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

Created by
G Krishna Chaitanya
B.Tech
Computer Engineering
CVSR College of Engineering
College Teacher
NA
Cross-Checked by
K. V. P. Pradeep

June 3, 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: P. K. Palanisamy

Publisher: Scitech Publications(india)

Edition: 8

Year: 2009

ISBN: 9788183710985

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 To determine whether the gaseous molecule is stable

```
1 //To determine whether the gaseous molecule is
    stable
2 IE_1 = 502;      //first ionisation energy, kJ/mol
3 EA_B = -335;     //electron affinity for B atom, kJ/
    mol
4 e = 1.602*10^-19;
5 r = 0.3;       //inter ionic seperation, nm
6 r = r*10^-9;       //inter ionic seperation, m
7 N = 6.022*10^23*10^-3;
8 epsilon0 = 8.85*10^-12;       //permittivity of free
    space, C/N-m
9 E = (-e^2*N)/(4*%pi*epsilon0*r);       //
    electrostatic attraction energy, kJ/mol
10 printf("electrostatic attraction energy is %d kJ/mol
    ",E);
11 dE = IE_1+EA_B+E;      //net change in energy per
    mole
12 printf("net change in energy is %d kJ/mol",dE);
13 printf("since the net change in energy is negative,
    A+B- molecule will be stable");
14
```

```
15 //answer for net change, dE given in the book is
     wrong
```

Scilab code Exa 1.2 To calculate the energy required and seperation between ion pair

```
1 //To calculate the energy required and seperation
     between ion pair
2 IP_K = 4.1;      //IP of K, eV
3 EA_Cl = 3.6;      //EA of Cl, eV
4 e = 1.602*10^-19;
5 epsilon0 = 8.85*10^-12;
6 delta_E = IP_K - EA_Cl;       //energy required, eV
7 printf("energy required to form K+Cl- ion pair is %f
     eV",delta_E);
8 //if their total energy is 0, delta_E = Ec
9 Ec = delta_E;
10 R = e/(4*%pi*epsilon0*Ec);       //seperation between
     ion pair, m
11 R = R*10^9;           //seperation between ion pair,
     nm
12 printf("seperation between ion pair is %5.2f nm",R);
```

Scilab code Exa 1.3 To calculate the bond energy for NaCl molecule

```
1 //To calculate the bond energy for NaCl molecule
2 e = 1.602*10^-19;
3 EA = 3.65;      //electron affinity of Cl, eV
4 IP = 5.14;       //ionisation energy of Na, eV
5 epsilon0 = 8.85*10^-12;
6 r0 = 236;       //equilibrium distance, pm
7 r0 = r0*10^-12;      //equilibrium distance, m
```

```
8 V = (-e^2)/(4*%pi*epsilon0*r0);        //potential
    energy, J
9 V = V/e;          //potential energy, eV
10 Ue = V;
11 BE = -Ue - IP + EA;          //bond energy, eV
12 printf("bond energy for NaCl molecule is %5.2f eV",
    BE);
```

Scilab code Exa 1.4 To calculate the cohesive energy of NaCl

```
1 //To calculate the cohesive energy of NaCl
2 e = 1.602*10^-19;
3 epsilon0 = 8.85*10^-12;
4 r0 = 0.281;       //equilibrium seperation, nm
5 r0 = r0*10^-9;       //equilibrium seperation, m
6 A = 1.748;       //Madelung constant
7 n = 9;     //born repulsive exponent
8 CE = (A*e^2)*(1-(1/n))/(4*%pi*epsilon0*r0);     //
    cohesive energy, J
9 CE = CE/e;          //cohesive energy, eV
10 printf("cohesive energy of NaCl is %5.3f eV",CE);
```

Chapter 2

Crystal Structures and X ray diffraction

Scilab code Exa 2.1 To calculate the maximum radius of the interstitial sphere

```
1 //To calculate the maximum radius of the
    interstitial sphere
2 //We know that body diagonal = 4*r = sqrt(3).a
3 //hence a = (4/sqrt(3)).r
4 //R = (a-2*r)/2
5 //R = ((4/(2*sqrt(3)))*r) - (2*r/2) = (2*r/sqrt(3))
    - r
6 //R = 1.155*r - r = 0.155*r
7 printf("The maximum radius of the interstitial
    sphere is 0.155*r");
```

Scilab code Exa 2.2 To calculate the percent volume change

```
1 //To calculate the percent volume change
2 r_BCC = 1.258;      //atomic radius, A
```

```
3  r_FCC = 1.292;      //atomic radius, A
4  a_BCC = (4*r_BCC)/sqrt(3);      //In BCC, A
5  a_BCCm = a_BCC*10^-10;       //converting a from A to
       m
6  V_BCC = a_BCCm^3;      //volume of unit cell, m^3
7  n_BCC = ((1/8)*8)+1;       //number of atoms per unit
       cell
8  V1_BCC = V_BCC/n_BCC;      //volume occupied by 1
     atom, m^3
9  a_FCC = 2*sqrt(2)*r_FCC;      //In FCC, A
10 a_FCCm = a_FCC*10^-10;       //converting a from A to
       m
11 V_FCC = a_FCCm^3;      //volume of unit cell, m^3
12 n_FCC = ((1/2)*6) + ((1/8)*8);       //number of
     atoms per unit cell
13 V1_FCC = V_FCC/n_FCC;      //volume occupied by 1
     atom, m^3
14 delta_V = (V1_BCC - V1_FCC)*100/V1_BCC;       //
     change in volume in %
15 printf("decrease of volume during conversion from
     BCC to FCC is %5.1f percent",delta_V);
```

Scilab code Exa 2.3 To calculate the volume of the unit cell and density of Zinc

```
1  //To calculate the volume of the unit cell and
     density of Zinc
2  a = 0.27;       //nearest neighbour distance, nm
3  c = 0.494;      //height of unit cell, nm
4  N = 6.023*10^26;     //avagadro number, k/mol
5  M = 65.37;      //atomic weight, kg
6  a_m = a*10^-9;     //nearest neighbour distance, m
7  c_m = c*10^-9;      //height of unit cell, m
8  V = 3*sqrt(3)*(a_m^2)*c_m/2;       //volume of unit
     cell, m^3
```

```
9 //if rho is density then mass = V*rho
10 //V*rho = 6*M/N
11 rho = (6*M)/(N*V);        //density, kg/m^3
12 printf("Volume of the unit cell in m^3 is");
13 disp(V);
14 printf("density of Zinc is %d kg/m^3",rho);
15
16 //answer for density given in the book is wrong
```

Scilab code Exa 2.4 To calculate the maximum radius of the sphere

```
1 //To calculate the maximum radius of the sphere
2 //a = 4*r/sqrt(2) and R = (a/2) - r
3 //R = (4*r/(2*sqrt(2))) - r = (2*r/sqrt(2)) - r
4 //R = r*(sqrt(2)-1) = 0.414*r
5 printf("maximum radius of the sphere that can fit
    into the void is 0.414*r");
```

Scilab code Exa 2.5 To calculate the density of diamond

```
1 //To calculate the density of diamond
2 a = 0.356;      //cube edge of diamond, nm
3 aw = 12.01;       //atomic weight of carbon in kg
4 N = 6.023*10^26;       //avagadro's number
5 a_m = a*10^-9;       //cube edge of diamond, m
6 n = 8/(a_m^3);       //number of atoms/m^3
7 M = aw/N;       //mass of 1 carbon atom, kg
8 rho =  M*n;
9 printf("number of atoms per m^3 is");
10 disp(n);
11 printf("density of diamong is %d kg/m^3",rho);
12
13 //answer for density given in the book is wrong
```

Scilab code Exa 2.6 To calculate the distance between two adjacent atoms

```
1 //To calculate the distance between two adjacent
    atoms
2 MW = 23+35.5;      //molecular weight of NaCl, gm
3 N = 6.023*10^23;      //avagadro number, mol-1
4 rho = 2.18;      //density of NaCl, gm/cm^3
5 M = MW/N;       //mass of NaCl molecule, gm
6 n = rho/M;      //number of molecules per unit volume
7 n = 2*n;       //since NaCl is diatomic, atoms/cm^3
8 //length of edge of unit cube is n*a
9 //volume V = n^3*a^3 = 1 cm^3
10 V = 1;     //volume of unit cube, cm^3
11 a = (V/n)^(1/3);       //distance between two
    adjacent atoms, cm
12 a = a*10^8;        //distance between two adjacent
    atoms,A
13 printf("distance between two adjacent atoms is %5.2f
    A",a);
```

