-31;      //mass of electron(kg)
7 h=6.626*10^-34;     //planck's constant
8 lamda=1.66*10^-10;     //wavelength(m)
```

```
9
10 //Calculation
11 v=h/(m*lamda);      //velocity(m/s)
12 KE=(1/2)*m*v^2;      //kinetic energy(eV)
13
14 //Result
15 printf("\n velocity is %0.0f km/s",v/10^3)
16 printf("\n kinetic energy is %0.2f eV",KE
   /(1.6*10^-19))
```

Scilab code Exa 6.12 de broglie wavelength

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 V=15000;     //voltage(V)
7
8 //Calculation
9 lamda=12.26/sqrt(V);        //de broglie wavelength(
   angstrom)
10
11 //Result
12 printf("\n de broglie wavelength is %0.1f angstrom",
   lamda)
```

Scilab code Exa 6.13 spacing of crystal

```
1 clc//
2 //
3 //
4
```

```
5 //Variable declaration
6 V=344;     //voltage(V)
7 n=1;
8 theta=60*%pi/180;     //angle(radian)
9
10 //Calculation
11 lamda=(12.26/sqrt(V));       //de broglie wavelength(
      angstrom)
12
13 d=n*lamda/(2*sin(theta));     //spacing of crystal(
      angstrom)
14
15 //Result
16 printf("\n spacing of crystal is %0.4f angstrom",d)
```

Scilab code Exa 6.14 wavelength

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 E=1.5*9.1*10^-31;     //energy(joule)
7 m=1.676*10^-27;      //mass(kg)
8 h=6.62*10^-34;     //planck's constant
9
10 //Calculation
11 v=sqrt(2*E/m);
12 lamda=h/(m*v);        //wavelength(m)
13
14 //Result
15 printf("\n wavelength is %0.3f *10^-6 m",lamda*10^6)
16 printf("\n answer varies due to rounding off errors"
      )
```

Chapter 7

Band Theory of Solids

Scilab code Exa 7.1 mobility of electrons

```
clc//
//
//

//Variable declaration
rho_s=10.5*10^3;      //density(kg/m^3)
NA=6.02*10^26;        //avagadro number(per k mol)
MA=107.9;       //atomic mass
sigma=6.8*10^7;       //conductance(ohm-1 m-1)
e=1.6*10^-19;      //charge(coulomb)

//Calculation
n=rho_s*NA/MA;      //density of electrons
mew=sigma/(n*e);     //mobility of electrons(m^2/Vs)

//Result
printf("\n density of electrons is %0.2f *10^28",n
   /10^28)
printf("\n mobility of electrons is %0.3f *10^-2 m^2
    V-1 s-1",mew*10^2)
```

Scilab code Exa 7.2 average time of collision

```
clc//
//
//

//Variable declaration
d=8.92*10^3;      //density(kg/m^3)
rho=1.73*10^-8;      //resistivity of copper(ohm m)
NA=6.02*10^26;       //avagadro number(per k mol)
Aw=63.5;       //atomic weight
m=9.1*10^-31;      //mass(kg)
e=1.6*10^-19;      //charge(coulomb)

//Calculation
n=d*NA/Aw;      //density of electrons
mew=1/(rho*n*e);      //mobility of electrons(m^2/Vs)
t=m/(n*e^2*rho);      //average time of collision(s)

//Result
printf("\n mobility of electrons is %0.3f *10^-2 m V
   -1 s-1",mew*10^2)
printf("\n average time of collision is %0.2f
   *10^-14 s",t*10^14)
```

Scilab code Exa 7.3 relaxation time of conduction electrons

```
clc//
//
//

//Variable declaration
```

```
P=1.54*10^-8;       //resistance(ohm m)
n=5.8*10^28;        //number of electrons(per m^3)
m=9.108*10^-31;       //mass(kg)
e=1.602*10^-19;       //charge(coulomb)

//Calculation
t=m/(n*e^2*P);        //relaxation time of conduction
   electrons(s)

//Result
printf("\n relaxation time of conduction electrons
   is %0.2f *10^-14 s",t*10^14)
```

Scilab code Exa 7.4 drift velocity of electrons

```
clc//
//
//

//Variable declaration
R=0.06;     //resistance(ohm)
I=15;       //current(A)
D=5;      //length(m)
MA=26.98;         //atomic mass
rho_s=2.7*10^3;     //density(kg/m^3)
NA=6.025*10^26;         //avagadro number(per k mol)
e=1.602*10^-19;       //charge(coulomb)

//Calculation
n=3*rho_s*NA/MA;       //free electron concentration(
   electrons/m^2)
mew=1/(n*e*rho_s*10^-11);       //mobility(m s-1 V-1)
E=I*R/D;      //electric field(V/m)
vd=mew*E;     //drift velocity of electrons(m/s)

```

