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

Created by
Ankush Kumar Bedyal
Solid State Physics
Physics
Shri Mata Vaishno Devi University
College Teacher
Dr. Vinay Kumar
Cross-Checked by
Lavitha Pereira

November 8, 2013

[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: Solid State Physics

Author: P. K. Palanisamy

Publisher: Scitech Publication (India) Pvt. Ltd., Chennei

Edition: 1

Year: 2004

ISBN: 8188429279

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 Stability of gaseous molecules

```
1 // Scilab Code Ex1.1 : Page-1.8 (2004)
2 clc;clear;
3 N = 6.022e+23;        //  Avogadro Number; /mol
4 E_A = 502;            // First ionization energy of A
    atom, kJ/mol
5 E_B = -335;          // Electron affinity for B atom,
    kJ
6 r = 3e-10;           // Velocity of the particle at the
     mean position, angstrom
7 E_o = 8.85e-12; //Permittivity of free space, C/N^2-
    m^2
8 e = 1.6e-19;      // Electronic charge, C
9 E_C = N*(-e^2)/(4*%pi*E_o*r)*1e-03;    // The
    coulombic electrostatic attaction energy, kJ/mol
10 E = E_A + E_B + E_C;     // Net change in energy per
    mol, kJ/mol
11 printf("\nNet change in energy per mol = %3d kJ/mol"
    , E);
12 disp("Since net change in energy is negative, the
    molecule A+B- is stable.");
13
```

```
14 // Result
15 // Net change in energy per mol = -295 kJ/mol
16 // Since net change in energy is negative, the
   molecule A+B- is stable.
```

Scilab code Exa 1.2 Energy and separation in KCl ion pair

```
1 // Scilab Code Ex1.2 : Page-1.8 (2004)
2 clc;clear;
3 A = 4.1;     // Ionization energy of K, eV
4 B = 3.6;     // Electron affinity of Cl, eV
5 C = A - B;     // Net energy to produce the ion pair,
    eV
6 E = C;      // Coulomb energy, eV
7 e = 1.6e-19;     // Electronic charge, C
8 E_o = 8.85e-12; //Permittivity of free space, C/N^2-
   m^2
9 R = e/(4*%pi*E_o*C);    // R is the separation
   between K and Cl , nm
10 printf("\nThe coulomb energy E = %3.1f eV", E);
11 printf("\nSeparation between K and Cl, R = %4.2f nm"
   , R/1e-09);
12
13 // Result
14 //  The coulomb energy E = 0.5 eV
15 // Separation between K and Cl, R = 2.88 nm
```

Scilab code Exa 1.3 Bond energy of NaCl molecule

```
1 // Scilab Code Ex1.3 : Bond energy of NaCl molecule
    Page-1.9 (2004)
2 clc;clear;
3 A = 5.14;     // Ionization energy of Na, eV
```

```
4  B = 3.65;     // Electron affinity of Cl, eV
5  r0 = 236e-12;  // Inter ionic equilibrium distance,
     m
6  e = 1.6e-19;     // Electronic charge, C
7  E_o = 8.85e-12; //Permittivity of free space, C/N^2-
     m^2
8  Ue = -e/(4*%pi*E_o*r0);     // Potential energy to
     produce the ion pair, eV
9  BE = -Ue-A+B
10 printf("\nThe potential energy Ue = %2.1f eV",Ue);
11 printf("\nThe Binding energy BE = %4.2f eV",BE);
12
13 // Result
14 //  The potential energy Ue = -6.1 eV
15 //  The Binding energy BE = 4.61 eV
```

Chapter 2

Crystal Structure

Scilab code Exa 2.1 Maximum radius of interstitial sphere in BCC

```
1 // Scilab Code Ex2.1 Page-2.21 (2004)
2 clc;clear;
3 r = 1;  // For simplicity assume radius of atom to
     be unity, unit
4 a = 4*r/sqrt(3);  // Lattice constant, unit
5 R = (a-2*r)/2;      // R be the radius of
     interstitial sphere that can fit into void, unit
6
7 printf ("\nRadius of interstitial sphere that can
     fit into void R = %5.3fr", R);
8
9 //  Result
10 //  Radius of interstitial sphere that can fit into
     void R = 0.155r
```

Scilab code Exa 2.2 Volume Phase change BCC to FCC

```
1 // Scilab Code Ex2.2 : Page-2.22 (2004)
```

```
clc;clear;
r1 = 1.258e-10; // Radius of atom for BCC, m
a = 4*r1/(3^(0.5)); // Lattice constant for BCC
    atom, m
V_bcc = a^3; // Volume of unit cell of BCC , in m^3
V_one = V_bcc/2; // Volume occupied by one atom in
    BCC, in m^3
r2 =    1.292e-10; // Radius of atom for FCC, m
b = 2*(sqrt(2))*r2    // Lattice constant for FCC
    atom
V_fcc = b^3; // Volume of unit cell of FCC , in m^3
V_two = V_bcc/4; // Volume occupied by one atom in
    FCC, in m^3
DV = (V_one-V_two)/V_one; // Change in volume ,
    percentage

printf("\nChange in volume DV =%3.1f percentage.",
    DV);

// Result
// Change in volume DV =0.5 percentage.
```

Scilab code Exa 2.3 Calculation of volume and density of Zinc

```
// Scilab Code Ex2.3 : Page-2.23 (2004)
clc;clear;
a = 0.27e-9; // Nearest neighbour distance, m
c = 0.494e-9;   // Height of unit cell, m
V = 3*(sqrt(3))*(a^2)*c/2; // Volume of unit cell
    of HCP , in metre cube
N = 6.023e+26; // Avodagro number , per k-mol
M = 65.37;  // Atomic weight of zinc,
rho = 6*M/(V*N);    // Density of zinc, kg per
    metre cube
printf("\nVolume  = %4.3e metre cube", V);
```

```
10 printf("\nDensity of zinc = %d kg per metre cube",
     rho);
11
12 // Result
13 //  Volume  = 9.356e-29 metre cube
14 //  Density of zinc = 6960 kg per metre cube
```

Scilab code Exa 2.4 Maximum radius of interstitial sphere in FCC

```
1 // Scilab Code Ex2.4 : Page-2.23 (2004)
2 clc;clear;
3 r = 1;  // For simplicity assume radius of atom to
     be unity, unit
4 a = 4*r/sqrt(2);  // Lattice constant, unit
5 R = (a/2)-r;      // R be the radius of interstitial
     sphere that can fit into void, unit
6
7 printf ("\nMaximum Radius of interstitial sphere
     that can fit into FCC = %5.3fr", R);
8
9 //  Result
10 //  Maximum Radius of interstitial sphere that can
     fit into FCC = 0.414r
```

Scilab code Exa 2.5 Density of diamond

```
1 // Scilab Code Ex2.5 : Page-2.24 (2004)
2 clc;clear;
3 a = 0.356e-9;   // Cube edge of diamond, m
4 n = 8/a^3;    // Number of atoms per unit volume,
     per metre cube
5 M = 12.01;  // Atomic weight of Carbon, g per mol
6 N = 6.023e+26;  //  Avagadro number, per kmol
```

```
7 m = M/N;     // Mass of one carbon atom, kg
8 rho = m*n;    // Density of diamond, kg per metre
    cube
9 printf("\nDensity of diamond = %4.0f kg per metre
    cube", round(rho));
10
11 // Result
12 // Density of diamond = 3536 kg per metre cube
```

Scilab code Exa 2.6 Distance between two adjacent atoms in NaCl

```
1 // Scilab Code Ex2.6 : Page-2.25 (2004)
2 clc;clear;
3 M = 58.5;  // Molecular weight of Carbon, g per mol
4 N = 6.023e+23;  //  Avagadro number, per mol
5 m = M/N;     // Mass of one NaCl molecule, g
6 n = 2.18/m;    // Number of NaCl molecules per unit
    volume, molecules per cm cube
7 n1 = 2*n;    // Since NaCl is a diatomic, number of
    atoms per unit volume is twice
8 a = 1/n1^(1/3); // Distance between two adjacent
    atoms in NaCl, angstrom
9
10 printf("\nDistance between two adjacent atoms in
    NaCl = %4.2f angstrom", a/1e-8);
11
12 // Result
13 //  Distance between two adjacent atoms in NaCl =
    2.81 angstrom
```

Scilab code Exa 2.7 Density of copper crystal

```
1 // Scilab Code Ex2.7: Page-2.25 (2004)
```

```
clc;clear;
M = 63.5;  // Atomic weight of Copper, g per mol
N = 6.023e+23;  //  Avagadro number, per mol
r = 1.278e-8;   // Atomic radius of copper, cm
m = M/N;    // Mass of one Cu atom, kg
a = (4*r)/sqrt(2); // Distance between two adjacent atom in Cu, angstrom
n1 = 4*m;   // Since number of atom per unit cell is 4
d = n1/a^3;   // Density of copper, kg per metre cube

printf("\nDensity of copper = %4.2f g per cm cube", d);

// Result
//  Density of copper = 8.93 g per cm cube
```

Chapter 3

Crystal Planes and Defects

Scilab code Exa 3.1 Number of atoms per square mm in SC

```
1 // Scilab Code Ex3.1 : Page-3.4 (2004)
2  // In a SC structure number of planes are having
      three arrangement (100),(110) and (111)
3  clc;clear;
4 a =  1;      // For simplicity lattice constant is
     taken to be unity
5 A_100 = a^2;     // Area of the plane (100), mm^2
6 N_100 = 1/A_100;     //   Number of atoms along (100)
      plane, atoms per square mm
7 A_110 = sqrt(2)*a^2;     // Area of the plane (110),
     mm^2
8 N_110 = 1/A_110;     // //   Number of atoms along
     (110) plane, atoms per square mm
9 A_111 = 1/2*a*sqrt(2)*sqrt(2)*a^2*cosd(30);     //
     Area of the plane (110), mm^2
10 A_111t = 0.5;    //   Total no of atoms in (111)
     plane
11 N_111 = A_111t/A_111;     // //   Number of atoms
     along (110) plane, atoms per square mm
12 printf("\nNumber of atoms along (100) plane= %d /a^2
      atoms per square mm", N_100);
```

```
13 printf("\nNumber of atoms along (110) plane= %f
      atoms per square mm", N_110);
14 printf("\nNumber of atoms along (111) plane= %5.3f /
      a^2 atoms per square mm", N_111);
15 // Result
16 // Number of atoms along (100) plane= 1 /a^2 atoms
      per square mm
17 //  Number of atoms along (110) plane= 0.707107
      atoms per square mm
18 //  Number of atoms along (111) plane= 0.577 /a^2
      atoms per square mm
```