Scilab code Exa 2.7 To calculate the density of Copper crystal

```
1 //To calculate the density of Copper crystal
2 AW = 63.5;     //atomic weight of Cu, gm/mol
3 N = 6.023*10^23;      //avagadro's number, mol-1
4 r = 1.278;      //atomic radius, A
5 n = 4;      //number of atoms in unit cell
6 r = r*10^-8;       //atomic radius, cm
7 M = AW/N;       //mass of each Copper atom, gm
8 a = 4*r/sqrt(2);       //lattice constant, cm
9 m = n*M;     //mass of unit cell, gm
```

```
10 rho = m/a^3;      //density of copper crystal, gm/cm
   ^3
11 printf("density of Copper crystal is %5.3f gm/cm^3",
   rho);
```

Scilab code Exa 2.8 To calculate the free volume per unit cell

```
1 //To calculate the free volume per unit cell
2 r = 0.1249;      //atomic radius, nm
3 a = 4*r/sqrt(3);       //lattice constant, nm
4 a_m = a*10^-9;         //lattice constant, m
5 V = a_m^3;         //volume of unit cell, m^3
6 PF = 0.68;        //packing factor for BCC
7 FV = 1 - PF;       //free volume
8 FV1 = FV*V;        //free volume per unit cell, m^3
9 printf("free volume per unit cell in m^3 is");
10 disp(FV1);
```

Scilab code Exa 2.9 To sketch the crystal plane structures

```
1 //To sketch the crystal plane structures
2 //sketching is not possible
```

Scilab code Exa 2.10 To sketch the crystal plane structures

```
1 //To sketch the crystal plane structures
2 //sketching is not possible
```

Scilab code Exa 2.11 To calculate the number of atoms per square millimetre

```
1 //To calculate the number of atoms per square
    millimetre
2 //in (100) plane the total number of atoms are n
3 n = (1/4)*4;
4 //A = a^2. number of atoms per mm^2 is n/a^2
5 printf("number of atoms in (100) plane are %d",n);
6 printf("number of atoms per mm^2 is 1/a^2");
7 //in (110) plane, area is sqrt(2)*a*a = sqrt(2)*a^2
8 printf("number of atoms in (100) plane is 1");
9 printf("unit area contains 1/(sqrt(2)*a^2) = 0.707/a
    ^2 atoms/mm^2");
10 //in (111) plane, area = (1/2)*base*height = (1/2)*a
    *sqrt(2)*a*sqrt(2)*cosd(30)
11 x = cosd(30);
12 //area = (1/2)*a*sqrt(2)*a*sqrt(2)*x = 0.866*a^2 =
    0.58/a^2
13 n1 = (1/360)*60*3;
14 //number of atoms per unit area is 0.5/(0.866*a^2) =
15 printf("total number of atoms in (111) plane is %5.1
    f",n1);
16 printf("number of atoms per unit area is 0.58/a^2
    atoms/mm^2");
```

Scilab code Exa 2.12 To calculate the interplanar spacing for the planes

```
1 //To calculate the interplanar spacing for the
    planes
2 h1 = 1;
3 k1 = 1;
4 l1 = 0;      //for (110) plane
5 h2 = 2;
6 k2 = 1;
```

```
7  l2 = 2;      //for (212) plane
8  r = 0.1278;       //atomic radius, nm
9  r = r*10^-9;      //atomic radius, m
10 x1 = sqrt(h1^2+k1^2+l1^2);
11 a = 4*r/x1;      //nearest neighbouring distance, m
12 a = a*10^9;      //nearest neighbouring distance, nm
13 d_110 = a/x1;    //interplanar spacing for (110), nm
14 x2 = sqrt(h2^2+k2^2+l2^2);
15 d_212 = a/x2;    //interplanar spacing for (212), nm
16 printf("interplanar spacing for (110) is %5.4f nm",
      d_110);
17 printf("interplanar spacing for (212) is %5.4f nm",
      d_212);
```

Scilab code Exa 2.13 To calculate the ratio of the seperation between successive lattice planes

```
1  //To calculate the ratio of the seperation between
      successive lattice planes
2  h1 = 1;
3  k1 = 0;
4  l1 = 0;     //for (100) plane
5  x1 = sqrt(h1^2+k1^2+l1^2);
6  h2 = 1;
7  k2 = 1;
8  l2 = 0;     //for (110) plane
9  x2 = sqrt(h2^2+k2^2+l2^2);
10 h3 = 1;
11 k3 = 1;
12 l3 = 1;     //for (111) plane
13 x3 = sqrt(h3^2+k3^2+l3^2);
14 //d = a/sqrt(h^2+k^2+l^2)
15 //d100:d110:d111 = a:a/sqrt(2):a/sqrt(3)
16 //d100:d110:d111 = 1:1/sqrt(2):1/sqrt(3) =
      1:0.71:0.58
```

```
17 printf("ratio of the seperation between successive
     lattice planes is 1:0.71:0.58");
```

Scilab code Exa 2.14 To calculate the miller indices of a plane

```
1 //To calculate the miller indices of a plane
2 //plane intercepts at a,b/2,3*c
3 //therefore intercepts are (1 1/2 3)
4 //reciprocal of the intercepts is (1/2 2 1/3)
5 //thus miller indices are (3 6 1)
6 printf("miller indices of the plane are (3 6 1)");
```

Scilab code Exa 2.15 To calculate the wavelength of X rays and maximum order of diffraction possible

```
1 //To calculate the wavelength of X-rays and maximum
     order of diffraction possible
2 d = 0.282;      //lattice spacing, nm
3 n = 1;      //first order
4 theta = 8+(35*0.0166666667);       //glancing angle
     in degrees
5 d = d*10^-9;       //lattice spacing, m
6 lamda = 2*d*sind(theta)/n;      //wavelength of X-
     rays, m
7 lamda_nm = lamda*10^9;      //wavelength of X-rays,
     nm
8 theeta = 90;      //maximum value possible in degrees
9 n1 = 2*d*sind(theeta)/lamda;       //maximum order
     of diffraction possible
10 printf("wavelength of X-rays is %5.4f nm",lamda_nm);
11 printf("maximum order of diffraction possible is %d"
     ,n1);
```

Scilab code Exa 2.16 To calculate the highest order for which Braggs reflection can be seen

```
1 //To calculate the highest order for which Bragg's
    reflection can be seen
2 lamda = 1.5;       //wavelength, A.U
3 d = 1.6;       //interplanar spacing, A.U
4 theta = 90;      //maximum glancing angle possible,
    degrees
5 n = 2*d*sind(theta)/lamda;       //maximum possible
    diffraction order
6 printf("maximum possible diffraction order is %d",n)
    ;
```

Scilab code Exa 2.17 To calculate the interatomic spacing

```
1 //To calculate the interatomic spacing
2 h = 1;
3 k = 1;
4 l = 1;       //for (111) plane
5 theta = 30;     //glancing angle, degrees
6 n = 1;      //first order
7 x = sqrt(h^2+k^2+l^2);
8 lamda = 1.5418;      //wavelength of X-rays, A
9 lamda = lamda*10^-10;      //wavelength of X-rays, m
10 d = lamda/(2*sind(theta));
11 a = d*x;       //interatomic spacing, m
12 a = a*10^10;       //interatomic spacing, A
13 printf("the interatomic spacing is %5.3f A",a);
```

Scilab code Exa 2.18 To calculate the glancing angle

```
1 //To calculate the glancing angle
2 d100 = 0.28;      //lattice constant, nm
3 h = 1;
4 k = 1;
5 l = 0;        //for (110) plane
6 n = 2;        //second order
7 lamda = 0.071;      //wavelength of X-rays, nm
8 lamda_m = lamda*10^-9;      //wavelength of X-rays, m
9 d110 = d100/sqrt(h^2+k^2+l^2);      //interatomic
     spacing. nm
10 d110 = d110*10^-9;       //interatomic spacing. m
11 theta = asind(n*lamda_m/(2*d110));
12 printf("the glancing angle is %d degrees",theta);
```