```
20 //Result
21 printf("\n free electron concentration is %0.4f
      *10^29 electrons/m^2",n/10^29)
22 printf("\n mobility is %0.3f *10^-3 m s-1 V-1",mew
      *10^3)
23 printf("\n drift velocity of electrons is %0.2f
      *10^-3 m s-1",vd*10^3)
```

Chapter 8

Semiconductors

Scilab code Exa 8.1 resistivity

```
clc//
//
//

//Variable declaration
ni=2.37*10^19;      //carrier density(per m^3)
mew_e=0.38;      //electron mobility(m^2/Vs)
mew_h=0.18;      //hole mobility(m^2/Vs)
e=1.6*10^-19;

//Calculation
sigma_i=ni*e*(mew_e+mew_h);
rho=1/sigma_i;       //resistivity(ohm m)

//Result
printf("\n resistivity is %0.3f ohm m",rho)
```

Scilab code Exa 8.2 position of fermi level

```
clc//
//
//

//Variable declaration
Eg=1.12;     //band gap(eV)
T=300;       //temperature(K)
m0=1;     //assume
me=0.12*m0;
mh=0.28*m0;
k=1.38*10^-23;    //boltzmann constant
e=1.6*10^-19;

//Calculation
EF=(Eg/2)+(3*k*T*log(mh/me)/(4*e));     //position of
    fermi level(eV)

//Result
printf("\n position of fermi level is %0.3f eV",EF)
```

Scilab code Exa 8.3 concentration of intrinsic charge carriers

```
clc//
//
//

//Variable declaration
T=300;       //temperature(K)
k=1.38*10^-23;    //boltzmann constant
m=9.109*10^-31;     //mass(kg)
h=6.626*10^-34;     //plancks constant
Eg=0.7;      //energy(eV)
e=1.6*10^-19;

//Calculation
```

```
14 x=(2*%pi*m*k/h^2)^(3/2);
15 y=exp(-Eg*e/(2*k*T));
16 ni=2*x*(T^(3/2))*y;                    //concentration of
      intrinsic charge carriers(per m^3)
17
18 //Result
19 printf("\n concentration of intrinsic charge
      carriers is %0.2f *10^18 per m^3",ni/10^18)
```

Scilab code Exa 8.4 resistivity

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 ni=2.4*10^19;      //carrier density(per m^3)
7 mew_e=0.39;      //electron mobility(m^2/Vs)
8 mew_h=0.19;      //hole mobility(m^2/Vs)
9 e=1.6*10^-19;
10
11 //Calculation
12 sigma_i=ni*e*(mew_e+mew_h);
13 rhoi=1/sigma_i;       //resistivity(ohm m)
14
15 //Result
16 printf("\n resistivity is %0.3f ohm m",rhoi)
```

Scilab code Exa 8.5 resistance

```
1 clc//
2 //
3 //
```

```
4
5 //Variable declaration
6 ni=2.5*10^19;      //carrier density(per m^3)
7 mew_e=0.39;       //electron mobility(m^2/Vs)
8 mew_p=0.19;       //hole mobility(m^2/Vs)
9 e=1.6*10^-19;
10 l=1*10^-2;       //length(m)
11 A=10^-3*10^-3;      //area(m^2)
12
13 //Calculation
14 R=l/(ni*e*A*(mew_p+mew_e));      //resistance(ohm)
15
16 //Result
17 printf("\n resistance is %0.2f *10^3 ohm",R/10^3)
```

Scilab code Exa 8.8 answer varies due to rounding off errors

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 e=1.6*10^-19;
7 mew_e=0.36;       //electron mobility(m^2/Vs)
8 mew_h=0.17;       //hole mobility(m^2/Vs)
9 rho=2.12;         //resistivity(ohm m)
10 T=300;        //temperature(K)
11 k=1.38*10^-23;    //boltzmann constant
12 m=9.109*10^-31;      //mass(kg)
13 h=6.626*10^-34;      //plancks constant
14
15 //Calculation
16 sigma=1/rho;
17 ni=sigma/(e*(mew_e+mew_h));
18 C=2*((2*%pi*m*k/h^2)^(3/2));
```

```
y=C*T^(3/2)/ni;
z=log(y);
Eg=2*k*T*z/(1.6*10^-19);          //forbidden energy
   gap(eV)

//Result
printf("\n forbidden energy gap is %0.3f eV",Eg)
printf("\n answer varies due to rounding off errors"
   )
```

Scilab code Exa 8.9 answer varies due to rounding off errors