Scilab code Exa 3.2 Maximum radius of sphere in BCC lattice

```
1 // Scilab Code Ex3.2 :  Page-3.5(2004)
2 clc;clear;
3 r = 1;  // For simplicity assume radius of atom to
     be unity, unit
4 a = 4*r/sqrt(3);  // Lattice constant, unit
5 R = (a/2)-r;      // R be the radius of interstitial
     sphere that can fit into void, unit
6 printf ("\nMaximum Radius of sphere that can fit
     into BCC = %5.3fr", R);
7
8 //  Result
9 //  Maximum Radius of sphere that can fit into BCC =
     0.155r
```

Scilab code Exa 3.3 Volume change during BCC to FCC

```
1 // Scilab Code Ex3.3 :  Page-3.6 (2010)
2 clc;clear;
3 r1 = 1.258e-10;  // Atomic radius in BCC, metre
```

```
4 a1 = 4*r1/sqrt(3);  // Lattice constant for BCC,
    metre
5 V1 = a1^3;   // Volume of unit cell in BCC, metre
    cube
6 Vpa = V1/2; // Volume occupied by one atom in BCC,
    metre cube
7 r2 = 1.292e-10;  // Atomic radius in FCC, metre
8 a2 = 2*r2*sqrt(2);  // Lattice constant for F    CC,
     cube
9 V2 = a2^3;   // Volume of unit cell in FCC, meter
    cube
10 Vpa1 = V2/4; // Volume occupied by one atom in FCC,
    metre cube
11 dV = (Vpa-Vpa1)/Vpa*100;    // Change in volume,
    percentage
12 printf("\nChange in volume in percentage = %4.3f
    percentage", dV);
13
14 // Result
15 // Change in volume in percentage = 0.493 percentage
```

Scilab code Exa 3.4 Volume and density of unit cell in HCP Zn structure

```
1 // Scilab Code Ex3.4 : Page-3.7 (2010)
2 clc;clear;
3 a = 0.27e-9;  // Lattice constant for BCC,  metre
4 c = 0.494e-9;  //  Height of the unit cell, metre
5 M = 65.37;  // Atomic weight of zn, kg
6 N = 6.02e+26;   // Avogadro number per k mol
7 m = 6*M/N;  // Mass per unit cell in HCP structure,
    kg
8 V = 3*sqrt(3)*a^2*c/2;   // Volume of unit cell in
    HCP, metre cube
9 rho = m/V;    // Density of HCP Zn structure, kg per
     metrecube
```

```
10
11 printf("\nVolume of HCP Zn structure = %4.3e
      metrecube", V);
12 printf("\nDensity of HCP Zn structure = %4.0f kg per
       metrecube", rho);
13
14 // Result
15 //  Volume of HCP Zn structure = 9.356e-29 metrecube
16 //  Density of HCP Zn structure = 6963 kg per
      metrecube
```

Scilab code Exa 3.5 Interplanar spacing in 110 and 212 planes in FCC lattice

```
1 /// Scilab Code Ex3.5 : Page-3.5 (2004)
2 clc;clear;
3 r = 0.1278; // Atomic radius, nm
4 a = 4*r/sqrt(2);  // Lattice constant, nm
5 h1 = 1, k1 = 1, l1 = 0;     // Miller Indices of
      (110) planes
6 d_110 = a/sqrt(h1^2 + k1^2 + l1^2);     //
      Interplanar spacing for (110) planes, nm
7 h2 = 2, k2 = 1, l2 = 2;     // Indices of third set
      of parallel planes
8 d_212 = a/sqrt(h2^2 + k2^2 + l2^2);     //
      Interplanar spacing for (111) planes, nm
9 printf("\nInterplanar spacing for (110) planes = %6
      .4f nm", d_110);
10 printf("\nInterplanar spacing for (212) planes = %6
      .4f nm", d_212);
11
12 // Result
13 // Interplanar spacing for (110) planes = 0.2556 nm
14 //  Interplanar spacing for (212) planes = 0.1205 nm
```

Scilab code Exa 3.6 Ratio of interplanar spacing in SC lattice

```
1 // Scilab Code Ex3.6 : Page-3.8 (2004)
2 clc;clear;
3 a = 1;  // For simplicity we assume a to be unity,
     unit
4 h1 = 1, k1 = 0, l1 = 0;     // Indices of first set
     of parallel planes
5 d_100 = a/sqrt(h1^2 + k1^2 + l1^2);     //
     Interplanar spacing for (100) planes, unit
6 h2 = 1, k2 = 1, l2 = 0;     // Indices of second set
     of parallel planes
7 d_110 = a/sqrt(h2^2 + k2^2 + l2^2);     //
     Interplanar spacing for (110) planes, unit
8 h3 = 1, k3 = 1, l3 = 1;     // Indices of third set
     of parallel planes
9 d_111 = a/sqrt(h3^2 + k3^2 + l3^2);     //
     Interplanar spacing for (111) planes, unit
10 printf("\nd_100 : d_110 : d_111 = %1d : %4.2f : %4.2
     f", d_100, d_110, d_111);
11
12 // Result
13 // d_100 : d_110 : d_111 = 1 : 0.71 : 0.58
```

Scilab code Exa 3.7 Miller indices of a plane in SC lattice

```
1 // Scilab Code Ex3.7 : Page-3.8 (2004)
2 clc;clear;
3 m = 1; n = 1/2; p = 3; // Coefficients of intercepts
     along three axes
4 m_inv = 1/m;          // Reciprocate the first
     coefficient
```

```
5  n_inv = 1/n;          // Reciprocate the second
       coefficient
6  p_inv = 1/p;          // Reciprocate the third
       coefficient
7  mul_fact = double(lcm(int32([1, 1, 3]))); // Find l.
       c.m. of 1, 1 and 3
8  m1 = m_inv*mul_fact;     // Clear the first fraction
9  m2 = n_inv*mul_fact;     // Clear the second fraction
10 m3 = p_inv*mul_fact;     // Clear the third fraction
11 printf("\nThe required miller indices are : (%d %d
       %d) ", m1,m2,m3);
12
13 // Result
14 // The required miller indices are : (3 6 1)
```

Scilab code Exa 3.8 Ratio of vacancies in metal

```
1  // Scilab Code Ex3.8 : Page-3.13 (2004)
2  clc;clear;
3  N = 1;  // For simplicity assume total number of
       metal ions to be unity
4  e = 1.6e-019;   // Electronic charge, C
5  k = 1.38e-023/e;  // Boltzmann constant, eV/K
6  T1 = 500;    // First temperature for metal, K
7  T2 = 1000; // Second temperature for metal, K
8  E_v = 1;    // Average energy required to create a
       vacancy in metal, eV
9  n_500 = N*exp(-E_v/(k*T1));  // Number of vacancies
       at 500 K
10 n_1000 = N*exp(-E_v/(k*T2));  // Number of vacancies
       at 500 K
11 n_ratio = n_1000/n_500;      // Ratio of vacancies in
       metal
12 printf("\nThe ratio of vacancies in metal = %5.3e",
       n_ratio);
```

```
13
14 // Result
15 // The ratio of vacancies in metal = 1.085e+05
```

Scilab code Exa 3.9 Fraction of vacancy sites in metal

```
1 // Scilab Code Ex3.9 : Page-3.14 (2004)
2 clc;clear;
3 T1 = 500+273;   // First temperature for metal, K
4 T2 = 1000+273; // Second temperature for metal, K
5 frac_vac = 1e-010;  // n1/N, the fraction of vacancy
      sites at 500 degree celsius
6 e = 1.6e-019;   // Electronic charge, C
7 k = 1.38e-023/e;  // Boltzmann constant, eV/K
8 // n1 = N*exp(-E_f/(k*T1));  // Number of vacancies
     at 500 K
9 // n2 = N*exp(-E_f/(k*T2));  // Number of vacancies
     at 500 K, solving for n2/N = x
10 x = exp((T1/T2)*log(frac_vac));
11 printf("\nThe fraction of vacancy sites in metal =
     %6.4e", x);
12
13 // Result
14 // The fraction of vacancy sites in metal = 8.4670e
     -07
```

Scilab code Exa 3.10 Average energy required to create Schottky defect

```
1 // Scilab Code Ex3.10 :Page-3.16 (2004)
2 T = 273+25; // Temperature , K
3 r = 2.82e-10;   // Interionic distance, m
4 N = 4/((2*r)^3);    // Density of ion pairs, ion
     pairs
```

```
5 k = 8.625e-5;  // Boltzmann constant, J/K
6 n = 5e+11;  // Density od Schottky effects, per unit
      volume
7 E_s = 2*k*T*2.303*log10(N/n);   // Average energy
    required to creat Schottky defect
8 printf("\nAverage energy required to create Schottky
      defect = %5.3f eV", E_s);
9
10 // Result
11 // Average energy required to create Schottky defect
      = 1.971 eV
```