Scilab code Exa 2.19 To calculate the distance between 110 planes

```
1 //To calculate the distance between (110) planes
2 a = 0.38;       //lattice constant, nm
3 h = 1;
4 k = 1;
5 l = 0;        //for (110) plane
6 d = a/sqrt(h^2+k^2+l^2);
7 printf("the distance between (110) planes is %5.2f
     nm",d);
```

Scilab code Exa 2.20 To compare the density of lattice points

```
1 //To compare the density of lattice points
2 //area of (110) plane is a*sqrt(2)*a = sqrt(2)*a^2
3 n = (1/4)*4;      //number of atoms
4 theta = 30;      //glancing angle
```

```
x = cosd(theta);
//area of (111) plane is (a/sqrt(2))*x*a*sqrt(2)
//hence area is (sqrt(3)/2)*a^2
n1 = 3*(1/6);      //number of atoms
printf("area of (110) plane contains %d atom",n);
printf("density of lattice points is 1/(sqrt(2)*a^2)
   ");
printf("area of (111) plane contains %5.1f atom",n1)
   ;
//density of lattice points is (1/2)/(sqrt(3)*a^2/2)
printf("density of lattice points is 1/(sqrt(3)*a^2)
   ");
//density of lattice points (111) plane : (110)
   plane is 1/(sqrt(3)*a^2) : 1/(sqrt(2)*a^2) = sqrt
   (2):sqrt(3)
printf("density of lattice points (111) plane :
   (110) plane is sqrt(2):sqrt(3)");
```

Scilab code Exa 2.21 To calculate the glancing angle

```

//To calculate the glancing angle
n = 2;      //second order
h = 1;
k = 1;
l = 0;    //plane (110)
lamda = 0.065;      //wavelength of X-rays, nm
lamda_m = lamda*10^-9;     //wavelength of X-rays, m
a = 0.26;       //axial length, nm
a_m = a*10^-9;     //axial length, m
x = sqrt(h^2+k^2+l^2);
theta = asind(n*lamda_m*x/(2*a_m));        //glancing
    angle, degrees
deg = int(theta);        //glancing angle, degrees
t = 60*(theta-deg);
```

```
15 mint = int(t);          //glancing angle, minutes
16 printf("the glancing angle is %d degrees %d minutes"
      ,deg,mint);
```

Scilab code Exa 2.22 To calculate the cube edge

```
1  //To calculate the cube edge
2  n = 1;     //first order
3  h = 1;
4  k = 1;
5  l = 1;      //for (111) plane
6  lamda = 1.54;     //wavelength, A.U
7  lamda_m = lamda*10^-10;       //wavelength, m
8  theta = 19.2;       //Bragg's angle, degrees
9  d = n*lamda_m/(2*sind(theta));      //interplanar
      spacing, m
10 a = d*sqrt(h^2+k^2+l^2);       //cube edge, m
11 a_AU = a*10^10;       //cube edge, A.U
12 printf("the cube edge is %5.3f A.U",a_AU);
```

Scilab code Exa 2.23 To calculate the cube edge

```
1  //To calculate the cube edge
2  n = 1;     //first order
3  h = 1;
4  k = 1;
5  l = 1;      //for (111) plane
6  lamda = 1.54;     //wavelength, A.U
7  lamda_m = lamda*10^-10;       //wavelength, m
8  theta = 19.2;       //Bragg's angle, degrees
9  d = n*lamda_m/(2*sind(theta));      //interplanar
      spacing, m
10 a = d*sqrt(h^2+k^2+l^2);       //cube edge, m
```

```
11 printf("the cube edge in m is");
12 disp(a);
```

Chapter 3

Principles of Quantum Mechanics

Scilab code Exa 3.1 To calculate the de Broglie wavelength

```
1 //To calculate the de Broglie wavelength
2 c = 3*10^8;      //velocity of light, m/sec
3 v = (1/10)*c;       //velocity of proton, m/sec
4 m = 1.67*10^-27;      //mass of proton, kg
5 h = 6.626*10^-34;     //planck's constant
6 lamda = h/(m*v);      //de Broglie wavelength, m
7 printf("de Broglie wavelength of proton in m is");
8 disp(lamda);
```

Scilab code Exa 3.2 To calculate the de Broglie wavelength

```
1 //To calculate the de Broglie wavelength
2 V = 400;        //potential, V
3 lamda = 12.26/sqrt(V);       //de Broglie wavelength,
    armstrong
4 printf("de Broglie wavelength of electron is %5.3f
   armstrong",lamda);
```

Scilab code Exa 3.3 To calculate the de Broglie wavelength

```
1 //To calculate the de Broglie wavelength
2 m = 1.674*10^-27;      //mass of neutron, kg
3 h = 6.626*10^-34;     //planck's constant
4 e = 1.6*10^-19;
5 KE = 0.025;      //kinetic energy, eV
6 E = KE*e;      //kinetic energy, J
7 lamda = h/sqrt(2*m*E);       //de Broglie wavelength,
    m
8 lamda_nm = lamda*10^9;       //de Broglie wavelength,
    nm
9 printf("de Broglie wavelength is %5.3f nm",lamda_nm)
   ;
```

Scilab code Exa 3.4 To calculate the wavelength

```
1 //To calculate the wavelength
2 V = 1600;       //potential, V
3 lamda = 12.26/sqrt(V);       //de Broglie wavelength,
    armstrong
4 printf("de Broglie wavelength of electron is %5.4f
   armstrong",lamda);
```

Scilab code Exa 3.5 To calculate the uncertainity in momentum

```
1 //To calculate the uncertainity in momentum
2 delta_x = 0.2;       //electron distance, armstrong
```

```
3 delta_x = delta_x*10^-10;       //electron distance,
    m
4 h = 6.626*10^-34;     //planck's constant
5 delta_p = h/(2*%pi*delta_x);       //uncertainity in
    momentum, kg.m/s
6 printf("uncertainity in momentum in kg m/s is");
7 disp(delta_p);
```

Scilab code Exa 3.6 To calculate the lowest energy of an electron

```
1 //To calculate the lowest energy of an electron
2 n1 = 1;
3 n2 = 1;
4 n3 = 1;       //for lowest energy
5 e = 1.6*10^-19;
6 h = 6.62*10^-34;       //planck's constant
7 m = 9.1*10^-31;       //mass of electron, kg
8 L = 0.1;      //side of box, nm
9 L = L*10^-9;     //side of box, m
10 E1 = h^2*(n1^2+n2^2+n3^2)/(8*m*L^2);       //lowest
    energy, J
11 E1 = E1/e;       //lowest energy, eV
12 printf("lowest energy of electron is %5.1f eV",E1);
```

Scilab code Exa 3.7 To calculate the energy of electron

```
1 //To calculate the energy of electron
2 n1 = 1;
3 n2 = 1;
4 n3 = 2;       //for level next to the lowest
5 e = 1.6*10^-19;
6 h = 6.62*10^-34;       //planck's constant
7 m = 9.1*10^-31;       //mass of electron, kg
```

```
L = 0.1;      //side of box, nm
L = L*10^-9;     //side of box, m
E1 = h^2*(n1^2+n2^2+n3^2)/(8*m*L^2);       //lowest
   energy, J
E1 = E1/e;       //lowest energy, eV
printf("energy of electron is %5.2f eV",E1);

//answer given in the book is wrong in the 2nd
   decimal
```

Scilab code Exa 3.8 To calculate the wavelength

```
//To calculate the wavelength
m = 9.1*10^-31;      //mass of electron, kg
h = 6.626*10^-34;     //planck's constant
e = 1.6*10^-19;
E = 2000;      //energy, eV
E = E*e;      //energy, J
lamda = h/sqrt(2*m*E);       //wavelength, m
lamda_nm = lamda*10^9;       //velength, nm
printf("wavelength is %5.4f nm",lamda_nm);
```

Scilab code Exa 3.9 To calculate the minimum energy

```
//To calculate the minimum energy
n = 1;      //for minimum energy
h = 6.626*10^-34;       //planck's constant, J sec
m = 9.1*10^-31;       //mass of electron, kg
L = 4*10^-10;      //side of box, m
E1 = h^2*n^2/(8*m*L^2);       //lowest energy, J
printf("energy of electron in J is");
disp(E1);