```
clc//
//
//

//Variable declaration
x=0.6532;
y=0.3010;
T1=273+20;      //temperature(K)
T2=273+32;      //temperature(K)
k=8.616*10^-5;

//Calculation
dy=x-y;
dx=(1/T1)-(1/T2);
Eg=2*k*dy/dx;       //energy band gap(eV)

//Result
printf("\n energy band gap is %0.3f eV",Eg)
printf("\n answer varies due to rounding off errors"
   )
```

Scilab code Exa 8.10 temperature

```
clc//
//
//

//Variable declaration
k=1.38*10^-23;    //boltzmann constant
EF=0.18;         //fermi shift(eV)
E=1.2;          //energy gap(eV)
e=1.6*10^-19;
r=5;

//Calculation
T=EF*e*4/(3*k*log(r));      //temperature(K)

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

Scilab code Exa 8.11 electron concentration

```
clc//
//
//

//Variable declaration
Na=5*10^23;       //number of atoms(atoms)
Nd=3*10^23;       //number of atoms(atoms)
ni=2*10^16;      //intrinsic charge carriers(per m^3)

//Calculation
p=2*(Na-Nd)/2;      //hole concentration(per m^3)
n=ni^2/p;         //electron concentration(per m^3)

//Result
```

```
15 printf("\n electron concentration is %0.3f *10^9 per
       m^3",n/10^9)
```

Scilab code Exa 8.12 conductivity

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  ni=1.5*10^16;      //carrier density(per m^3)
7  mew_e=0.13;       //electron mobility(m^2/Vs)
8  mew_h=0.05;       //hole mobility(m^2/Vs)
9  e=1.6*10^-19;
10 d=2.33*10^3;      //density(kg/m^3)
11 n=28.1;
12 na=6.02*10^26;      //number of atoms
13
14 //Calculation
15 sigma=ni*e*(mew_e+mew_h);      //conductivity(ohm-1 m
       -1)
16 Nd=d*na/(n*10^8);
17 p=ni^2/Nd;
18 sigma_ex1=Nd*e*mew_e;       //conductivity(ohm-1 m-1)
19 n=p;
20 Na=Nd;
21 sigma_ex2=Na*e*mew_h;       //conductivity(ohm-1 m-1)
22
23 //Result
24 printf("\n conductivity is %0.3f *10^-3 ohm-1 m-1",
       sigma*10^3)
25 printf("\n conductivity is %0.2f ohm-1 m-1",
       sigma_ex1)
26 printf("\n conductivity is %0.2f ohm-1 m-1",
       sigma_ex2)
```

Scilab code Exa 8.13 position of fermi level

```
clc//
//
//

//Variable declaration
ni=1.5*10^16;       //carrier density(per m^3)
mew_e=0.135;       //electron mobility(m^2/Vs)
mew_h=0.048;       //hole mobility(m^2/Vs)
e=1.6*10^-19;
Nd=10^23;
T=300;     //temperature(K)
k=1.38*10^-23;

//Calculation
sigma=ni*e*(mew_e+mew_h);       //conductivity(ohm-1 m
   -1)
p=ni^2/Nd;       //hole concentration(per m^3)
sigma_ex=Nd*e*mew_e;       //conductivity(ohm-1 m-1)
x=3*k*T*log(mew_e/mew_h)/4;

//Result
printf("\n conductivity is %0.3f *10^-3 ohm-1 m-1",
   sigma*10^3)
printf("\n hole concentration is %0.3f per m^3",p)
printf("\n conductivity is %0.3f *10^3 ohm-1 m-1",
   sigma_ex/10^3)
printf("\n position of fermi level is %0.2f eV",x
   /(1.6*10^-19))
```

Scilab code Exa 8.14 answer varies due to rounding off errors

```
clc//
//
//

//Variable declaration
mew_e=0.19;      //electron mobility(m^2/Vs)
e=1.6*10^-19;
T=300;     //temperature(K)
k=1.38*10^-23;

//Calculation
Dn=mew_e*k*T/e;      //diffusion coefficient(m^2 s-1)

//Result
printf("\n diffusion coefficient is %0.3f *10^-4 m^2
    s-1",Dn*10^4)
printf("\n answer varies due to rounding off errors"
   )
```

Scilab code Exa 8.15 hall voltage

```
clc//
//
//

//Variable declaration
RH=3.66*10^-4;    //hall coefficient(m^3/coulomb)
I=10^-2;      //current(amp)
B=0.5;      //magnetic field(wb/m^2)
t=1*10^-3;     //thickness(m)

//Calculation
VH=RH*I*B*10^3/t;      //hall voltage(mV)