Scilab code Exa 3.11 Ratio of vacancies in metal to create Frenkel defect

```
1 // Scilab Code Ex3.11 : Ratio of vacancies in metal
    to create Frenkel defect:Page-3.18 (2004)
2 N = 1;  // For simplicity assume total number of
    metal ions to be unity
3 Ni = 1; // For simplicity assume total number of
    metal ions to be unity
4 k = 8.625e-5;  // Boltzmann constant, J/K
5 T1 = 273+20;   // First temperature for metal, K
6 T2 = 300+273; // Second temperature for metal, K
7 E_v = 1.4;   // Average energy required to create a
    vacancy in metal, eV
8 n_293 = N*exp(-E_v/(2*k*T1));  // Number of
    vacancies at 500 K
9 n_573 = N*exp(-E_v/(2*k*T2));  // Number of
    vacancies at 500 K
10 n_ratio1 = n_573/n_293;      // Ratio of vacancies in
      metal
11 n_ratio2 = n_293/n_573;      // Ratio of vacancies in
      metal
12
13 printf("\nThe ratio 1 of vacancies in metal to
```

```
    create Frenkel defect = %5.3e", n_ratio1);
14 printf("\nThe ratio 2 of vacancies in metal to
    create Frenkel defect = %5.3e", n_ratio2);
15
16 // Result
17 //  The ratio 1 of vacancies in metal to create
    Frenkel defect = 7.558e+05
18 // The ratio 2 of vacancies in metal to create
    Frenkel defect = 1.323e-06
```

Chapter 4

Line Defects and Crystal Structure Determination

Scilab code Exa 4.1 Wavelength of X ray and order of diffraction

```
1 // Scilab Code Ex4.1 : Page-4.13 (2004)
2 clc;clear;
3 function thet = degree_minute (d, m)
4     thet = d + m/60;
5 endfunction
6 deg = 8, minutes = 35;    // Given glancing angle,
    degrees-minutes
7 theta = degree_minute (deg, minutes);     // Convert
    degree-minutes to degrees
8 d = 0.282;     // lattice spacing for NaCl crystal,
    nm
9 n = 1;         // Order of diffraction
10 lambda = 2*d*sind(theta)/n;    // Wavelength from
    Bragg's law, nm
11 printf("\nWavelength of X rays = %6.4f nm ", lambda)
    ;
12 theta = 90; // maximum possible value for theta for
    maximum order of diffraction
13 n = 2*d*sind(theta)/lambda;  // order of diffraction
```

```
    from Bragg's law
14 printf("\nMaximum possible order of diffraction =
   %1d", n);
15
16 // Result
17 // Wavelength of X rays = 0.0842 nm
18 // Maximum possible order of diffraction = 6
```

Scilab code Exa 4.2 Ratio of interplanar spacing in simple cube

```
1  // Scilab Code Ex4.2 : Page-4.13 (2004)
2  clc;clear;
3  a = 1;  // For simplicity we assume a to be unity,
      unit
4  h1 = 1, k1 = 0, l1 = 0;    // Indices of first set
      of parallel planes
5  d_100 = a/sqrt(h1^2 + k1^2 + l1^2);    //
      Interplanar spacing for (100) planes, unit
6  h2 = 1, k2 = 1, l2 = 0;    // Indices of second set
      of parallel planes
7  d_110 = a/sqrt(h2^2 + k2^2 + l2^2);    //
      Interplanar spacing for (110) planes, unit
8  h3 = 1, k3 = 1, l3 = 1;    // Indices of third set
      of parallel planes
9  d_111 = a/sqrt(h3^2 + k3^2 + l3^2);    //
      Interplanar spacing for (111) planes, unit
10 printf("\nd_100 : d_110 : d_111 = %1d : %4.2f : %4.2
      f", d_100, d_110, d_111);
11
12 // Result
13 // d_100 : d_110 : d_111 = 1 : 0.71 : 0.58
```

Scilab code Exa 4.3 Wavelength of X ray from Bragg Law

```
// Scilab Code Ex4.3 : Page-4.14 (2004)
clc;clear;
function thet = degree_minute (d, m)
    thet = d + m/60;
endfunction

degr = 8, minutes = 35;   // Given glancing angle, degrees-minutes
theta = degree_minute (degr, minutes);    // Convert degree-minutes to degrees
d = 0.282;    // lattice spacing for NaCl crystal, nm
n = 1;        // Order of diffraction
lambda = 2*d*sind(theta)/n;   // Wavelength from Bragg's law, nm

printf("\nWavelength of X rays = %6.4f nm ", lambda);

// Result
// Wavelength of X rays = 0.0842 nm
```

Scilab code Exa 4.4 Interatomic spacing

```
// Scilab Code Ex4.4 : Page-4.14 (2004)
clc;clear;
theta = 30; // bragg's angle, degree
lambda = 1.5418e-10;    // X-ray wavelength, m
n = 1;        // Order of diffraction
d = (n*lambda)/(2*sind(theta));   // Wavelength from Bragg's law, nm
h = 1, k = 1, l = 1;  // plane
a = d*(h^2+k^2+l^2)^(1/2); // interatomic spacing, angstrom
printf("\nWavelength of X rays = %6.4e m ", d);
```

```
10 printf("\nInteratomic spacing = %5.3f angstrom", a/1
   e-10);
11
12 // Result
13 // Wavelength of X rays = 1.5418e-10 m
14 //  Interatomic spacing = 2.670 angstrom
```

Scilab code Exa 4.5 Glancing angle for second order diffraction

```
1 // Scilab Code Ex4.5 : Page-4.14 (2004)
2 clc;clear;
3 lambda = 0.071;     // X-ray wavelength, nm
4 n = 2;         // Second order of diffraction
5 d_100 = 0.28;    // Interplanar spacing for (100)
   plane, nm
6 d_110 = d_100/sqrt(2);  // Interplanar spacing for
   (110) plane, nm
7 x = n*lambda/(2*d_110);    // sine of angle , degree
8 theta = asind(x);    // Glancing angle for second
   order diffraction
9
10 printf("\nGlancing angle for second order
   diffraction  = %d degree ", round(theta));
11
12 // Result
13 // Glancing angle for second order diffraction  = 21
    degree
```

Scilab code Exa 4.6 Distance between two adjacent atoms in NaCl

```
1 // Scilab Code Ex4.6 : Page-4.15 (2004)
2 clc;clear;
3 n = 4;         // Second order of diffraction
```

```
4 M = 58.5;    // Molecular weight of crystal
5 d = 2180;  // Density of crystal, kg per cm cube
6 N = 6.02e+26;    // Avogadro number , k / mol
7 a = ((n*M)/(d*N))^(1/3);     // Distance between two
    adjacent atoms of same kind, nm
8 b = a/2;     // Distance between two adjacent atoms
    of different kind, nm
9 printf("\nDistance between two adjacent atoms of
    same kind = %5.3f nm ", a/1e-9);
10 printf("\nDistance between two adjacent atoms of
    different kind  = %5.3f nm ", b/1e-9);
11 // Result
12
13 //  Distance between two adjacent atoms of same kind
    = 0.563 nm
14 //  Distance between two adjacent atoms of different
    kind  = 0.281 nm
```

Scilab code Exa 4.7 Distance between 110 plane

```
1 // Scilab Code Ex4.7  Page-4.16 (2004)
2 clc;clear;
3 a = 0.38;     // Lattice constant of copper, nm
4 h =1, k = 1, l = 0; // Miller Indices (hkl)= (110)
5 d = a/sqrt(h^2 + k^2 + l^2);     // Interplanar
    spacing for (110) plane, unit
6 printf("\nInterplanar spacing for (110) plane = %4.2
    f nm", d);
7
8 // Result
9 // Interplanar spacing for (110) plane = 0.27 nm
```

Scilab code Exa 4.8 Density of Iron

```
// Scilab Code Ex4.8 : Page-4.16 (2004)
clc;clear;
r = 0.123e-9;    // Atomic radius of iron, m
n = 4;           // Second order of diffraction
M = 55.8;    // Molecular weight of crystal
a = 2*r*sqrt(2);  // lattice constant, m
N = 6.023e+26;    // Avogadro number, k/mol
d = n*M/(N*a^3);     // Density of iron, kg/m-cube
printf("\nDensity of iron = %5.4d kg/metrecube ", d)
   ;

// Result
// Density of iron =  8800 kg/metrecube
```

Chapter 5

Principles of Quantum Mechanics

Scilab code Exa 5.1 de Broglie wavelength of proton

```
1 // Scilab Code Ex5.1 : Page-5.7 (2004)
2 clc;clear;
3 h = 6.626e-34;  //  Planck's const in Js
4 m = 1.67e-27;  // Mass of the proton in kg
5 c = 3e+8;     //  Charge of electron in C
6 v = c/10;    //  Proton velocity 1/10th of c
7 E = 0.025;  //  Kinetic energy of the neutron in J
8 lam = h/(m*v);     // de Broglie wavelength in m
9 printf("\nde Broglie wavelength = %5.3e m", lam);
10
11 // Result
12 // de Broglie wavelength = 1.323e-14 m
```

Scilab code Exa 5.2 de Broglie wavelength of electron

```
1 // Scilab Code Ex5.2 : de Broglie wavelength of
    electron: Page-5.8 (2004)
```

```
clc;clear;
V = 400;     // Accelerating potential, volts
lam = 12.26/sqrt(V);     // de Broglie wavelength, angstrom
printf("\nde Broglie wavelength = %5.3f angstrom", lam);

// Result
// de Broglie wavelength = 0.613 angstrom
```

Scilab code Exa 5.3 de Broglie wavelength of neutron

```
// Scilab Code Ex5.3 : Page-5.8 (2004)
clc;clear;
h = 6.626e-34;  //  Planck's const in Js
m = 1.67e-27;  // Mass of the neutron in kg
e = 1.6e-19;     //  charge of electron in C
E = 0.025;  //  kinetic energy of the neutron in J
lam = h/(sqrt(2*m*E*e));     // de Broglie wavelength in m
printf("\nde Broglie wavelength = %5.3f nm", lam/1e-9);

// Result
// de Broglie wavelength = 0.181 nm
```

Scilab code Exa 5.4 Uncertainty in momentum of electron

```
// Scilab Code Ex5.4 : Uncertainty in momentum of electron: Page-5.13 (2004)
clc;clear;
h = 6.62e-34 // Planck's const, Js
delx = 4e-10 // Uncertainty in position, m
```

```
5 delp =  h/(delx); // Uncertainity principle
6
7 printf("\nUncertainty in momentum = %5.3e kg m/sec",
      delp);
8
9 //Results
10 //  Uncertainty in momentum = 1.655000e-24 kg m/sec
```