```

```
10 //answer given in the book is wrong
```

Scilab code Exa 3.10 To calculate the energy values

```
1 //To calculate the energy values
2 n1 = 1;      //for ground state
3 n2 = 2;      //for 1st excited state
4 n3 = 3;      //for 2nd excited state
5 h = 6.626*10^-34;       //planck's constant, J sec
6 m = 9.1*10^-31;       //mass of electron, kg
7 L = 1*10^-10;      //width, m
8 E1 = h^2*n1^2/(8*m*L^2);       //energy in ground
    state, J
9 E2 = n2^2*E1;       //energy in 1st excited state, J
10 E3 = n3^2*E1;       //energy in 2nd excited state, J
11 printf("energy in ground state in J is");
12 disp(E1);
13 printf("energy in 1st excited state in J is");
14 disp(E2);
15 printf("energy in 2nd excited state in J is");
16 disp(E3);
```

Scilab code Exa 3.11 To calculate the velocity and kinetic energy of electron

```
1 //To calculate the velocity and kinetic energy of
    electron
2 h = 6.626*10^-34;       //planck's constant, J sec
3 m = 9.1*10^-31;       //mass of electron, kg
4 e = 1.6*10^-19;
5 lamda = 1.66*10^-10;       //wavelength, m
6 v = h/(m*lamda);       //velocity of electron, m/sec
7 v_km = v*10^-3;       //velocity of electron, km/sec
```

```
8 printf("velocity of electron is %d km/sec",v_km);
9 KE = (1/2)*m*v^2;     //kinetic energy, J
10 KE_eV = KE/e;         //kinetic energy, eV
11 printf("kinetic energy of electron is %5.2f eV",
   KE_eV);
```

Scilab code Exa 3.12 To calculate the wavelength

```
1 //To calculate the wavelength
2 V = 15;       //potential, kV
3 V = V*10^3;      //potential, V
4 lamda = 12.26/sqrt(V);       //de Broglie wavelength,
    armstrong
5 printf("wavelength of electron waves is %5.1f
   armstrong",lamda);
```

Scilab code Exa 3.13 To calculate the spacing of crystal

```
1 //To calculate the spacing of crystal
2 V = 344;       //potential, V
3 lamda = 12.26/sqrt(V);       //de Broglie wavelength,
    armstrong
4 lamda_m = lamda*10^-10;      //de Broglie wavelength,
    m
5 n = 1;      //for 1st reflection maximum
6 theta = 60;      //glancing angle, degrees
7 d = n*lamda_m/(2*sind(theta));       //interatomic
   spacing, m
8 d = d*10^10;        //interatomic spacing, armstrong
9 printf("interatomic spacing of crystal is %5.4f
   armstrong",d);
```

Chapter 4

Electron Theory of Metals

Scilab code Exa 4.1 To calculate the density and mobility of electrons

```
1 //To calculate the density and mobility of electrons
2 rho_s = 10.5*10^3;     //density of silver, kg/m^3
3 NA = 6.02*10^26;      //avagadro number, /k-mol
4 MA = 107.9;      //atomic weight of silver
5 n = rho_s*NA/MA;       //molar volume of silver
6 printf("density of electrons in silver is");
7 disp(n);
8 sigma = 6.8*10^7;      //conductivity of silver, ohm
    -1 m-1
9 e = 1.6*10^-19;
10 mew = sigma/(n*e);      //mobility of electrons, m^2/
    Vs
11 printf("mobility of electrons is %5.5f m^2/Vs",mew);
```

Scilab code Exa 4.2 To calculate the mobility of electrons

```
1 //To calculate the mobility of electrons
2 d = 8.92*10^3;       //density, kg/m^3
```

```
e = 1.6*10^-19;
m = 9.1*10^-31;       //mass of electron, kg
N = 6.02*10^26;       //avagadro's number per k-mol
AW = 63.5;      //atomic weight
rho = 1.73*10^-8;      //resistivity of copper, ohm-m
n = d*N/AW;        //number of cu atoms/m^3
mew = 1/(rho*n*e);       //mobility of electrons, m/
   Vs
printf("mobility of electrons is %f m/Vs",mew);
tow = m/(n*e^2*rho);     //relaxation time, s
printf("relaxation time in sec is");
disp(tow);
```

Scilab code Exa 4.3 To calculate the relaxation time

```
//To calculate the relaxation time
rho = 1.54*10^-8;       //resistivity, ohm-m
n = 5.8*10^28;       //conduction electrons per m^3
m = 9.108*10^-31;     //mass of electron, kg
e = 1.602*10^-19;
tow = m/(n*e^2*rho);     //relaxation time, sec
printf("relaxation time of conduction electrons in
   sec is");
disp(tow);
```

Scilab code Exa 4.4 To calculate the free electron concentration mobility and drift velocity

```
//To calculate the free electron concentration,
   mobility and drift velocity
R = 0.06;       //resistance, ohm
D = 5;       //length of Al wire, m
e = 1.602*10^-19;
```

```
rho = 2.7*10^-8;      //resistivity of Al, ohm-m
MA = 26.98;      //atomic weight
NA = 6.025*10^26;      //avagadro number, k/mol
rho_s = 2.7*10^3;      //density, kg/m^3
n = 3*rho_s*NA/MA;      //free electron concentration, electrons/m^3
printf("free electron concentration in electrons/m^3 is");
disp(n);
mew = 1/(n*e*rho);      //mobility, m/Vs
printf("mobility is %f m/Vs",mew);
I = 15;      //current, A
E = I*R/D;      //electric field, V/m
vd = mew*E;      //drift velocity, m/s
printf("drift velocity is %f m/s",vd);
```

Scilab code Exa 4.5 To calculate the lowest energy of an electron

```
//To calculate the lowest energy of an electron
n1 = 1;
n2 = 1;
n3 = 1;      //for lowest energy
h = 6.62*10^-34;      //planck's constant, Js
e = 1.6*10^-19;
m = 9.1*10^-31;      //mass of electron, kg
L = 0.1;      //side of box, nm
L = L*10^-9;      //side of box, m
E1 = h^2*(n1^2+n2^2+n3^2)/(8*m*L^2);      //lowest energy, J
E1 = E1/e;      //lowest energy, eV
printf("lowest energy of electron is %5.1f eV",E1);
```

Scilab code Exa 4.6 To evaluate the fermi function

```
1 //To evaluate the fermi function
2 //Fermi function F(E) = 1/(1+exp((E-Ef)/(kT)))
3 //given E-Ef = kT. therefore F(E) = 1/(1+exp(1))
4 F_E = 1/(1+exp(1));
5 printf("fermi function is %5.3f",F_E);
```

Scilab code Exa 4.7 To calculate the temperature

```
1
2 //To calculate the temperature
3 F_E = 10;       //probability in percent
4 k = 1.38*10^-23;
5 e = 1.6*10^-19;     //conversion factor
6 EF = 5.5;       //fermi energy, eV
7 E = EF+(EF/100);      //energy, eV
8 X = E-EF;      //E-EF, eV
9 X = X*e;       //E-EF, J
10 T = X/(k*log(F_E-1));       //temperature, K
11 printf("temperature is %5.1f K",T);
```

Scilab code Exa 4.8 To calculate the temperature

```
1 //To calculate the temperature
2 F_E = 0.01;       //probability in percent
3 k = 1.38*10^-23;
4 e = 1.6*10^-19;
5 //let E-EF be X
6 X = 0.5;       //E-EF, eV
7 kT = X/(2.303*log10((1-F_E)*100));     //value of kT,
    eV
8 T = kT*e/k;         //temperature, K
9 printf("temperature is %d K",T);
10
```

```
11 //answer given in the book is wrong
```

Chapter 6

Dielectric Properties

Scilab code Exa 6.1 To calculate the energy stored in the condensor and polarizing the dielectric

```
1 //To calculate the energy stored in the condensor
    and polarizing the dielectric
2 C = 2;      //capacitance, micro-farad
3 C = C*10^-6;       //capacitance, farad
4 V = 1000;     //voltage applied, V
5 epsilon_r = 100;      //permitivity
6 W = (C*V^2)/2;       //energy stored in capacitor, J
7 printf("energy stored in capacitor is %d J",W);
8 C0 = C/epsilon_r;    //capacitance removing the
    dielectric
9 W0 = C0*V^2/2;       //energy stored without
    dielectric, J
10 E = 1-W0;        //energy stored in dielectric, J
11 printf("energy stored in the dielectric is %5.2f J",
    E);
```