//Result
```

```
15 printf("\n hall voltage is %0.3f mV",VH)
```

Scilab code Exa 8.16 hall coefficient

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  Vy=37*10^-6;      //voltage(V)
7  t=10^-3;     //thickness(m)
8  Bz=0.5;       //magnetic field(wb/m^2)
9  Ix=20*10^-3;    //current(A)
10
11 //Calculation
12 RH=Vy*t/(Ix*Bz);      //hall coefficient(m^3/coulomb)
13
14 //Result
15 printf("\n hall coefficient is %e C-1 m^3",RH)
```

Scilab code Exa 8.17 mobility of charge carriers

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  RH=6.85*10^-5;    //hall coefficient(m^3/coulomb)
7  e=1.6*10^-19;
8  sigma=250;       //conductivity(m-1 ohm-1)
9
10 //Calculation
11 n=1/(RH*e);        //density of charge carriers(m^3)
```

```
12 mew=sigma/(n*e);     //mobility of charge carriers(m
    ^2/Vs)
13
14 //Result
15 printf("\n density of charge carriers is %0.3f
    *10^22 m^3",n/10^22)
16 printf("\n mobility of charge carriers is %0.3f
    *10^-3 m^2 V-1 s-1",mew*10^3)
```

Scilab code Exa 8.18 hall voltage

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 I=30;     //current(A)
7 B=1.75;    //magnetic field(T)
8 n=6.55*10^28;     //electron concentration(/m^3)
9 t=0.35*10^-2;     //thickness(m)
10 e=1.6*10^-19;
11
12 //Calculation
13 VH=I*B*10^6/(n*e*t);      //hall voltage(micro V)
14
15 //Result
16 printf("\n hall voltage is %0.3f micro V",VH)
```

Scilab code Exa 8.19 mobility of charge carriers

```
1 clc//
2 //
3 //
```

```
4
5 //Variable declaration
6 RH=3.66*10^-4;    //hall coefficient(m^3/coulomb)
7 e=1.6*10^-19;
8 Pn=8.93*10^-3;     //resistivity(ohm m)
9
10 //Calculation
11 n=1/(RH*e);       //density of charge carriers(per m
     ^3)
12 mew_e=RH/Pn;     //mobility of charge carriers(m^2/Vs
     )
13
14 //Result
15 printf("\n density of charge carriers is %0.3f
     *10^22 per m^3",n/10^22)
16 printf("\n mobility of charge carriers is %0.3f m^2
     V-1 s-1",mew_e)
```

Chapter 9

Physics of Semiconductor Devices

Scilab code Exa 9.1 wavelength of radiation

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  h=6.62*10^-34;      //planck's constant(J sec)
7  c=3*10^8;      //velocity of light(m/sec)
8  Eg=1.43*1.6*10^-19;     //energy gap(J)
9
10 //Calculation
11 lamda=h*c*10^6/Eg;     //wavelength of radiation(
       micro m)
12
13 //Result
14 printf("\n wavelength of radiation is %0.3f micro m"
       ,lamda)
```

Scilab code Exa 9.2 time taken

```
clc//
//
//

//Variable declaration
d=5*10^-6;      //thickness(m)
Dc=3.4*10^-3;     //diffusion coefficient(m^2 S-1)

//Calculation
tow_diff=d^2/(2*Dc);      //time taken(s)

//Result
printf("\n time taken is %0.1f *10^-9 s",tow_diff
   *10^9)
```

Scilab code Exa 9.3 transit time

```
clc//
//
//

//Variable declaration
w=5*10^-6;      //thickness(m)
vsat=10^5;      //velocity(m/s)

//Calculation
tow_drift=w/vsat;      //transit time(s)

//Result
printf("\n transit time is %0.3f s",tow_drift)
```

Scilab code Exa 9.4 frequency bandwidth

```
clc//
//
//

//Variable declaration
A=10^-6;      //area(m^2)
e=1.6*10^-19;      //charge(coulomb)
Nd=10^21;      //electron concentration(m^-3)
epsilonr=11.7;
epsilon0=8.85*10^-12;
V=10;     //potential(V)
RL=50;    //resistance(ohm)

//Calculation
Cj=(A/2)*sqrt(2*e*epsilonr*epsilon0*Nd/V);       //
   diode capacitance(F)
delta_fel=1/(2*%pi*RL*Cj);     //frequency bandwidth(
   Hz)

//Result
printf("\n diode capacitance is %0.1f pF",Cj*10^12)
printf("\n frequency bandwidth is %0.0f MHz",
   delta_fel*10^-6)
printf("\n answer varies due to rounding off errors"
   )
```

Chapter 10

Dielectric properties

Scilab code Exa 10.1 energy stored in the dielectric

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  C=2*10^-6;     //capacitance(F)
7  V=1000;       //voltage(V)
8  epsilon_r=100;
9
10 //Calculation
11 W=C*V^2/2;     //energy stored in the condenser(J)
12 C0=C/epsilon_r;
13 W0=C0*V^2/2;
14 E=1-W0;        //energy stored in the dielectric(J)
15
16 //Result
17 printf("\n energy stored in the condenser is %0.3f J
      ",W)
18 printf("\n energy stored in the dielectric is %0.3f
      J",E)
```