Scilab code Exa 5.5 Uncertainty in velocity of electron

```
1 // Scilab Code Ex5.5 : Page-5.13 (2004)
2 clc;clear;
3 h = 6.62e-34; // Planck's const, Js
4 me = 9.1e-31; //Mass of electron, kg
5 delx = 1e-9; // Uncertainty in position, m
6 delp =  h/(delx); // Uncertainty principle
7 delv =  (delp/me); // Uncertainty in velocity, m/sec
8
9 printf ("\nUncertainty in velocity = %4.2e m/sec",
    delv)
10
11 // Result
12 // Uncertainty in velocity=7.2747e+05 m/sec
```

Scilab code Exa 5.6 Uncertainty in time

```
1 // Scilab Code Ex 5.6 : Uncertainty in time : Page
    -5.13 (2004)
2 clc;clear;
3 h = 6.62e-34; // planck's const, Js
4 n1 = 1; //  first state
5 n2 = 2; //  second state
```

```
En = (-13.6/n2^2)+(13.6/n1);    //Energy for
   transition , in eV
e = 1.6e-19;     //  Charge of electron , C
E2 = e*En;      // Energy for transition , J
delE = E2/100; // Uncertainty in position , m
delt =  h/(delE); // Uncertainty principle
printf("\nUncertainty in time = %1.3e sec", delt);

// Results
// Uncertainity in time = 4.056e-14 sec
```

Scilab code Exa 5.7 Lowest Energy for electron confinement

```
// Scilab Code Ex 5.7 : Lowest Energy for electron
   confinement: Page-5.22 (2004)
clc;clear;
h = 6.62e-34; // Planck's const , Js
m = 9.1e-31;     // Mass of electron in kg
L = 0.1e-9; //  Side of the box in m
n1 = 1; //    nx box lowest quantum number
n2 = 1; //    ny box lowest quantum number
n3 = 1; //    nz box lowest quantum number
e = 1.6e-19;     // Charge on electron in C
E = (h^2)/(8*e*m*L^2)*((n1)^2+(n2)^2+(n3)^2);    //
   Lowest Energy for electron confinement , in eV
printf("\nLowest Energy for electron confinement =
   %5.1f eV", E);


// Results
// Lowest Energy for electron confinement = 112.9 eV
```

Scilab code Exa 5.8 Next Lowest Energy level for electron confinement

```
// Scilab Code Ex 5.8 : Page-5.22 (2004)
clc;clear;
h = 6.62e-34; // Planck's constant, Js
m = 9.1e-31;     // Mass of electron, kg
L = 0.1e-9; //  Side of the box, m
n1 = 1; //   nx box lowest quantum number
n2 = 1; //   ny box lowest quantum number
n3 = 2; //   nz box lowest quantum number
e = 1.6e-19;     // Charge on electron, C
E = (h^2)/(8*e*m*L^2)*((n1)^2+(n2)^2+(n3)^2);    //
   Lowest Energy level for electron confinement , in
    eV

printf("\nLowest Energy level for electron
   confinement = %6.2f eV", E);

// Result
// Lowest Energy level for electron confinement =
   225.74 eV
```

Scilab code Exa 5.9 de Broglie wavelength from energy

```
// Scilab Code Ex5.9 : Page-5.23 (2004)
clc;clear;
h = 6.626e-34;  //  Planck's const in Js
m = 9.1e-31;  // Mass of the neutron in kg
e = 1.6e-19;     //  Charge of electron in C
E = 2000;  //  Kinetic energy of the neutron in eV
lam = h/(sqrt(2*m*E*e));     // de Broglie wavelength
    in m
printf("\nde Broglie wavelength of electron = %6.4f
   nm", lam/1e-9);

// Result
// de Broglie wavelength of electron = 0.0275 nm
```

Scilab code Exa 5.10 Minimum Energy

```
1 // Scilab Code Ex 5.10 : Page-5.24 (2004)
2 clc;clear;
3 h = 6.626e-34; // Planck's const, Js
4 m = 9.1e-31;    // Mass of electron, kg
5 L = 4e-10; //  Side of the box, m
6 n1 = 1; //   nx box lowest quantum number
7 E = (h^2)/(8*m*(L^2))*((n1)^2);   //Lowest Energy
    level for electron confinement , in eV
8 printf("\nMinimum Energy = %2.3e joule", E);
9
10 // Results
11 // Minimum Energy  = 3.769e-19 joule
```

Scilab code Exa 5.11 Energy of ground and first and second state

```
1 // Scilab Code Ex 5.11 : Page-5.24 (2004)
2 clc;clear;
3 h = 6.626e-34; // Planck's const, Js
4 m = 9.1e-31;    // Mass of electron in kg
5 L = 1e-10; //  Side of the box in m
6 n1 = 1; //   Ground state quantum number
7 n2 = 2; //  first quantum number
8 n3 = 3;     //  second quantum number
9 e = 1.6e-19;    // charge on electron, C
10 E1 = (h^2)/(8*m*(L^2))*((n1)^2);   //Ground state
    Energy , joule
11 E2 = (h^2)/(8*m*(L^2))*((n2)^2);   //first state
    Energy , joule
12 E3 = (h^2)/(8*m*(L^2))*((n3)^2);   //second state
    Energy , joule
```

```
13
14 printf("\nGround state Energy = %2.3e joule", E1);
15 printf("\nFirst state Energy = %2.3e joule", E2);
16 printf("\nSecond state Energy = %2.3e joule", E3);
17
18
19 // Results
20 // Ground state Energy = 6.031e-18 joule
21 // First state Energy = 2.412e-17 joule
22 // Second state Energy = 5.428e-17 joule
```

Scilab code Exa 5.12 Velocity and energy of electron

```
1 // Scilab Code Ex5.12 : Page-5.25 (2004)
2 clc;clear;
3 h = 6.626e-34;  //  Planck's const in Js
4 m = 9.1e-31;  // Mass of the electron in kg
5 lam = 1.66e-10;   //  de Broglie wavelength, m
6 e = 1.6e-19;    // Charge on electron, C
7 v = h/(m*lam);    // Velocity of electron, m/s
8 E = (m*v^2)/(2*e);  //  Kinetic energy of the
     electron in eV
9
10 printf("\nVelocity of electron = %d km/s", v/1000);
11 printf("\nKinetic energy of the electron = %5.2f eV"
     , E);
12
13 // Result
14 //  Velocity of electron = 4386 km/s
15 //  Kinetic energy of the electron = 54.71 eV
```

Scilab code Exa 5.13 Wavelength of electron waves

```
// Scilab Code Ex5.13 : Page-5.25 (2004)
clc;clear;
V = 15000;     // Accelerating potential, volts
lam = 12.26/sqrt(V);     // de Broglie wavelength,
   angstrom
printf("\nde Broglie wavelength of electron wave =
   %5.1f angstrom", lam);

// Result
// de Broglie wavelength of electron wave =   0.1
   angstrom
```

Scilab code Exa 5.14 Uncertainty in momentum of electron

```
// Scilab Code Ex5.14 : Page-5.26 (2004)
clc;clear;
h = 6.62e-34 // Planck's const, Js
delx = 1e-8 // Uncertainity in position, m
m = 9.1e-31;     //  Mass of electron, kg
delv =  h/(m*delx); // Uncertainty in velocity, km/s

printf("\nUncertainty in velocity = %3.2f km/sec",
   delv/1000);

//Results
//  Uncertainty in velocity = 72.75 km/sec
```

Chapter 6

Electron Theory of Metals

Scilab code Exa 6.1 Electron Density and mobility in silver

```
1 // Scilab Code Ex6.1 : Page-6.8 (2004)
2 clc;clear;
3 M = 107.9;          // Molecular weight of silver, kg
4 d = 10.5e+03;      // Density of of silver, kg per
    metrecube
5 N = 6.023D+26;      // Avogadro's Number., atoms/k-mol
6 a = 6.8e+07;      // conductivity of silver, per ohm
    per sec
7 e = 1.6e-19;      //  charge of electron, C
8 n = d*N/M;  //  Density of electron
9 mu = a/(n*e);     // Mobility of electron
10 printf("\nDensity of electron = %4.2e ", n);
11 printf("\nMobility of electron = %5.3e metersquare
    per volt per sec", mu);
12
13 // Result
14 // Density of electron = 5.86e+28
15 // Mobility of electron = 7.251e-03 metersquare per
    volt per sec
```

Scilab code Exa 6.2 Mobility and average time of collision in copper

```
1 // Scilab Code Ex6.2 : Page-6.9 (2004)
2 clc;clear;
3 M = 63.5;          // Molecular weight of copper, kg
4 d = 8.92e+03;      // Density of of copper, kg per
     metrecube
5 rho = 1.73e-8;     // Resistivity of copper, ohm m
6 m = 9.1e-31;      //  Mass of electron, kg
7 N = 6.023D+26;     // Avogadro's No., atoms/k-mol
8 e = 1.6e-19;      //  Charge of electron, C
9 n = d*N/M;  //  Density of electron
10 mu = 1/(rho*n*e);    // Mobility of electron
11 t = m/(n*(e^2)*rho);     // Average time of collision
     , sec
12 printf("\nNumber Density of electron = %4.2e ", n);
13 printf("\nMobility of electron = %3.3e meter per
     volt per sec", mu);
14 printf("\naverage time of collision = %3.2e sec", t)
     ;
15
16 // Result
17 //  Number Density of electron = 8.46e+28
18 //  Mobility of electron = 4.270e-03 meter per volt
     per sec
19 //  average time of collision = 2.429e-14 sec
```

Scilab code Exa 6.3 Electrical resistivity in Sodium metal

```
1 // Scilab Code Ex6.3 : Page-6.10 (2004)
2 clc;clear;
3 t = 3e-14;  // Mean free time, sec
```

```
4 m = 9.1e-31;     //  Mass of electron, kg
5 e = 1.6e-19;     // Charge of electron, C
6 r = 1.85e-10;    //  Radius of sodium atom, m
7 a = 4*r/sqrt(3);     //  Sodium has BCC structure
8 n = 2/(a^3);  //  Number of electron per unit volume
9 rho = m/(n*(e^2)*t);     // Electrical resistivity,
    ohm m
10
11 printf("\nElectrical resistivity  = %3.3e ohm m",
    rho);
12
13 // Result
14 //  Electrical resistivity  = 4.620e-08 ohm m
```