Scilab code Exa 6.2 To calculate the ratio between electronic and ionic polarizability

```
1 //To calculate the ratio between electronic and
    ionic polarizability
2 epsilon_r = 4.94;
3 n_2 = 2.69;      //square of index of refraction
4 alpha_i = 0;     //at optical frequencies
5 //(epsilon_r-1)/(epsilon_r+2) = N*(alpha_e+alpha_i)
    /(3*epsilon0)
6 X = (epsilon_r-1)/(epsilon_r+2);
7 //epsilon_r = n^2. therefore (n^2-1)/(n^2+2) = N*
    alpha_e/(3*epsilon0)
8 Y = (n_2-1)/(n_2+2);
9 //N*(alpha_e+alpha_i)/N*alpha_e = X/Y
10 //let alpha = alpha_i/alpha_e
11 alphai_e = (X/Y)-1;      //ratio between electronic
    ionic and electronic polarizability
12 printf("ratio between electronic ionic and
    electronic polarizability is %5.4f",alphai_e);
13 alphae_i = 1/alphai_e;       //ratio between
    electronic and ionic polarizability
14 printf("ratio between electronic and ionic
    polarizability is %5.3f",alphae_i);
```

Scilab code Exa 6.3 To calculate the difference in magnetic potential energy

```
1 //To calculate the difference in magnetic potential
    energy
2 epsilon_r = 2.56;
3 tan_delta = 0.7*10^-4;
4 f = 1;      //frequency, MHz
5 A = 8;      //area, cm^2
6 A = A*10^-4;       //area, m^2
7 d = 0.08;      //diameter, mm
8 d = d*10^-3;      //diameter, m
9 epsilon_rdash = epsilon_r*tan_delta;
```

```
epsilon0 = 8.85*10^-12;
omega = 2*%pi*10^6;
Rp = d/(omega*epsilon0*epsilon_rdash*A);       //
   parallel loss resistance, ohm
Rp = Rp*10^-6;       //parallel loss resistance, Mega
    ohm
Cp = A*epsilon0*epsilon_r/d;      //capacitance,
   farad
printf("parallel loss resistance is %5.3f ohm",Rp);
printf("capacitance in Farad is");
disp(Cp);
```

Scilab code Exa 6.4 To calculate the dielectric constant of material

```
//To calculate the dielectric constant of material
N = 3*10^28;      //density, atoms/m^3
alpha_e = 10^-40;     //electronic polarizability,
   Farad-m^2
epsilon0 = 8.854*10^-12;
epsilon_r = 1+(N*alpha_e/epsilon0);        //
   dielectric constant of material
printf("dielectric constant of material is %5.3f",
   epsilon_r);
```

Scilab code Exa 6.5 To calculate the electronic polarizability

```
//To calculate the electronic polarizability
epsilon0 = 8.854*10^-12;
epsilon_r = 1.0000684;       //dielectric constant
N = 2.7*10^25;      //density, atoms/m^3
alpha_e = epsilon0*(epsilon_r-1)/N;         //
   electronic polarizability, Fm^2
printf("electronic polarizability in Fm^2 is");
```

```
7 disp(alpha_e);
```

Scilab code Exa 6.6 To calculate the capacitance and charge on plates

```
1 //To calculate the capacitance and charge on plates
2 epsilon0 = 8.85*10^-12;
3 V = 100;      //potential, V
4 A = 100;      //area, cm^2
5 A = A*10^-4;      //area, m^2
6 d = 1;     //plate seperation, cm
7 d = d*10^-2;     //plate seperation, m
8 C = epsilon0*A/d;      //capacitance, farad
9 printf("capacitance of capacitor in F is");
10 disp(C);
11 Q = C*V;      //charge on plates
12 printf("charge on plates in coulomb is");
13 disp(Q);
```

Scilab code Exa 6.7 To calculate the electronic polarizability

```
1 //To calculate the electronic polarizability
2 N = 6.02*10^26;      //avagadro number
3 d = 2050;       //density, kg/m^3
4 AW = 32;        //atomic weight of sulphur
5 n = N*d/AW;     //number of atoms/m^3
6 epsilon_r = 3.75;     //relative dielectric constant
7 epsilon0 = 8.55*10^-12;
8 alpha_e = ((epsilon_r-1)/(epsilon_r+2))*3*epsilon0/n
    ;      //electronic polarizability, Fm^2
9 printf("electronic polarizability in Fm^2 is");
10 disp(alpha_e);
```

Scilab code Exa 6.8 To calculate the resultant voltage across capacitors

```
1 //To calculate the resultant voltage across
     capacitors
2 Q = 2*10^-10;          //charge, coulomb
3 d = 4;       //plate seperation, mm
4 d = d*10^-3;         //plate seperation, m
5 epsilon_r = 3.5;      //dielectric constant
6 A = 650;       //area, mm^2
7 A = A*10^-6;       //area, m^2
8 epsilon0 = 8.85*10^-12;
9 V = Q*d/(epsilon0*epsilon_r*A);         //voltage
     across capacitors, V
10 printf("resultant voltage across capacitors is %5.2f
     V",V);
```

Scilab code Exa 6.9 To calculate the dielectric displacement

```
1 //To calculate the dielectric displacement
2 V = 10;        //potential, V
3 d = 2*10^-3;         //plate seperation, m
4 E = V/d;       //electric field, V/m
5 epsilon_r = 6;
6 epsilon0 = 8.85*10^-12;
7 D = epsilon0*epsilon_r*E;         //dielectric
     displacement, C/m^2
8 printf("dielectric displacement in C/m^2 is");
9 disp(D);
```

Chapter 7

Magnetic Properties

Scilab code Exa 7.1 To calculate the relative permeability

```
1 //To calculate the relative permeability
2 M = 1.4;       //field , T
3 H = 6.5*10^-4;       //magnetic field , T
4 chi = M/H;       //susceptibility
5 mew_r = 1+chi;        //relative permeability
6 printf("relative permeability of iron is %d",mew_r);
7
8 //answer given in the book is wrong
```

Scilab code Exa 7.2 To calculate the relative permeability of material

```
1 //To calculate the relative permeability of material
2 M = 3300;      //magnetisation , amp/m
3 H = 220;       //field strength , amp/m
4 mew_r = (M/H)+1;       //relative permeability
5 printf("relative permeability of material is %d",
     mew_r);
```

Scilab code Exa 7.3 To calculate the magnetisation and flux density

```
1 //To calculate the magnetisation and flux density
2 H = 10^6;       //magnetic field intensity, amp/m
3 chi = 1.5*10^-3;      //susceptibility
4 mew0 = 4*%pi*10^-7;
5 M = chi*H;       //magnetisation, A/m
6 printf("magnetisation of material is %d A/m",M);
7 B = mew0*(M+H);       //flux density, T
8 printf("flux density is %5.4f T",B);
9
10 //answer given in the book is wrong
```

Scilab code Exa 7.4 To calculate the magnetisation and flux density

```
1 //To calculate the magnetisation and flux density
2 H = 10^4;       //magnetic field intensity, amp/m
3 chi = 3.7*10^-3;      //susceptibility
4 mew0 = 4*%pi*10^-7;
5 M = chi*H;       //magnetisation, A/m
6 printf("magnetisation of material is %d A/m",M);
7 B = mew0*(M+H);       //flux density, Wb/m^2
8 printf("flux density is %5.4f Wb/m^2",B);
```

Scilab code Exa 7.5 To calculate the magnetic moment

```
1 //To calculate the magnetic moment
2 I = 500;      //current, mA
3 I = I*10^-3;      //current, A
```

```
4 d = 10;      //diameter, cm
5 r = d/2;      //radius, cm
6 r = r*10^-2;     //radius, m
7 A = 2*%pi*r^2;    //area, m^2
8 mew_m = I*A;      //magnetic moment, Am^2
9 printf("magnetic moment associated with the loop is
    %f Am^2",mew_m);
```