Scilab code Exa 10.2 ratio betwen electronic and ionic polarizability

```
clc//
//
//

//Variable declaration
epsilon_r=4.94;
n2=2.69;

//Calculation
x=(epsilon_r-1)/(epsilon_r+2);
y=(n2-1)/(n2+2);
r=(x/y)-1;          //ratio betwen electronic and ionic
    polarizability

//Result
printf("\n ratio betwen electronic and ionic
   polarizability is %0.3f ",1/r)
```

Scilab code Exa 10.3 parallel loss capacitance

```
clc//
//
//

//Variable declaration
epsilon_r=2.56;
epsilon_R=2.65*0.7*10^-4;
tan_delta=0.7*10^-4;
A=8*10^-4;      //area(m^2)
d=0.08*10^-3;      //diameter(m)
```

```
11 f=1*10^6;     //frequency(Hz)
12 epsilon0=8.85*10^-12;
13
14 //Calculation
15 Rp=d/(2*%pi*f*epsilon0*epsilon_R*A);     //parallel
      loss resistance(ohm)
16 Cp=A*epsilon0*epsilon_r/d;      //parallel loss
      capacitance(Farad)
17
18 //Result
19 printf("\n parallel loss resistance is %0.0f ohm",Rp
      /10^6)
20 printf("\n answer varies due to rounding off errors"
      )
21 printf("\n parallel loss capacitance is %0.2f
      *10^-12 Farad",Cp*10^12)
```

Scilab code Exa 10.4 dielectric constant of material

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 N=3*10^28;      //number of atoms(per m^3)
7 alphae=10^-40;
8 epsilon0=8.854*10^-12;
9
10 //Calculation
11 epsilon_r=1+(N*alphae/epsilon0);    //dielectric
      constant of material
12
13 //Result
14 printf("\n dielectric constant of material is %0.3f
      ",epsilon_r)
```

Scilab code Exa 10.5 electronic polarizability

```
clc//
//
//

//Variable declaration
N=2.7*10^25;      //number of atoms(per m^3)
epsilon0=8.854*10^-12;
epsilon_r=1.0000684;

//Calculation
alphae=epsilon0*(epsilon_r-1)/N;     //electronic
   polarizability(Fm^2)

//Result
printf("\n electronic polarizability is %0.3f
   *10^-41 Fm^2",alphae*10^41)
```

Scilab code Exa 10.6 charge on plates

```
clc//
//
//

//Variable declaration
epsilon0=8.85*10^-12;
A=100*10^-4;      //area(m^2)
d=10^-2;      //diameter(m)
V=100;        //potential(V)

```

```
11 //Calculation
12 C=epsilon0*A/d;     //capacitance(F)
13 Q=C*V;          //charge on plates(coulomb)
14
15 //Result
16 printf("\n capacitance is %e F",C)
17 printf("\n charge on plates is %e coulomb",Q)
```

Scilab code Exa 10.7 electronic polarizability

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 n=6.02*10^26;      //avagadro number
7 d=2050;      //density(kg/m^3)
8 w=32;        //atomic weight
9 gama=1/3;     //internal field constant
10 epsilon0=8.55*10^-12;
11 epsilon_r=3.75;
12
13 //Calculation
14 N=n*d/w;        //number of atoms(per m^3)
15 alphae=3*epsilon0*((epsilon_r-1)/(epsilon_r+2))/N;
         //electronic polarizability(Fm^2)
16
17 //Result
18 printf("\n electronic polarizability is %0.3f
   *10^-40 Fm^2",alphae*10^40)
```

Scilab code Exa 10.8 resultant voltage

```
clc//
//
//

//Variable declaration
Q=2*10^-10;     //charge(C)
d=4*10^-3;      //seperation(m)
epsilon_r=3.5;
A=650*10^-6;     //area(m^2)
epsilon0=8.85*10^-12;

//Calculation
V=Q*d/(epsilon0*epsilon_r*A);       //resultant
   voltage(V)

//Result
printf("\n resultant voltage is %0.2f Volts",V)
```

Scilab code Exa 10.9 dielectric displacement

```
clc//
//
//

//Variable declaration
d=2*10^-3;      //seperation(m)
epsilon_r=6;
V=10;       //voltage(V)
epsilon0=8.85*10^-12;

//Calculation
E=V/d;
D=epsilon0*epsilon_r*E;       //dielectric
   displacement(C m^-2)