Scilab code Exa 6.4 Resistivity of sodium at zero degree Celsius

```
1 // Scilab Code Ex6.4:  Page-6.11 (2004)
2 clc;clear;
3 t = 3.1e-14;  // Mean free time, sec
4 m = 9.1e-31;     //  Mass of electron, kg
5 e = 1.6e-19;     //  Charge of electron, C
6 r = 0.429e-9;    //  Side of the unit cell. m
7 n = 2/(r^3);  //  Number of electron per unit
    cubemetre
8 rho = m/(n*(e^2)*t);     // Electrical resistivity,
    ohm m
9 printf("\nElectrical resistivity of sodium at zero
    degree Celsius = %2.3e ohm m", rho);
10
11 // Result
12 //  Electrical resistivity of sodium at zero degree
    Celsius = 4.527e-08 ohm m
```

Scilab code Exa 6.5 Next Lowest Energy level for electron confinement

```
// Scilab Code Ex 6.5 : Page-6.15 (2004)
clc;clear;
h = 6.62e-34;  // Planck's const, Js
m = 9.1e-31;    // Mass of electron in kg
L = 0.1e-9;  //  Side of the box in m
n1 = 1;       //   nx box lowest quantum number
n2 = 1;       //   ny box lowest quantum number
n3 = 1;      //   nz box lowest quantum number
e = 1.6e-19;    // Charge on electron, C
E = (h^2)/(8*e*m*L^2)*((n1)^2+(n2)^2+(n3)^2);    //
   Lowest Energy level for electron confinement , in
    eV
printf("\nLowest Energy level for electron
   confinement = %2.1f eV", E);

// Results
// Lowest Energy level for electron confinement =
   112.9 eV
```

Scilab code Exa 6.7 Energy level for electron confinement and equivalent temperature

```
// Scilab Code Ex 6.7: Page-6.17 (2004)
clc;clear;
h = 6.62e-34;    // Planck's const, Js
m = 9.1e-31;    // Mass of electron in kg
L = 1e-9;      //  length of cube box , m
n1 = 1;          //   nx box quantum number
n2 = 1;          //   ny box lowest quantum number
n3 = 2;          //   nz box quantum number
k = 1.38e-23;        //  Boltzmann constant, joule
   per kelvin
e = 1.6e-19;    // Charge on electron, C
```

```
11 E = (h^2)/(8*m*L^2)*((n1)^2+(n2)^2+(n3)^2);    //
   Lowest Energy level for electron confinement , in
    joule
12 T = 2*E/(3*k);     // Equivalent temperature of the
   molecules , kelvin
13
14 printf("\nEnergy for electron confinement = %5.3e
   joule", E);
15 printf("\nEquivalent temperature of the molecules =
   %5.3e kelvin", T);
16
17 // Results
18 //  Energy for electron confinement = 3.61e-19 joule
19 // Equivalent temperature of the molecules = 1.74e
   +04 kelvin
```

Scilab code Exa 6.9 Temperature from Fermi function

```
1 // Scilab Code Ex 6.9: Page-6.18 (2004)
2 clc;clear;
3 k = 1.38e-23;        //  Boltzmann constant , joule
   per kelvin
4 T = 300;     // For simplicity room temperature is
   taken , kelvin
5 e = 1.6e-19;     // Charge on electron , C
6 E = k*T;     //  Given Energy , eV
7 F_E = 1/(1+exp(E/(k*T)));    //Fermi function ,
   unitless
8 T = 300;     // For simplicity room temperature is
   taken , kelvin
9 printf("\nFermi fucntion = %5.3f ", F_E);
10
11 // Results
12 //  Fermi fucntion = 0.269
```

Scilab code Exa 6.10 Temperature for occupation of a state above Fermi level

```
// Scilab Code Ex 6.10: Page-6.18 (2004)
clc;clear;
k = 1.38e-23;        //  Boltzmann constant, joule per kelvin
T = 300;     // For simplicity room temperature is taken, kelvin
e = 1.6e-19;     // Charge on electron, C
EF = 5.5;     //  Fermi Energy, eV
E = EF+(EF/100);     // New energy, eV
F_E = 0.1;  // Fermi function, unitless
T = e*(E-EF)/(k*log((1-F_E)/F_E));  //Temperature for 10% probabilty that electron in silver have an energy 1% above the fermi energy, kelvin

printf("\nTemperature = %5.1f kelvin", T);

// Results
//  Temperature = 290.2 kelvin
```

Scilab code Exa 6.11 Number of energy state in unit volume

```
// Scilab Code Ex 6.11: Page-6.22 (2004)
clc;clear;
h = 6.62e-34;  // Planck's constant, Js
m = 9.14e-31;    //  Mass of the electron, kg
e = 1.6e-19;     // Charge on electron, C
E1=3.22*e;       //  First state energy, joules
E2=3.24*e;       //  Second state energy, joules
```

```
n = 8*%pi*(2*m)^(3/2)*(E2^(3/2)-E1^(3/2))/(3*h^3);
   // Number of energy state in unit volume

printf("\nNumber of energy state in unit volume= %4
   .3e ", n);

//Result
// Number of energy state in unit volume= 2.463e+26
```

Scilab code Exa 6.12 Temperature needed to fill a state above Fermi level

```
// Scilab Code Ex 6.12: Page-6.22 (2004)
clc;clear;
k = 1.38e-23;        //  Boltzmann constant, joule
   per kelvin
T = 300;     // For simplicity room temperature is
   taken, kelvin
e = 1.6e-19;     // Charge on electron, C
EF = 1;     //  For simplicity Fermi Energy is taken
   as unity, eV
E = EF+0.5;     // New energy, eV
F_E = 0.01;  // Fermi function, unitless
T = e*(E-EF)/(k*log((1-F_E)/F_E));  //Temperature
   for 1% probabilty that electron have an energy
   0.5eV above the fermi energy, kelvin

printf("\nTemperature = %d kelvin", round(T));

// Results
//  Temperature = 1262 kelvin
```

Chapter 7

Dielectric Properties

Scilab code Exa 7.1 Net energy stored in capacitor

```
1 // Scilab Code Ex7.1: Page-7.23 (2004)
2 clc;clear;
3 C = 2e-6;     // Capacitance, farad
4 V = 1000;     // Applied Voltage, volt
5 W = C*V^2/2;     // Energy stored in capacitor, joule
6 er = 100;    //Electric permittivity
7 Co = C/er;  //  New Capacitance without dielectric,
    farad
8 Wo = Co*V^2/2;  //  New Energy without dielectric,
    farad
9 Wt = W-Wo;  // Net energy stored in capacitor,
    joules
10 printf("\nNet energy stored in capacitor = %3.2f
    joules", Wt);
11
12 // Result
13 // Net energy stored in capacitor = 0.99 joules
```

Scilab code Exa 7.2 Ratio of polarization

```
1 // Scilab Code Ex7.2: Page-7.24 (2004)
2 clc;clear;
3 er = 4.94;   //Electric permittivity
4 n = sqrt(2.69); //  Index of refration, unitless
5 a = (n^2-1)/(n^2+2);    //  Variable 1
6 b = (er-1)/(er+2);  // Variable 2
7 alpha = (b/a)-1;    // Ratio between ionic and
    electrical polarization
8 alp = 1/alpha;      //Ratio between electrical and
    ionic  polarization
9
10 printf("\nRatio between ionic and electrical
    polarization = %3.3f ", alpha);
11 printf("\nRatio between electrical and ionic
    polarization = %3.3f ", alp);
12
13 // Result
14 //  Ratio between ionic and electrical polarization
    = 0.576
15 //  Ratio between electrical and ionic  polarization
    = 1.738
```

Scilab code Exa 7.3 Parallel loss resistance and capacitance

```
1 // Scilab Code Ex7.3: Page-7.25 (2004)
2 clear;clc;
3 er = 2.56;   // Relative permittivity
4 tan_deta = 0.7e-4;  // Loss tangent
5 f = 1e+6;   // frequency, Hz
6 er2 = er*tan_deta;  // Imaginary part of relative
    permittivity
7 A = 8e-4;   // Area between plates, squaremetre
8 eo = 8.854e-12; //Permittivity of the free space,
    farad per metre
9 w = 2*%pi*f;    // angular frequency, Hz
```

```
10 d = 0.08e-3;     // Distance between plates, m
11 Rp = d/(w*eo*er2*A);     // Parallel loss resistance,
       ohm
12 Cp = A*eo*er/d;          // Capacitance , farad
13
14 printf("\nParallel loss resistance = %3.3e ohm", Rp)
       ;
15 printf("\nCapacitance  = %3.3e farad", Cp);
16
17 // Result
18 //Parallel loss resistance = 1.003e+07 ohm
19 //Capacitance  = 2.267e-10 farad
```

Scilab code Exa 7.4 Dielectric constant of the material

```
1 // Scilab Code Ex7.4:  Page-7.26 (2004)
2 clc;clear;
3 N = 3e+28;    // Number density, atoms per metrecube
4 eo = 8.854e-12; //Permittivity of the free space,
      farad per metre
5 alpha = 10e-40;     // Electrical polarization, farad
       metresquare
6 er = 1+(N*alpha/eo);     // Dielectric constant of
      the material
7 printf("\nDielectric constant of the material = %3.3
      f ", er);
8
9
10 // Result
11 //  Dielectric constant of the material = 4.388
```

Scilab code Exa 7.5 Electric polarizability of He atoms

```
// Scilab Code Ex7.5: Page-7.26 (2004)
clc;clear;
N = 2.7e+25;    // Number density, atoms per
   metrecube
eo = 8.854e-12; //Permittivity of the free space,
   farad per metre
er = 1.0000684;     // Dielectric constant of the
   material
alpha = eo*(er-1)/N;     // Electrical polarization,
   farad metresquare

printf("\nElectrical polarization = %3.3e farad
   metresquare", alpha);

// Result
//  Electrical polarization = 2.243e-41 farad
   metresquare
```