Scilab code Exa 7.6 To calculate the change in magnetic moment

```
1 //To calculate the change in magnetic moment
2 r = 5.29*10^-11;     //radius of orbit, m
3 B = 2;      //field applied, T
4 e = 1.602*10^-19;      //charge of electron, coulomb
5 m = 9.108*10^-31;     //mass of electron, kg
6 mew_ind = e^2*r^2*B/(4*m);       //change in magnetic
    moment
7 printf("change in magnetic moment in Am^2 is");
8 disp(mew_ind);
```

Scilab code Exa 7.7 To calculate the susceptibility

```
1 //To calculate the susceptibility
2 chi_1 = 2.8*10^-4;      //susceptibility
3 T1 = 350;       //temperature, K
4 T2 = 300;       //temperature, K
5 //chi = C/T where C is curie constant
6 chi_2 = chi_1*T1/T2;      //susceptibility at 300 K
7 printf("susceptibility at 300 K is %f",chi_2);
```

Scilab code Exa 7.8 To calculate the magnetic moment

```
1 //To calculate the magnetic moment
2 d = 8906;      //density, kg/m^3
3 n = 6.025*10^26;      //avagadro number
4 AW = 58.7;      //atomic weight
5 N = d*n/AW;      //number of atoms/m^3
6 Bs = 0.65;      //magnetic induction, Wb/m^2
7 mew0 = 4*%pi*10^-7;
8 mew_m = Bs/(N*mew0);      //magnetic moment, Am^2
9 mewB = 9.27*10^-24;
10 mew_m = mew_m/mewB;      //magnetic moment, mewB
11 printf("the magnetic moment of Ni is %5.2f mewB",
   mew_m);
```

Scilab code Exa 7.9 To calculate the temperature

```
1 //To calculate the temperature
2 H = 2;      //magnetic field, Wb/m^2
3 mew = 9.4*10^-24;
4 k = 1.38*10^-23;
5 //np = C*n0*exp(mew*H/(k*T))
6 //na = C*n0*exp(-mew*H/(k*T))
7 // np/na = exp(mew*H/(k*T))/exp(-mew*H/(k*T)) = exp
   (2*mew*H/(k*T))
8 //given np/na = 2. therefore exp(2*mew*H/(k*T)) = 2
9 T = 2*mew*H/(k*log(2));      //temperature, K
10 printf("temperature is %5.1f K",T);
```

Scilab code Exa 7.10 To calculate the magnetic moment and saturation magnetisation

```
1 //To calculate the magnetic moment and saturation
    magnetisation
2 AW = 157.26;       //atomic weight
3 d = 7.8*10^3;      //density, kg/m^3
4 A = 6.025*10^26;      //avagadro number
5 mew0 = 4*%pi*10^-7;
6 N = d*A/AW;       //number of atoms 1 kg contains
7 g = N/10^3;       //number of atoms 1 g contains
8 mew_B = 7.1;      //bohr magneton
9 mew_m = 9.27*10^-24;
10 mew_mg = g*mew_B*mew_m;      //magnetic moment per
    gram, Am^2
11 printf("magnetic moment per gram is %5.2f Am^2",
    mew_mg);
12 Bs = N*mew0*mew_m;       //saturation magnetisation,
    Wb/m^2
13 printf("saturation magnetisation is %5.4f Wb/m^2",Bs
    );
14
15 //answer for saturation magnetisation given in the
    book is wrong
```

Chapter 8

Semiconductors

Scilab code Exa 8.1 To calculate the resistivity

```
1 //To calculate the resistivity
2 ni = 2.37*10^19;       //intrinsic carrier density,
     per m^3
3 mew_e = 0.38;         //electron mobility, m^2/Vs
4 mew_h = 0.18;         //hole mobility, m^2/Vs
5 e = 1.6*10^-19;
6 sigma_i = ni*e*(mew_e+mew_h);       //conductivity,
     ohm-1 m-1
7 rho = 1/sigma_i;         //resistivity, ohm m
8 printf("resistivity is %5.3f ohm m",rho);
```

Scilab code Exa 8.2 To determine the position of Fermi level

```
1 //To determine the position of Fermi level
2 Eg = 1.12;          //band gap, eV
3 k = 1.38*10^-23;
4 T = 300;       //temperature, K
5 e = 1.6*10^-19;
```

```
6  m0 = 1;      //for simplicity assume value of m0 to
       be unity
7  mh = 0.28*m0;
8  me = 0.12*m0;
9  EF = (Eg/2)+(3*k*T*log(mh/me)/(4*e));       //
       position of Fermi level, eV
10 printf("position of Fermi level is %5.3f eV from the
       top of valence band",EF);
```

Scilab code Exa 8.3 To calculate the concentration of intrinsic charge carriers

```
1  //To calculate the concentration of intrinsic charge
       carriers
2  T = 300;       //temperature, K
3  pi = 22/7;     //value of pi
4  e = 1.6*10^-19;
5  m = 9.109*10^-31;      //mass of electron, kg
6  k = 1.38*10^-23;       //boltzmann's constant
7  h = 6.626*10^-34;      //planck's constant
8  Eg = 0.7;       //band gap, eV
9  Eg = Eg*e;      //band gap, J
10 A = (2*pi*m*k*T/h^2)^(3/2);
11 B = exp(-Eg/(2*k*T));
12 ni = 2*A*B;      //concentration of intrinsic charge
       carriers per m^3
13 printf("concentration of intrinsic charge carriers
       per m^3 is");
14 disp(ni);
```

Scilab code Exa 8.4 To calculate the resistivity

```
1  //To calculate the resistivity
```

```
ni = 2.4*10^19;       //intrinsic carrier density,
   per m^3
mew_e = 0.39;        //electron mobility, m^2/Vs
mew_h = 0.19;        //hole mobility, m^2/Vs
e = 1.6*10^-19;
sigma_i = ni*e*(mew_e+mew_h);       //conductivity,
   ohm-1 m-1
rho = 1/sigma_i;       //resistivity, ohm m
printf("resistivity is %5.3f ohm m",rho);
```

Scilab code Exa 8.5 To calculate the resistance

```
//To calculate the resistance
ni = 2.5*10^19;       //intrinsic carrier density,
   per m^3
mew_e = 0.39;        //electron mobility, m^2/Vs
mew_h = 0.19;        //hole mobility, m^2/Vs
e = 1.6*10^-19;
sigma_i = ni*e*(mew_e+mew_h);       //conductivity,
   ohm-1 m-1
w = 1;      //width, mm
w = w*10^-3;        //width, m
t = 1;      //thickness, mm
t = t*10^-3;      //thickness, m
A = w*t;        //area, m^2
l = 1;      //length, cm
l = l*10^-2;      //length, m
R = l/(sigma_i*A);       //resistivity, ohm m
printf("resistance of intrinsic Ge rod is %5.1f ohm"
   ,R);
```

Scilab code Exa 8.6 To calculate the conductivity

```
1 //To calculate the conductivity
2 m = 9.109*10^-31;     //mass of electron, kg
3 k = 1.38*10^-23;      //boltzmann constant
4 pi = 22/7;          //value of pi
5 h = 6.626*10^-34;       //planck's constant
6 C = 2*(2*pi*m*k/h^2)^(3/2);
7 T = 300;      //temperature, K
8 e = 1.6*10^-19;
9 Eg = 1.1;     //energy gap, eV
10 mew_e = 0.48;       //electron mobility, m^2/Vs
11 mew_h = 0.013;      //hole mobility, m^2/Vs
12 ni = C*T^(3/2)*exp(-Eg*e/(2*k*T));       //intrinsic
    carrier density per m^3
13 sigma_i = ni*e*(mew_e+mew_h);       //conductivity,
    ohm-1 m-1
14 printf("conductivity is %f ohm-1 m-1",sigma_i);
15
16 //answer given in the book is wrong
```