```

```
15 //Result
16 printf("\n dielectric displacement is %0.1f *10^-9 C
       m^-2",D*10^9)
```

Chapter 11

Magnetic properties

Scilab code Exa 11.1 relative permeability of iron

```
clc//
//
//

//Variable declaration
M=1.4;      //magnetic field(T)
H=6.5*10^-4;   //magnetic field(T)

//Calculation
chi=M/H;
mew_r=1+chi;       //relative permeability of iron

//Result
printf("\n relative permeability of iron is %0.3f ",
   mew_r)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 11.2 relative permeability

```
clc//
//
//

//Variable declaration
M=3300;      //magnetic field(amp/m)
H=220;   //magnetic field(amp/m)

//Calculation
chi=M/H;
mew_r=1+chi;       //relative permeability

//Result
printf("\n relative permeability is %0.3f ",mew_r)
```

Scilab code Exa 11.3 flux density

```
clc//
//
//

//Variable declaration
H=10^6;   //magnetic field(amp/m)
chi=1.5*10^-3;
mew0=4*%pi*10^-7;

//Calculation
M=chi*H;      //magnetisation of material(A/m)
B=mew0*(M+H);      //flux density(T)

//Result
printf("\n magnetisation of material is %0.3f *10^3
   A/m",M/10^3)
printf("\n flux density is %0.4f T",B)
printf("\n answer given in the book varies due to
```

```
rounding off errors")
```

Scilab code Exa 11.4 flux density

```
clc//
//
//

//Variable declaration
H=10^4;  //magnetic field(amp/m)
chi=3.7*10^-3;
mew0=4*%pi*10^-7;

//Calculation
M=chi*H;      //magnetisation of material(A/m)
B=mew0*(M+H);      //flux density(T)

//Result
printf("\n magnetisation of material is %0.3f A/m",M
   )
printf("\n flux density is %0.4f wb/m^2",B)
```

Scilab code Exa 11.5 magnetic moment

```
clc//
//
//

//Variable declaration
r=5*10^-2       //radius(m)
I=500*10^-3;     //current(A)

//Calculation
```

```
10 A=2*%pi*r^2;
11 mew_m=I*A;      //magnetic moment(Am^2)
12
13 //Result
14 printf("\n magnetic moment is %0.3f *10^-3 Am^2",
      mew_m*10^3)
15 printf("\n answer given in the book varies due to
      rounding off errors")
```

Scilab code Exa 11.6 change in magnetic moment

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 r=5.29*10^-11;      //radius(m)
7 B=2;      //magnetic field(T)
8 e=1.602*10^-19;    //charge(c)
9 m=9.108*10^-31;     //mass(kg)
10
11 //Calculation
12 mew_ind=e^2*r^2*B/(4*m);       //change in magnetic
      moment(Am^2)
13
14 //Result
15 printf("\n change in magnetic moment is %0.3f
      *10^-29 Am^2",mew_ind*10^29)
```

Scilab code Exa 11.7 susceptibility

```
1 clc//
2 //
```

```
3 //
4
5 //Variable declaration
6 chi1=2.8*10^-4;      //susceptibility
7 T1=350;             //temperature(K)
8 T2=300;             //temperature(K)
9
10 //Calculation
11 chi2=chi1*T1/T2;      //susceptibility
12
13 //Result
14 printf("\n susceptibility is %0.3f *10^-4",chi2
   *10^4)
```

Scilab code Exa 11.8 magnetic moment

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 Bs=0.65;       //magnetic induction(wb/m^2)
7 d=8906;        //density(kg/m^3)
8 n=6.025*10^26;      //avagadro number
9 mew0=4*%pi*10^-7;
10 w=58.7;      //atomic weight(kg)
11
12 //Calculation
13 N=d*n/w;      //number of nickel atoms(per m^3)
14 mew_m=Bs/(N*mew0*9.27*10^-24);      //magnetic moment
   (mewB)
15
16 //Result
17 printf("\n magnetic moment is %0.2f mewB",mew_m)
```

Scilab code Exa 11.9 temperature

```
clc//
//
//

//Variable declaration
mew=9.4*10^-24;
H=2;        //magnetic field(weber/m^2)
k=1.38*10^-23;      //boltzmann constant

//Calculation
T=2*mew*H/(log(2)*k);      //temperature(K)

//Result
printf("\n temperature is %0.1f K",T)
```

Scilab code Exa 11.10 magnetic moment per gram

```
clc//
//
//

//Variable declaration
d=7.8*10^3;      //density(kg/m^3)
n=6.025*10^26;      //number of atoms
w=157.26;       //atomic weight(kg)
mewm=9.27*10^-24;
mew=7.1*mewm;
mew0=4*%pi*10^-7;