Scilab code Exa 7.6 Capacitance and charge on the plates

```
// Scilab Code Ex7.6: Page-7.27 (2004)
clc;clear;
A = 100e-4;      //  Area of parallel plates,
   squaremetre
d = 1e-2;        //  Distance between plates, metre
eo = 8.854e-12; //  Permittivity of the free space,
   farad per metre
V = 100;         // Potential, volt
C = eo*A/d;      //  Capacitance, farad
Q = C*V;         // Charge on the plates of capacitor
   , C
printf("\nCapacitance = %5.3e F ", C);
printf("\nCharge on the plates of capacitor = %3.3e
   C", Q);

```

```
12 // Result
13 //  Capacitance = 8.854e-12 F
14 //  Charge on the plates of capacitor = 8.854e-10 C
```

Scilab code Exa 7.7 Electric polarizability of sulphur atoms

```
1 // Scilab Code Ex7.7:  Page-7.28 (2004)
2 clc;clear;
3 N = 385.66e+26;    // Number density, atoms per
    metrecube
4 eo = 8.854e-12; //Permittivity of the free space,
    farad per metre
5 er = 3.75;     // Dielectric constant of the material
6 alpha = 3*eo*(er-1)/(N*(er+2)); // Electric
    polarizability of sulphur atoms
7 printf("\nElectric polarizability of sulphur atoms =
    %3.3e farad metresquare", alpha);
8
9 // Result
10 //  Electric polarizability of sulphur atoms = 3.294
    e-40 farad metresquare
```

Chapter 8

Magnetic Properties

Scilab code Exa 8.1 Relative permeability

```
1 // Scilab Code Ex8.1: Page-8.33 (2004)
2 clc;clear;
3 M = 3300;     // Magnetization of ferromagnetic
     material, amp/metre
4 H = 220;     // Magnetic field strength, amp/metre
5 mu_r = M/H+1;     // Relative permeability, unitless
6 printf("\nRelative permeability = %d", mu_r);
7
8 // Result
9 // Relative permeability = 16
```

Scilab code Exa 8.2 Magnetization and flux density of ferromagnetic material

```
1 // Scilab Code Ex8.2: Page-8.33 (2004)
2 clc;clear;
3 H = 10^6;     // Magnetic field strength, amp/metre
4 ki = 1.5e-3;     // Magnetic susceptibility, units
```

```
5 M = ki*H;     // Magnetization of ferromagnetic
    material, amp/metre
6 muo = 4*%pi*1e-7;    //  Magnetic permeability, henry
    /metre
7 B = muo*(M+H);  //  Flux density,tesla
8 printf("\nMagnetization of ferromagnetic material, =
    %3.1e amp/metre", M);
9 printf("\nFlux density of ferromagnetic material, =
    %5.3f tesla", B);
10
11 // Result
12
13 //  Magnetization of ferromagnetic material, = 1.5e
    +03 amp/metre
14 //  Flux density of ferromagnetic material, = 1.259
    tesla
```

Scilab code Exa 8.3 Magnetization and flux density

```
1 // Scilab Code Ex8.3 :  Page-8.34 (2004)
2 clc;clear;
3 H = 10^4;     // Magnetic field strength, amp/metre
4 ki = 3.7e-3;     //  Magnetic susceptibility, units
5 M = ki*H;     // Magnetization of ferromagnetic
    material, amp/metre
6 muo = 4*%pi*1e-7;    //  Magnetic perbeability, henry
    /metre
7 B = muo*(M+H);  //  Flux density, weber/square meter
8 printf("\nMagnetization of ferromagnetic material, =
    %d amp/metre", M);
9 printf("\nFlux density of ferromagnetic material, =
    %3.4f weber/squaremetre ", B);
10
11 // Result
12
```

```
13 //  Magnetization of ferromagnetic material, = 37
     amp/metre
14 //  Flux density of ferromagnetic material, = 0.0126
     weber/squaremetre
```

Scilab code Exa 8.4 Magnetic moment

```
1  // Scilab Code Ex8.4: Page-8.34 (2004)
2  clc;clear;
3  d = 8906;    //  Density of nickel, kg metrecube
4  An = 6.025e+26; //  Avogadro number, per kmol
5  W = 58.7;    // Atomic weight, kg
6  N = d*An/W; //  Number of nickel atom, per cubemetre
7  Bs = 0.65;  // Saturation magnetic, weber per
     squaremetre
8  muo = (4*%pi*1e-7);    //  Magnetic perbeability,
     henry/metre
9  mum = Bs/(N*muo);  // Magnetic moment, ampere per
     sqauremetre
10 X = mum/(9.27e-24); // magnetic moment, bohr
     magneton
11
12 printf("\nNumber of nickel atom per cubemetre = %3.3
     e /cubemetre", N);
13 printf("\nMagnetic moment = %1.2e bohr magneton", X)
     ;
14
15 // Result
16
17 //  Number of nickel atom per cubemetre = 9.141e+28
     /cubemetre
18 //  Magnetic moment = 6.10e-01 bohr magneton
```

Scilab code Exa 8.5 Calculation of temperature using classical statistics

```
1 // Scilab Code Ex8.5: Page-8.35 (2004)
2 clc;clear;
3 mu = 9.4e-24;     // Magnetic moment, ampere metre
     square
4 H = 2;  //  Magnetic field , weber per squaremetre
5 k = 1.38e-23;    // Boltzmann Constant, joule per
     kelvin
6 T = (2*mu*H)/(log(2)*k);  //Temperature using
     classical statistics, K
7
8 printf("\nTemperature using classical statistics =
     %3.1f K", T);
9
10 // Result
11 // Temperature using classical statistics = 3.9 K
```

Scilab code Exa 8.6 Saturation magnetization

```
1 // Scilab Code Ex8.6: Page-8.36 (2004)
2 clc;clear;
3 A = 6.025e+26; //  Avogadro number, per k mol
4 W = 157.26;    // Atomic weight, kg
5 d = 7.8e+3;    // Density of nickel,kg metrecube
6 N = d*A/(W*1000);  //  No of atoms, per gm metrecube
7 muo = 4*%pi*1e-7;    //  Magnetic perbeability, henry
      per m
8 mum = N*7.1*(9.27e-24);  // Magnetic moment, ampere
     metersquare
9 Bs = mum*muo;  // Saturation magnetization, weber/
     squaremetre
10 printf("\nMagnetic moment = %6.4e ampere meter
     square", mum);
11 printf("\nSaturation magnetization = %6.4e weber/
```

```
    squaremetre", Bs);
12
13 // Result
14 //  Magnetic moment = 1.9669e+03 ampere meter square
15 //  Saturation magnetization = 2.4716e-03 weber/
    squaremetre
```

Scilab code Exa 8.7 Magnetic moment of nickel in Bohr Magneton

```
1  // Scilab Code Ex8.7:  Page-8.36 (2004)
2  clc;clear;
3  d = 8906;   //  Density of nickel,  kg per metrecube
4  An = 6.025e+26; //  Avogadro number, per k mol
5  W = 58.7;   // Atomic weight, kg
6  N = d*An/W; //  Number density of nickel atom,  per
    cubemetre
7  Bs = 0.65;  // Saturation magnetization,  wb per
    squaremetre
8  muo = (4*%pi*1e-7);   //  magnetic perbeability,
    henry/metre
9  mum = Bs/(N*muo);  // magnetic moment, ampere per
    squaremetre
10 X = mum/(9.27e-24); // magnetic moment, bohr
    magneton
11
12 printf("\nNumber density of nickel atom = %3.3e /
    cubemetre", N);
13 printf("\nMagnetic moment = %1.2f bohr magneton", X)
    ;
14
15 // Result
16
17 //  Number density of nickel atom per cubemetre =
    9.141e+28 /cubemetre
18 //  Magnetic moment = 6.10e-01 bohr magneton
```

Chapter 9

Semiconductors

Scilab code Exa 9.1 Resistivity

```
1 // Scilab Code Ex9.1: Page-9.24 ; (2004)
2 clc;clear;
3 ni = 2.37e+19; // Carrier concentration at room
   temperature
4 mu_e = 0.38; // mobility of electron; m^2V^-1s
   ^-1
5 e = 1.6e-19; // electronic charge, C
6 mu_h = 0.18; // mobility of holes; m^2V^-1s^-1
7 sigma = ni*e*(mu_e+mu_h); // conductivity, mho.m^-1
8 rho = 1/sigma; // Resistivity in Ge, ohm.m
9 printf("\nConductivity in Ge = %4.2f mho.per m",
   sigma);
10 printf("\nResistivity in Ge = %5.3f ohm.m", rho);
11
12 //Results
13 // Conductivity in Ge = 2.12 mho.per m
14 // Resistivity in Ge = 0.471 ohm.m
```

Scilab code Exa 9.2 Determination of Fermi level

```
// Scilab Code Ex9.2: Page-9.24 (2004)
clc;clear;
Eg = 1.12;     // Bandgap of silicon, eV
me = 0.12*9.1e-031;     // Effective Mass of the
   electron, kg
e = 1.6e-19; //  Electronic charge, C
mh = 0.28*9.1e-031;     // Effective Mass of the hole
   , kg
k = 1.38e-23;     // Boltzman constant, joule per
   kelvin
T = 300;          //  temperature, K
EF = (Eg/2)+3/4*k*T*(log(2.333))/e; // EF = E(Eg/2)
   +3/4*k*T*(log(2.333))/e;  Formula

printf("\nThe position of Fermi Level = %4.3f eV",
   EF);

// Result
// The position of Fermi Level = 0.576 eV
```

Scilab code Exa 9.3 Number of intrinsic carriers at 300K

```
// Scilab Code Ex9.3: Number of intrinsic carriers
   at 300K: Page-9.26 ; (2004)
clc;clear;
e = 1.6e-19;     //  Electronic charge, C
m   = 9.1e-31;  //  Mass of electron, kg
T = 300;     // Room temperature, K
k = 1.38e-23;    //  Boltzmann Constant, joule per
   kelvin
Eg = 0.7*e;    // Energy band gap of silicon, J
h = 6.626e-34;    // Plancks Constant, Js
C = 2*(2*%pi*m*k/h^2)^(3/2);     // A constant
ni = C*T^(3/2)*exp((-Eg)/(2*k*T));     // formula for
    carrier concentration at room temperature
```

```
11 printf("\nNumber of intrinsic carriers at 300K = %3
   .1e per cubemetre ", ni);
12
13 //Results
14 //  Number of intrinsic carriers at 300K = 3.3e+19
   per cubemetre
```