Scilab code Exa 8.7 To calculate the intrinsic carrier density and conductivity

```
1 //To calculate the intrinsic carrier density and
    conductivity
2 m = 9.109*10^-31;     //mass of electron, kg
3 k = 1.38*10^-23;      //boltzmann constant
4 pi = 22/7;          //value of pi
5 h = 6.626*10^-34;       //planck's constant
6 T = 300;      //temperature, K
7 e = 1.6*10^-19;
8 Eg = 0.7;     //energy gap, eV
9 ni = 2*(2*pi*m*k*T/h^2)^(3/2)*exp(-Eg*e/(2*k*T));
       //intrinsic carrier density per m^3
10 printf("intrinsic carrier density per m^3 is");
11 disp(ni);
```

```
12 mew_e = 0.4;       //electron mobility, m^2/Vs
13 mew_h = 0.2;      //hole mobility, m^2/Vs
14 sigma = ni*e*(mew_e+mew_h);       //conductivity, ohm
   -1 m-1
15 printf("conductivity is %5.2f ohm-1 m-1",sigma);
16
17 //answer given in the book is wrong
```

Scilab code Exa 8.8 To calculate the forbidden energy gap

```
1  //To calculate the forbidden energy gap
2  rho = 2.12;       //resistivity, ohm m
3  sigma = 1/rho;     //conductivity, ohm-1 m-1
4  e = 1.6*10^-19;
5  pi = 22/7;         //value of pi
6  m = 9.109*10^-31;     //mass of electron, kg
7  k = 1.38*10^-23;      //boltzmann constant
8  h = 6.626*10^-34;       //planck's constant
9  mew_e = 0.36;       //electron mobility, m^2/Vs
10 mew_h = 0.17;      //hole mobility, m^2/Vs
11 ni = sigma/(e*(mew_e+mew_h));       //intrinsic
   carrier density per m^3
12 C = 2*(2*pi*m*k/h^2)^(3/2);
13 T = 300;      //temperature, K
14 //let exp(Eg/(2*k*T)) be a
15 a = (C*T^(3/2))/ni;
16 //Eg/(2*k*T) = log(a) and Eg = 2*k*T*log(a)
17 Eg = 2*k*T*log(a)/e;       //forbidden energy gap, eV
18 printf("forbidden energy gap is %5.3f eV",Eg);
19
20 //answer given in the book is wrong
```

Scilab code Exa 8.9 To calculate the energy band gap

```
//To calculate the energy band gap
rho_2 = 4.5;        //resistivity at 20C
rho_1 = 2;          //resistivity at 32C
T1 = 20;        //temperature, C
T1 = T1+273;       //temp, K
T2 = 32;     //temp, C
T2 = T2+273;       //temp, K
k = 8.616*10^-5;
dy = log10(rho_2)-log10(rho_1);
dx = (1/T1)-(1/T2);
Eg = 2*k*dy/dx;         //energy band gap, eV
printf("energy band gap is %5.3f eV",Eg);

//answer given in the book is wrong
```

Scilab code Exa 8.10 To calculate the temperature

```
//To calculate the temperature
Eg = 1;       //band gap, eV
e = 1.602*10^-19;
Eg = Eg*e;       //band gap, J
me = 1;      //for simplicity assume me to be unity
mh = 4*me;       //effective mass of holes is 4 times
   of electrons
E_Ef = 10/100;       //fermi level shift, eV
E_Ef = E_Ef*e;       //fermi level shift, J
k = 1.38*10^-23;       //boltzmann constant
//E_Ef = 3*k*T*log(mh/me)/4
T = 4*E_Ef/(3*k*log(mh/me));        //temperature, K
printf("temperature is %d K",T);
```

Scilab code Exa 8.11 To calculate the electron concentration

```
//To calculate the electron concentration
Na = 5*10^23;       //atoms of boron
Nd = 3*10^23;       //arsenic atoms
p = 2*(Na-Nd)/2;       //hole concentration per m^3
ni = 2*10^16;      //intrinsic charge carriers per m
    ^3
n = ni^2/p;        //electron concentration per m^3
printf("electron concentration per m^3 is");
disp(n);
```

Scilab code Exa 8.12 To calculate the conductivity of intrinsic Silicon

```
//To calculate the conductivity of intrinsic Silicon
ni = 1.5*10^16;       //intrinsic charge carriers per
    m^3
e = 1.6*10^-19;
mew_e = 0.13;       //electron mobility, m^2/Vs
mew_h = 0.05;       //hole mobility, m^2/Vs
sigma = ni*e*(mew_e+mew_h);       //conductivity, ohm
    -1 m-1
printf("conductivity is %f ohm-1 m-1",sigma);
AW = 28.1;       //atomic weight of Si, kg
d = 2.33*10^3;       //density of Si, kg/m^3
N = 6.02*10^26;      //avagadro number
Nd = d*N/AW;       //impurity atoms per m^3
Nd = Nd/10^8;      //extent of 10^8 Si atoms
p = ni^2/Nd;       //hole concentration per m^3
sigma_ex = Nd*e*mew_e;        //conductivity, ohm-1 m
    -1
printf("conductivity if donor type impurity is added
    is %5.2f ohm-1 m-1",sigma_ex);
Na = Nd;
n = ni^2/Na;       //electron concentration per m^3
sigma_EX = Na*e*mew_h;        //conductivity, ohm-1 m
    -1
```

```
19 printf("conductivity if acceptor type impurity is
     added is %5.2f ohm-1 m-1",sigma_EX);
```

Scilab code Exa 8.13 To calculate the conductivity equilibrium hole concentration and position of Fermi level

```
1  //To calculate the conductivity, equilibrium hole
     concentration and position of Fermi level
2  ni = 1.5*10^16;      ////intrinsic charge carriers
     per m^3
3  e = 1.6*10^-19;
4  mew_e = 0.135;       //electron mobility, m^2/Vs
5  mew_h = 0.048;       //hole mobility, m^2/Vs
6  sigma = ni*e*(mew_e+mew_h);       //conductivity, ohm
     -1 m-1
7  printf("conductivity is %f ohm-1 m-1",sigma);
8  Nd = 10^23;       //phosphorus atoms per m^3
9  p = ni^2/Nd;      //hole concentration per m^3
10 printf("hole concentration per m^3 is");
11 disp(p);
12 sigma_ex = Nd*e*mew_e;       //conductivity, ohm-1 m
     -1
13 k = 1.38*10^-23;      //boltzmann constant
14 T = 300;      //temperature, K
15 //EF = (Eg/2)+(3*k*T*log(mew_e/mew_h)/4)
16 X = 3*k*T*log(mew_e/mew_h)/(4*e);
17 //EF = (Eg/2)+X
18 printf("EF = Eg/2 + %5.2f",X);
19 printf("Fermi level will be %5.2f eV above intrinsic
     level",X);
```

Scilab code Exa 8.14 To calculate the diffusion coefficient of electrons

```
1 //To calculate the diffusion coefficient of
    electrons
2 mew_e = 0.19;       //electron mobility, m^2/Vs
3 k = 1.38*10^-23;      //boltzmann constant
4 T = 300;      //temperature, K
5 e = 1.6*10^-19;
6 Dn = mew_e*k*T/e;      //diffusion coefficient, m^2/s
7 printf("diffusion coefficient of electrons is %f m
    ^2/s",Dn);
```

Scilab code Exa 8.15 To calculate the Hall voltage

```
1 //To calculate the Hall voltage
2 RH = 3.66*10^-4;      //Hall coefficient, m^3/coulomb
3 I = 10^-2;       //current, amp
4 B = 0.5;       //magnetic field, Wb/m^2
5 t = 1;       //thickness, mm
6 t = t*10^-3;      //thickness, m
7 VH = RH*I*B/t;       //Hall voltage, V
8 VH = VH*10^3;       //Hall voltage, mV
9 printf("Hall voltage developed is %5.2f mV",VH);
```

Scilab code Exa 8.16 To calculate the Hall coefficient

```
1 //To calculate the Hall coefficient
2 Vy = 37;       //voltage, micro-V
3 Vy = Vy*10^-6;      //voltage, V
4 t = 1;      //thickness, mm
5 t = t*10^-3;      //thickness, m
6 Bz = 0.5;       //flux density, Wb/m^2
7 Ix = 20;      //current, mA
8 Ix = Ix*10^-3;      //current, A
9 RH = Vy*t/(Ix*Bz);      //Hall coefficient, C-1 m^3
```

```
10 printf("Hall coefficient of semiconductor in C-1 m^3
       is");
11 disp(RH);
```