//Calculation
```

```
14 N=d*n/w;          //number of atoms
15 mew_B=N*mew/10^3;      //magnetic moment per gram(Am
       ^2)
16 Bs=N*mew0*mew;
17
18 //Result
19 printf("\n magnetic moment per gram %0.3f Am^2",
       mew_B)
20 printf("\n magnetic moment per gram is %0.4f Wb/m^2"
       ,Bs)
21 printf("\n answer given in the book varies due to
       rounding off errors")
```

Scilab code Exa 11.11 critical field

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 Tc=3.7;      //temperature(K)
7 Hc0=0.0306;     //critical field(T)
8 T=2;        //temperature(K)
9
10 //Calculation
11 Hc2=Hc0*(1-(T/Tc)^2);       //critical field(T)
12
13 //Result
14 printf("\n critical field is %0.5f Tesla",Hc2)
```

Scilab code Exa 11.12 critical current

```
1 clear//
```

```
2 //
3 //
4
5 //Variable declaration
6 Tc=7.18;      //temperature(K)
7 H0=6.5*10**4;     //critical field(T)
8 T=4.2;       //temperature(K)
9 d=1*10**-3;     //diameter(m)
10
11 //Calculation
12 Hc=H0*(1-(T/Tc)**2);       //critical field(T)
13 ic=%pi*d*Hc;        //critical current(A)
14
15 //Result
16 printf("\n critical current is %0.2f A",ic)
17 printf("\n answer given in the book is wrong")
```

Chapter 12

Lasers

Scilab code Exa 12.1 relative population

```
clc//
//
//

//Variable declaration
e=1.6*10^-19;       //charge(coulomb)
h=6.6*10^-34;       //planck's constant(J sec)
c=3*10^8;       //velocity of light(m/sec)
lamda=6943*10^-10;       //wavelength(m)
k=8.61*10^-5;
T=300;      //temperature(K)

//Calculation
dE=h*c/(e*lamda);
N2byN1=exp(-dE/(k*T));       //relative population

//Result
printf("\n relative population is %0.4f *10^-30",
   N2byN1*10^30)
printf("\n answer given in the book is wrong")
```

Scilab code Exa 12.2 divergence

```
clc//
//
//

//Variable declaration
a1=4*10^-3;       //diameter(m)
a2=6*10^-3;       //diameter(m)
d1=1;      //distance(m)
d2=2;      //distance(m)

//Calculation
theta=(a2-a1)/(2*(d2-d1));      //divergence(radian)

//Result
printf("\n divergence is %0.3f milli radian",theta
   *10^3)
```

Scilab code Exa 12.3 spot size

```
clc//
//
//

//Variable declaration
s=1*10^-3;     //size(m)
l=1*10^-3;     //length(m)
lamda=650*10^-9;     //wavelength(m)

//Calculation
tantheta=(l/2)/s;
```

```
12 theta=atan(tantheta);      //angle(radian)
13 sintheta=(sin(theta));
14
15 ss=0.6*lamda/sintheta;       //spot size(m)
16
17 //Result
18 printf("\n spot size is %0.3f micro m",ss*10^6)
```

Chapter 13

Fiber Optics

Scilab code Exa 13.1 numerical aperture

```
clc//
//
//

//Variable declaration
n1=1.55;      //refractive index of core
n2=1.50;      //refractive index of cladding

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

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

Scilab code Exa 13.2 angle of acceptance

```
clc//
//
```

```
3 //
4
5 //Variable declaration
6 n1=1.563;      //refractive index of core
7 n2=1.498;      //refractive index of cladding
8
9 //Calculation
10 NA=sqrt(n1^2-n2^2);     //numerical aperture
11 alpha_i=asin(NA);       //angle of acceptance(radian)
12 alpha_i=(alpha_i*180/%pi);     //angle(degrees)
13 alpha_id=int(alpha_i);
14 alpha_im=60*(alpha_i-alpha_id);
15
16 //Result
17 printf("\n angle of acceptance is %0.3f degrees %0.1
     f minutes",alpha_id,alpha_im)
18 printf("\n answer varies due to rounding off errors"
     )
```

Scilab code Exa 13.3 refractive index of core

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 NA=0.39;        //numerical aperture
7 delta=0.05;     //difference of indices
8
9 //Calculation
10 n1=NA/sqrt(2*delta);      //refractive index of core
11
12 //Result
13 printf("\n refractive index of core is %0.4f ",n1)
14 printf("\n answer varies due to rounding off errors"
```

```
)
```