Scilab code Exa 9.4 Resistivity of Ge sample

```
1 // Scilab Code Ex9.4: Page-9.26 : (2004)
2 clc;clear;
3 ni = 2.4e+19;  // Carrier concentration at room
   temperature
4 mu_e = 0.39;    //  Mobility of electron; m^2V^-1s
   ^-1
5 e = 1.6e-19;    //  Electronic charge, C
6 mu_h = 0.19;    //  Mobility of holes, m^2V^-1s^-1
7 sigma = ni*e*(mu_e+mu_h);  // Conductivity, mho.m^-1
8 rho = 1/sigma;  //  Resistivity in Ge, ohm.m
9 printf("\nConductivity in Ge = %4.4f mho.per m",
   sigma);
10 printf("\nResistivity in Ge = %5.3f ohm.m", rho);
11
12
13 //Results
14 //  Conductivity in Ge = 2.2272 mho.per m
15 //  Resistivity in Ge = 0.449 ohm.m
```

Scilab code Exa 9.5 Resistance of Ge rod

```
1 // Scilab Code Ex9.5: Page-9.26 : (2004)
2 clc;clear;
```

```
ni = 2.5e+19;  // Carrier concentration at room
   temperature
mu_e = 0.39;    //  Mobility of electron; m^2V^-1s
   ^-1
e = 1.6e-19;    //  Electronic charge, C
l = 1e-2;   //  length of Ge rod, m
w = 1e-3;   // width of Ge rod,m
t = 1e-3;   //  thickness of Ge rod, m
A = w*t;    // Area of Ge rod, meter square
mu_h = 0.19;    //  Mobility of holes, m^2V^-1s^-1
sigma = ni*e*(mu_e+mu_h);  // Conductivity, mho.m^-1
R = 1/(sigma*A);  //  Resistivity in Ge, ohm.m
printf("\nResistance of Ge rod = %4.2e ohm", R);

//Results
//  Resistance of Ge rod = 4.31e+03 ohm
```

Scilab code Exa 9.6 Conductivity of Si

```
// Scilab Code Ex9.6: Page-9.27 ; (2004)
clc;clear;
mu_e = 0.48;    //  Mobility of electron; m^2V^-1s
   ^-1
e = 1.6e-19;    //  Electronic charge, C
m   = 9.1e-31;  //  Mass of electron, kg
mu_h = 0.013;    //  Mobility of holes, m^2V^-1s^-1
T = 300;    // Room temperature, K
k = 1.38e-23;   //  Boltzmann Constant, joule per
   kelvin
Eg = 1.1*e;   // Energy band gap of silicon, J
h = 6.626e-34;   // Plancks Constant, Js
C = 2*(2*%pi*m*k/h^2)^(3/2);    // A constant
ni = C*T^(3/2)*exp((-Eg)/(2*k*T));    // formula for
    carrier concentration at room temperature
sigma = ni*e*(mu_e+mu_h);  // Conductivity, mho per
```

```
    metre
14
15 printf("\nConductivity = %3.1e mho per metre ",
    sigma);
16
17 //Results
18 //  Conductivity = 1.2e-03 mho per metre
```

Scilab code Exa 9.7 Electron and hole concentration in silicon

```
1 // Scilab Code Ex9.7: Page-9.27 ; (2004)
2 clc;clear;
3 Na = 5e+23; // Concentration of boron atoms, per
    metrecube
4 Nd = 3e+23; // Concentration of arsenic atoms, per
    metrecube
5 p = Na-Nd;  // Hole concentration, per metrecube
6 ni = 2e+16; //  Intrinsic concentration ,per
    metrecube
7 n = ni^2/p; //  Electron concentration, per
    metrecube
8
9 printf("\nHole concentration = %3.1e per metrecube "
    , p);
10 printf("\nElectron concentration = %3.1e per
    metrecube ", n);
11
12 //Results
13 //  Hole concentration = 2.0e+23 per metrecube
14 //  Electron concentration = 2.0e+09 per metrecube
```

Scilab code Exa 9.8 Temperature that shift the fermi level

```
// Scilab Code Ex9.8:  Page-9.28 (2004)
clc;clear;
Eg = 1;     // Bandgap of silicon, eV
e = 1.6e-19; //  Electronic charge, C
k = 1.38e-23;     // Boltzman constant,joule per
    kelvin
E_F = (0.6-0.5)*e;  // Fermi energy, joules
// E_F =((Ev+Ec)/2)+3/4*k*T1*(log(4));    // Ev & Ec=
    valance and conduction band energies (formula)
T = 4*E_F/(3*k*log(4)); //Temperature that shift the
    fermi level, K

printf("\nTemperature that shift the fermi level =
    %4.3d K", T);

// Result
// Temperature that shift the fermi level = 1115 K
```

Scilab code Exa 9.9 Conductivity of intrinsic silicon at 300 K

```
// Scilab Code Ex9.9: Page-9.29 ; (2004)
clc;clear;
ni = 1.5e+16;  // Intrinsic Carrier concentration at
    room temperature
mu_e = 0.13;     //  Mobility of electron; m^2V^-1s
    ^-1
e = 1.6e-19;     //  Electronic charge, C
Nd = 4.99e+20;  // Impurity atoms , per metrecube
mu_h = 0.05;     //  Mobility of holes, m^2V^-1s^-1
sigma = ni*e*(mu_e+mu_h);  // Conductivity, mho per
    meter
sigma_d = Nd*e*mu_e;  // Conductivity with donor
    type impurities, mho per meter
sigma_a = Nd*e*mu_h;  // Conductivity with acceptor
    type impurities, mho per meter
```

```
11 printf("\nConductivity of silicon = %3.2e mho per
     meter", sigma);
12 printf("\nConductivity with donor type impurities =
     %4.2f mho per meter", sigma_d);
13 printf("\nConductivity with acceptor type impurities
     = %4.2f mho per meter", sigma_a);
14
15 //Results
16 // Conductivity of silicon = 4.32e-04 mho per meter
17 // Conductivity with donor type impurities = 10.38
     mho per meter
18 // Conductivity with acceptor type impurities= 3.99
     mho per meter
```

Scilab code Exa 9.10 Conductivity and Position of Ef above the intrinsic level

```
1 // Scilab Code Ex9.10: Page-9.31 ; (2004)
2 clc;clear;
3 ni = 1.5e+16;  // Intrinsic Carrier concentration at
     room temperature
4 mu_e = 0.135;    //  Mobility of electron; m^2V^-1s
     ^-1
5 e = 1.6e-19;     //  Electronic charge, C
6 Nd = 1e+23;  // Impurity atoms , per metrecube
7 T = 300;     // Temperature, Kelvin
8 k = 1.38e-23;     // Boltzman constant,joule per
     kelvin
9 mu_h = 0.048;     //  Mobility of holes, m^2V^-1s^-1
10 sigma = ni*e*(mu_e+mu_h);  // Conductivity, mho per
     meter
11 p = ni^2/Nd;     // Hole concentration, per metrecube
12 sigma_ex = Nd*e*mu_e;  // Conductivity with donor
     type impurities, mho per meter
13 E_F =(3/(4*e))*k*T*(log(0.135/0.048)); // Position
```

```
    of fermi level above the intrinsic level, eV
// mu is inversely propotional to mass
printf("\nConductivity of silicon = %3.2e mho per
    meter", sigma);
printf("\nHole concentration = %4.2e per metrecube",
     p);
printf("\nConductivity with donor type impurities =
    %4.2e mho per meter", sigma_ex);
printf("\nPosition of fermi level above the
    intrinsic level = %4.2f eV", E_F);

//Results
//  Conductivity of silicon = 4.39e-04 mho per meter
//  Hole concentration = 2.25e+09 per metrecube
//  Conductivity with donor type impurities = 2.16e
    +03 mho per meter
//    Position of fermi level above the intrinsic
    level = 0.02 eV
```

Scilab code Exa 9.11 Intrinsic carrier concentration and conductivity in germanium

```
// Scilab Code Ex9.11: Page-9.32 ; (2004)
clc;clear;
e = 1.6e-19;     //  Electronic charge, C
Eg = 0.7*e;  // Band gap energy, joules
mu_e = 0.4;     //  Mobility of electron; m^2V^-1s^-1
mu_h = 0.2;     //  Mobility of holes, m^2V^-1s^-1
m   = 9.1e-31;  //  Mass of electron, kg
h = 6.63e-34;    // Plancks Constant, Js
T = 300;     // Temperature, Kelvin
k = 1.38e-23;     // Boltzman constant,joule per
    kelvin
C = 2*(2*%pi*T*m*k/h^2)^(3/2); // Constant parameter
ni = C*exp((-Eg)/(2*k*T));     // Carrier
```

```
    concentration at room temperature
13 sigma = ni*e*(mu_e+mu_h);  // Conductivity, mho per
    meter
14 printf("\nCarrier concentration at room temperature
    = %4.2e per metrecube", ni);
15 printf("\nConductivity of silicon = %3.2f mho per
    meter", sigma);
16
17
18 //Results
19 //  Carrier concentration at room temperature = 3.34
    e+19 per metrecube
20 //  Conductivity of silicon = 3.20 mho per meter
```