Scilab code Exa 8.17 To calculate the density and mobility of charge carriers

```
1 //To calculate the density and mobility of charge
     carriers
2 RH = -7.35*10^-5;        //Hall coefficient, m^3/C
3 e = 1.6*10^-19;
4 sigma = 200;        //conductivity, ohm-1 m-1
5 n = -1/(RH*e);         //density, m^3
6 printf("density of charge carriers in m^3 is");
7 disp(n);
8 mew = sigma/(n*e);       //mobility, m^2/Vs
9 printf("mobility of charge carriers is %5.4f m^2/Vs"
     ,mew);
```

Scilab code Exa 8.18 To calculate the magnitude of Hall voltage

```
1 //To calculate the magnitude of Hall voltage
2 I = 50;       //current, A
3 B = 1.5;      //magnetic field, T
4 e = 1.6*10^-19;
5 n = 8.4*10^28;        //free electron concentration,
     electron/m^3
6 t = 0.5;       //thickness, cm
7 t = t*10^-2;       //thickness, m
8 VH = I*B/(n*e*t);       //hall voltage, V
9 VH = VH*10^6;       //hall voltage, micro-V
10 printf("magnitude of Hall voltage is %5.3f micro-V",
     VH);
```

Scilab code Exa 8.19 To calculate the value of mew and n

```
1 //To calculate the value of mew and n
2 RH = 3.66*10^-4;       //Hall coefficient , m^3/C
3 e = 1.6*10^-19;
4 n = 1/(RH*e);
5 rho_n = 8.93*10^-3;      //resistivity , ohm m
6 printf("value of n per m^3 is");
7 disp(n);
8 mew_e = RH/rho_n;
9 printf("value of mew_e is %5.3f m^2/Vs",mew_e);
```

Chapter 9

Superconductivity

Scilab code Exa 9.1 To calculate the critical field

```
1 //To calculate the critical field
2 Tc = 3.7;       //critical temperature, K
3 Hc_0 = 0.0306;     //critical field, T
4 T = 2;      //temperature, K
5 Hc_2 = Hc_0*(1-(T/Tc)^2);      //critical field, T
6 printf("critical field at 2K is %5.5f T",Hc_2);
```

Scilab code Exa 9.2 To calculate the critical current

```
1 //To calculate the critical current
2 T = 4.2;      //temperature, K
3 d = 1;      //diameter, mm
4 d = d*10^-3;      //diameter, m
5 Tc = 7.18;     //critical temperature, K
6 H0 = 6.5*10^4;      //critical field, A/m
7 Hc = H0*(1-(T/Tc)^2);     //critical field at 2K, A/m
8 ic = %pi*d*Hc;      //critical current, A
9 printf("critical current for lead is %5.2f A",ic);
```

```
10
11 //answer given in the book is wrong
```

Scilab code Exa 9.3 To calculate the penetration depth

```
1 //To calculate the penetration depth
2 lamda_T = 750;     //penetration depth of mercury,
     Armstrong
3 T = 3.5;     //temperature, K
4 Tc = 4.12;     //critical temperarure, K
5 lamda_0 = lamda_T*((1-(T/Tc)^4))^(1/2);      //
     penetration depth, Armstrong
6 printf("penetration depth at 0K is %d armstrong",
     lamda_0);
```

Scilab code Exa 9.4 To calculate the critical temperature

```
1 //To calculate the critical temperature
2 T1 = 3;     //temperature, K
3 T2 = 7.1;     //temperature, K
4 lamda_T1 = 396;     //penetration depth, armstrong
5 lamda_T2 = 1730;     //penetration depth, armstrong
6 A = (((lamda_T2/lamda_T1)^2)*T2^4) - T1^4;
7 B = ((lamda_T2/lamda_T1)^2)-1;
8 Tc = (A/B)^(1/4);      //critical temperature, K
9 printf("critical temperature for lead is %5.3f K",Tc
     );
```

Chapter 10

Lasers

Scilab code Exa 10.1 To calculate the relative population

```
1 //To calculate the relative population
2 c = 3*10^8;     //speed of light, m/sec
3 h = 6.6*10^-34;     //planck's constant
4 e = 1.6*10^-19;
5 T = 300;     //temperature, K
6 K = 8.61*10^-5;
7 lamda = 6943;     //wavelength, armstrong
8 lamda = lamda*10^-10;      //wavelength, m
9 // let E2 - E1 be E
10 E = h*c/lamda;       //energy, J
11 E = E/e;        //energy, eV
12 //let population ratio N2/N1 be N
13 N = exp(-E/(K*T));
14 printf("relative population of 2 states is");
15 disp(N);
16
17 //answer given in the book is wrong
```

Scilab code Exa 10.2 To calculate the divergence

```
1 //To calculate the divergence
2 a2 = 6;      //spot diameter, mm
3 a2 = a2*10^-3;     //spot diameter, m
4 a1 = 4;      //spot diameter, mm
5 a1 = a1*10^-3;      //spot diameter, m
6 d2 = 2;     //distance from laser, m
7 d1 = 1;     //distance from laser, m
8 theta = (a2-a1)/(2*(d2-d1));      //divergence,
    radian
9 theta = theta*10^3;       //divergence, milli radian
10 printf("divergence is %d milli radian",theta);
```

Scilab code Exa 10.3 To calculate the spot size

```
1 //To calculate the spot size
2 n = 1;      //for air
3 lamda = 650;      //wavelength, nm
4 lamda = lamda*10^-9;     //wavelength, m
5 bs = 1;      //beam size, mm
6 bs = bs*10^-3;     //beam size, m
7 fl = 1;     //focal length of lens, mm
8 fl = fl*10^-3;      //focal length of lens, m
9 tan_theta = fl/(2*bs);      //value of tan_theta
10 theta = atand(tan_theta);
11 NA = n*sind(theta);
12 ss = 0.6*lamda/NA;        //spot size, m
13 ss = ss*10^6;        //spot size, micro metre
14 printf("spot size is %5.3f micro metre",ss);
15
16 //answer given in the book is wrong
```

Chapter 11

Fiber Optics and Holography

Scilab code Exa 11.1 To calculate the numerical aperture

```
1 //To calculate the numerical aperture
2 n1 = 1.55;     //refractive index of core
3 n2 = 1.50;     //refractive index of cladding
4 NA = sqrt(n1^2 - n2^2);
5 printf("numerical aperture is %5.3f",NA);
```

Scilab code Exa 11.2 To calculate the angle of acceptance

```
1 //To calculate the angle of acceptance
2 n1 = 1.563;     //refractive index of core
3 n2 = 1.498;     //refractive index of cladding
4 NA = sqrt(n1^2 - n2^2);       //numerical aperture
5 alpha_i = asind(NA);          //angle of acceptance,
    degrees
6 deg = int(alpha_i);         //angle in degrees
7 t = 60*(alpha_i-deg);
8 mint = int(t);         //angle in minutes
9 printf("the angle of acceptance is %d degrees %d
    minutes",deg,mint);
```

```
10
11 //answer given in the book is wrong
```

Scilab code Exa 11.3 To calculate the refractive index of the core

```
1 //To calculate the refractive index of the core
2 delta = 0.05;      //difference in refractive indices
      of core and cladding
3 NA = 0.39;     //numerical aperture
4 n1 = NA/sqrt(2*delta);      //refractive index of
     core
5 printf("refractive index of the core is %5.4f",n1);
```

Scilab code Exa 11.4 To calculate the fractional index change

```
1 //To calculate the fractional index change
2 n1 = 1.563;     //refractive index of core
3 n2 = 1.498;     //refractive index of cladding
4 delta = (n1-n2)/n1;       //fractional index change
5 printf("fractional index change is %5.4f",delta);
```

Scilab code Exa 11.5 To calculate the numerical aperture and acceptance angle

```
1 //To calculate the numerical aperture and acceptance
      angle
2 n1 = 1.48;     //refractive index of core
3 n2 = 1.45;     //refractive index of cladding
4 NA = sqrt(n1^2 - n2^2);      //numerical aperture
5 printf("numerical aperture is %5.4f",NA);
```

```
6  alpha_i = asind(NA);       //angle of acceptance,
       degrees
7  deg = int(alpha_i);        //angle in degrees
8  t = 60*(alpha_i-deg);
9  mint = int(t);        //angle in minutes
10 printf("the angle of acceptance is %d degrees %d
       minutes",deg,mint);
11
12 //answer given in the book is wrong
```