Scilab code Exa 13.4 fractional index change

```
clc//
//
//

//Variable declaration
n1=1.563;      //refractive index of core
n2=1.498;      //refractive index of cladding

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

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

Scilab code Exa 13.5 angle of acceptance

```
clc//
//
//

//Variable declaration
n1=1.48;      //refractive index of core
n2=1.45;      //refractive index of cladding

//Calculation
NA=sqrt(n1^2-n2^2);     //numerical aperture
alpha_i=asin(NA);       //angle of acceptance(radian)
alpha_i=(alpha_i*180/%pi);     //angle(degrees)
alpha_id=int(alpha_i);
```

```
14 alpha_im=60*(alpha_i-alpha_id);
15
16 //Result
17 printf("\n numerical aperture is %0.4f ",NA)
18 printf("\n angle of acceptance is %0.3f degrees %0.0
     f minutes",alpha_id,alpha_im)
```

Scilab code Exa 13.6 attenuation loss

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 Pout=40;      //power(mW)
7 Pin=100;      //power(mW)
8
9 //Calculation
10 al=-10*log10(Pout/Pin);     //attenuation loss(dB)
11
12 //Result
13 printf("\n attenuation loss is %0.2f dB",al)
```

Chapter 14

Acoustics of buildings and acoustic quieting

Scilab code Exa 14.1 reverberation time of hall

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  V=475;      //volume(m^3)
7  aw=200;     //area of wall(m^2)
8  ac=100;     //area of ceiling(m^2)
9  ac_w=0.025;     //absorption coefficient of wall
10 ac_c=0.02;     //absorption coefficient of ceiling
11 ac_f=0.55;     //absorption coefficient of floor
12
13 //Calculation
14 sigma_as=(aw*ac_w)+(ac*ac_c)+(ac*ac_f);
15 T=0.165*V/sigma_as;          //reverberation time of
       hall(s)
16
17 //Result
18 printf("\n reverberation time of hall is %0.3f s",T)
```

Scilab code Exa 14.2 new reverberation time

```
clc//
//
//

//Variable declaration
V=12500;      //volume(m^3)
T1=1.5;       //reverberation time(sec)
n=200;     //number of cushioned chairs

//Calculation
sigma_as=0.165*V/T1;
T2=0.165*V/(sigma_as+n);      //new reverberation
   time(s)

//Result
printf("\n new reverberation time is %0.2f s",T2)
```

Scilab code Exa 14.3 total absorption in the hall

```
clc//
//
//

//Variable declaration
V=5000;     //volume(m^3)
T=1.25;     //time(s)

//Calculation
sigma_as=0.165*V/T;           //total absorption in
   the hall(OWU)
```

```
11
12 //Result
13 printf("\n total absorption in the hall is %0.3f OWU
   ",sigma_as)
```

Scilab code Exa 14.4 new period of reverberation

```
1  clc//
2  //
3  //
4
5  //Variable declaration
6  V=9500;     //volume(m^3)
7  T=1.5;     //time(s)
8  x=100;     //absorption(sabines)
9
10 //Calculation
11 sigma_as=0.165*V/T;           //total absorption in
   the hall(OWU)
12 T=0.165*V/(sigma_as+x);       //new period of
   reverberation(s)
13
14 //Result
15 printf("\n total absorption in the hall is %0.3f OWU
   ",sigma_as)
16 printf("\n new period of reverberation is %0.3f s",T
   )
```

Scilab code Exa 14.5 average absorption coefficient

```
1  clc//
2  //
3  //
```

```
4
5 //Variable declaration
6 V=20*15*5;      //volume(m^3)
7 T=3.5;      //time(s)
8 A=950;      //surface area(m^2)
9
10 //Calculation
11 sigma_as=0.165*V/T;           //total absorption in
      the hall(OWU)
12 ac=sigma_as/A;         //average absorption
      coefficient
13
14 //Result
15 printf("\n total absorption in the hall is %0.3f OWU
      ",sigma_as)
16 printf("\n average absorption coefficient is %0.3f
      sabine/m^2",ac)
```

Scilab code Exa 14.6 number of persons to be seated

```
1 clc//
2 //
3 //
4
5 //Variable declaration
6 V=2265;      //volume(m^3)
7 sigma_as=92.9;      //absorption(m^2)
8 a=18.6;       //area(m^2)
9
10 //Calculation
11 T=0.165*V/sigma_as;           //reverberation time of
      hall(s)
12 T1=0.165*V/2;
13 inc=T1-sigma_as;         //increase in absorption(OWU)
14 n=inc/a;          //number of persons to be seated
```

```
15
16 //Result
17 printf("\n reverberation time of hall is %0.3f s",T)
18 printf("\n number of persons to be seated is %0.3f "
      ,n)
```