Scilab code Exa 9.12 Forbidden energy band gap

```
1  // Scilab Code Ex9.12: Page-9.32 ; (2004)
2  clc;clear;
3  e = 1.6e-19;     //  Electronic charge, C
4  mu_e = 0.36;     //  Mobility of electron; m^2V^-1s
    ^-1
5  mu_h = 0.17;     //  Mobility of holes, m^2V^-1s^-1
6  rho = 2.12;      // Resistivity of sample, ohm metre
7  sigma = 1/rho;   // Conductivity of sample, mho per
    meter
8  m   = 9.1e-31;   //  Mass of electron, kg
9  h = 6.63e-34;    // Plancks Constant, Js
10 T = 300;     // Temperature, Kelvin
11 k = 1.38e-23;     // Boltzman constant,joule per
    kelvin
12 // But  ni = C*exp((-Eg)/(2*k*T));     // Carrier
    concentration at room temperature, therefore
13 C = 2*(2*%pi*T*m*k/h^2)^(3/2); // Constant parameter
14 ni = sigma/(e*(mu_e+mu_h));  // Carrier
    concentration, per metercube
```

```
15 b = C/ni;    // Ratio for simplicity
16 Eg = 2/e*k*T*log(b); // Band gap energy, joules
17
18 printf("\nBand gap energy = %5.4f eV", Eg);
19
20 //Result
21 //  Band gap energy= 0.7927 eV
```

Scilab code Exa 9.13 Hall Voltage of a semiconductor

```
1 // Scilab Code Ex9.13:  Page-9.45 ; (2004)
2 clc;clear;
3 RH = 3.66e-4;     // Hall coefficent, meter cube/C
4 t = 1e-03;  //  thickness of the specimen, m
5 Bz = 0.5;    // Magnetic flux density, wb per meter
     square
6 Ix = 1e-2; //  Current , A
7 VH = RH*Ix*Bz/t;     // Voltage across specimen, volt
8 printf("\nVoltage across specimen = %3.2f  millivolt
     ", VH/1e-3);
9
10 // Result
11 // Voltage across specimen = 1.83  millivolt
```

Scilab code Exa 9.14 Hall coefficient of a semiconductor

```
1 // Scilab Code Ex9.14: Hall coefficent of a
     semiconductor : Page-9.46 ; (2004)
2 clc;clear;
3 Vy = 37e-06;     // Voltage across specimen, volt
4 t = 1e-03;  //  thickness of the specimen, m
5 Bz = 0.5;    // Magnetic flux density, wb per meter
     square
```

```
Ix = 20e-3; //  Current , A
RH = Vy*t/(Ix*Bz);     // Hall coefficent , meter cube
   /C
printf("\nHall coefficent , meter cube/C = %3.1e
   meter cube/C", RH);

// Result
// Hall coefficent , meter cube/C = 3.7e-06  meter
   cube/C
```

Scilab code Exa 9.15 Mobility density and nature of semiconductor

```
// Scilab Code Ex9.15: Page-9.46 ; (2004)
clc;clear;
e = 1.6e-19;     //  Electronic charge , C
RH = -7.35e-5;     // Hall coefficent , meter cube/C
sigma = 200; //  Conductivity of the Si specimen ,
   per ohm per metre
n = -1/(RH*e);    // Electron density , per metre cube
mu = sigma/(n*e);     //  Mobility of the charge
   carriers , square meter per voly per sec
printf("\nElectron density = %3.3e per metre cube",
   n);
printf("\nMobility = %3.3f square meter per volt per
    sec", mu);
printf("\nAs the RH is negative , so specimen is n-
   type");

//Result
//  Electron density = 8.503e+22 per metre cube
//  Mobility = 0.015 square meter per volt per sec
//  As the RH is negative , so specimen is n-type
```

Scilab code Exa 9.16 Hall Voltage

```
1 // Scilab Code Ex9.16: Page-9.47 ; (2004)
2 clc;clear;
3 e = 1.6e-19;     //  Electronic charge, C
4 B = 1.5;  // Magnetic field, tesla
5 I = 50; //  Current, ampere
6 n = 8.4e+28;     // Electron density, per metre cube
7 t = 0.5e-2; //  thickness of slab, metre
8 RH = 1/(n*e);    // Hall coefficent
9 V_H = RH*I*B/t;     //  Hall voltage, volt
10 printf("\nHall Voltage = %3.3f micro volt", V_H/1e
   -6);
11
12 //Result
13 //  Hall Voltage = 1.116 micro volt
```

Scilab code Exa 9.17 Mobility and number of Charge carrier

```
1 // Scilab Code Ex9.17: Mobility and no of Charge
   carrier : Page-9.48 ; (2004)
2 clc;clear;
3 RH = 3.66e-4;  // Hall Coefficient, metrcube/C
4 e = 1.6e-19;     //  Electronic charge, C
5 rho = 8.93e-3;  // Resistivity of sample, ohm meter
6 n = 1/(RH*e);    // Number of charge carrier, per
   metre cube
7 mu_e = RH/rho;     //  Mobility of electron, m^2 per
   volt per sec
8 printf("\nNumber of charge carrier = %3.3e per metre
    cube", n);
9 printf("\nMobility of electron = %4.5f squaremetre
   per volt per sec", mu_e);
10
11 //Results
```

```
12 //  Number of charge carrier = 1.708e+22 per metre
       cube
13 //  Mobility of electron = 0.04099 m^2 per volt per
       sec
```

Chapter 10

Superconductivity

Scilab code Exa 10.1 Critical field

```
1 // Scilab Code Ex10.1: Page-10.3 (2004)
2 clc;clear;
3 Tc = 3.7;   // Critical temperature, K
4 Ho = 0.0306;    //  Critical field at zero kelvin,
    tesla
5 T = 2;  // temperature at which filed being
    calculated, K
6 Hc = Ho*[1-(T/Tc)^2]; //    Critical Magnetic field,
     tesla
7 printf("\nThe Critical field at two kelvin = %f
    tesla" , Hc);
8
9 // Result
10 // The Critical field at two kelvin = 0.021659 tesla
```

Scilab code Exa 10.2 Critical current and Critical field

```
1 // Scilab Code Ex10.2: Page-10.5 (2004)
```

```
2  clc;clear;
3  Tc = 7.18;    // Critical temperature, K
4  r = 1e-003; // diameter of wire, m
5  Ho = 6.5e+004;     //  Critical field at zero kelvin,
       A/m
6  T = 4.2;  // temperature at which filed being
       calculated, K
7  Hc = Ho*[1-(T/Tc)^2];
8  ic = %pi*r*Hc;     //  critical current , A
9  printf("\nThe Critical field  = %e A/m" , Hc);
10 printf("\nThe Critical current = %f A" , ic);
11
12 // Result
13 // The Critical field  = 4.275855e+04 A/m
14 //  The Critical current = 134.329954 A
```

Chapter 11

Lasers

Scilab code Exa 11.1 Ratio of relative population

```
1 // Scilab Code Ex11.1: Page-11.5(2004)
2 clc;clear;
3 h = 6.626e-034;     // Planck's constant, Js
4 c = 3e+08;     // Speed of light in free space, m/s
5 k = 8.61e-5;     // Boltzmann constant, J/K
6 T = 300;     // Temperature at absolute scale, K
7 lambda = 6943e-10;     // Wavelength of visible light
    , m
8 E = h*c/(lambda*1.6e-19); //  Energy in eV
9 rate_ratio = exp(-E/(k*T));     // Ratio of
    spontaneous emission to stimulated emission
10
11 printf("\nThe ratio of relative population = %3.2e",
     rate_ratio);
12 printf("\nEnergy = %3.2f eV", E);
13
14 // Results
15 //  The ratio of relative population = 8.20e-31
16 // Energy = 1.79 eV
```

Scilab code Exa 11.2 Energy of excited state of laser system

```
1 // Scilab Code Ex11.2: Page-11.16 (2004)
2 clc;clear;
3 a1 = 4; // diameter of laser beam for distance
     first, m
4 a2 = 6; // diameter of laser beam from second
     distance
5 d1 = 1; // First distance from laser to output beam
      spot, m
6 d2 = 2; // Second distance from laser to output
     beam spot, m
7 Div = (a2-a1)/(2*(d2-d1));
8 printf("\nDivergence in radian = %d milliradian",
     Div);
9
10 // Result
11 // Divergence in radian = 1 milliradian
```

Chapter 12

Fibre Optics

Scilab code Exa 12.1 Numerical aperture and acceptance angle of optical fibre

```
1 // Scilab Code Ex12.1: Page-12.6 (2004)
2 clc;clear;
3 n1 = 1.54;     // Refractive index of fibre core
4 n2 = 1.50;     // Refractive index of fibre cladding
5 NA = sqrt(n1^2 - n2^2);     // Numerical aperture for
     optical fibre
6 // As sin(theta_a) = sqrt(n1^2 - n2^2), solving for
    theta_a
7 theta_a = asind(sqrt(n1^2 - n2^2));     //
    acceptance angle of optical fibre, degrees
8 printf("\nNumerical aperture for optical fibre = %5
    .3f", NA);
9 printf("\nThe acceptance angle of optical fibre = %4
    .1f degrees", theta_a);
10
11 // Result
12 //  Numerical aperture for optical fibre = 0.349
13 //  The acceptance angle of optical fibre = 20.4
    degrees
```

Scilab code Exa 12.2 V and mode of optical fibre

```
1 // Scilab Code Ex12.2: Page-12.13 (2004)
2 clc;clear;
3 n1 = 1.53;     // Refractive index of fibre core
4 n2 = 1.50;     // Refractive index of fibre cladding
5 lamb = 1e-6;    // absolute wavelength. m
6 d = 50e-6;  //  radius of core in m
7 V = 2*%pi/lamb*d*sqrt(n1^2 - n2^2);     // volume of
     opical fibre
8 N = V^2/2;  //  no. of possible modes
9 printf("\nV of opical fibre = %6.2f ", V);
10 printf("\nNo. of possible modes = %5.3f ", N);
11
12 // Results
13 // V of opical fibre =  94.72
14 // No. of possible modes = 4485.735
```

Scilab code Exa 12.3 Loss in signal

```
1 // Scilab Code Ex12.3: Page-12.20 (2004)
2 clc;clear;
3 po = 40;     // Refractive index of fibre core
4 pi = 100;     // Refractive index of fibre cladding
5 p = po/pi;  // ratio of powers
6 L = -10*(log(po/pi));     // Refractive index of
     medium
7
8 printf("\nLoss in dB = %f dB", L);
9
10 // Result
11 // Loss in dB = 9.162907 dB
```