Scilab Textbook Companion for Solid State Physics Principles And Applications by R. Asokamani[1]

Created by
Pankaj Biswas
Solid State Physics
Physics
Shri Mata Vaishno Devi University
College Teacher
Dr. Kamni
Cross-Checked by
Lavitha Pereira

January 20, 2014

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

Book Description

Title: Solid State Physics Principles And Applications

Author: R. Asokamani

Publisher: Anamaya Publishers & New Delhi

Edition: 1

Year: 2006

ISBN: 81-88342-07-6

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

List of Figures

Chapter 1

Structure of Solids

Scilab code Exa 1.1 Lattice parameter and atomic radius of fcc structure

```
// Scilab Code Ex1.1  Page-13 (2006)
clc; clear;
r = 1.278e-010;     // Atomic radius of fcc structure, m
a = 4*r/sqrt(2);    // Lattice parameter of fcc strucure, m
V = a^3;    // Volume of fcc unit cell, metre, cube
printf("\nThe lattice parameter of fcc strucure = %4.2e m", a);
printf("\nThe volume of fcc unit cell = %5.2e metre, cube", V);

// Result
// The lattice parameter of fcc strucure = 3.61e-010 m
// The volume of fcc unit cell = 4.72e-029 metre cube
```

Scilab code Exa 1.2 Determining type of niobium cubic structure

```
// Scilab Code Ex1.2 Page-14 (2006)
clc; clear;
r = 0.143e-09;  // Radius of Nb unit cell , m
d = 8.57e+03; // Density of Nb unit cell , kg/metre-cube
M = 92.91e-03;        // Atomic weight of Nb, kg per mole
N = 6.023D+23;    // Avogadro's No.

// For fcc
a = 4*r/sqrt(2);    // Lattice parameter for fcc structure of Nb, m
n = a^3*d*N/M; // Number of lattice points per unit cell
if (modulo(n, int(n)) < 0.001) then
printf("\nThe number of atoms associated with the cell is %d, Nb should have fcc structure", int(n));
end

// For bcc
a = 4*r/sqrt(3);    // Lattice parameter for bcc structure of Nb, m
n = a^3*d*N/M; // Number of lattice points per unit cell
if (modulo(n, int(n)) < 0.001) then
printf("\nThe number of atoms associated with the cell is %d, Nb should have bcc structure", int(n));
end

// Result
// The number of atoms associated with the cell is 2, Nb should have bcc structure
```

Scilab code Exa 1.3 Lattice constants of hcp structure of Ti

```
// Scilab Code Ex1.3 : Page-17 (2006)
clc; clear;
V = 10.58e-29;  // Volume of the unit cell, metre cube
a = poly(0, 'a');       // Declare a variable
a = roots(3*sqrt(3)/2*1.58*a^3-V);        // First lattice parameter, m
c = 1.58*a(3);      // Third lattice parameter, m
printf("\nThe lattice parameters of hcp structure of Ti are:");
printf("\na = %4.2f angstorm, c = %4.2f angstorm", a(3)/1e-010, c/1e-010);

// Result
// The lattice parameters of hcp structure of Ti are:
// a = 2.95 angstorm, c = 4.67 angstorm
```

Scilab code Exa 1.4 c by a c by a ratios of Mg and Cd

```
// Scilab Code Ex1.4 : Page-17 (2006)
clc; clear;
c_by_a_ratio = 1.633;   // Ideal c/a ratio
A = cell(2,4);  // Declare a cell
// Assign values to the elements of the cell from the table
A(1,1).entries = 'Mg';
A(2,1).entries = 'Cd';
A(1,2).entries = 5.21;
A(2,2).entries = 5.62;
A(1,3).entries = 3.21;
A(2,3).entries = 2.98;
A(1,4).entries = A(1,2).entries/A(1,3).entries;
```

```
A(2,4).entries = A(2,2).entries/A(2,3).entries;
if (A(1,4).entries - c_by_a_ratio) < 0.01 then
    printf("\n%s satisfies ideal c/a ratio and %s has large deviation from this value.", A(1,1).entries, A(2,1).entries);
else if (A(1,4).entries - c_by_a_ratio) < 0.01 then
    printf("\n%s satisfies ideal c/a ratio and %s has large deviation from this value.", A(2,1).entries, A(1,1).entries);
    end
end

// Result
// Mg satisfies ideal c/a ratio and Cd has large deviation from this value.
```

Scilab code Exa 1.5 Lattice constant of NaCl unit cell

```
// Scilab Code Ex 1.5 : Page-18 (2006)
clc; clear;5
M_Na = 23;          // Atomic weight of Na, gram per mole
M_Cl = 35.5;          // Atomic weight of Cl, gram per mole
d = 2.18e+06;     // Density of Nacl salt , g per metre cube
n = 4;     // No. of atoms per unit cell for an fcc lattice of NaCl crystal
N = 6.023D+23;     // Avogadro's No.
// Volume of the unit cell is given by
// a^3 = M*n/(N*d)
// Solving for a
a = (n*(M_Na + M_Cl)/(d*N))^(1/3);     // Lattice constant of unit cell of NaCl
printf("\nLattice constant for the NaCl crystal = %4
.2f angstorm", a/1e-010);

// Result
// Lattice constant for the NaCl crystal = 5.63 angsotrm
```

Scilab code Exa 1.6 Ionic packing factor of fcc KCl

```
// Scilab Code Ex 1.6 : Page-18 (2006)
clc; clear;
r = 1.33;        // Ionic radii of K+ ion, angstrom
R = 1.81;      // Ionic radii of Cl- ion, angstrom
n = 4;     // No. of atoms per unit cell for an fcc lattice of NaCl crystal
APF = (n*(4*%pi*r^3/3)+n*(4*%pi*R^3/3))/(2*r+2*R)^3;    // Atomic packing factor of fcc KCl
printf("\nThe ionic packing factor of fcc KCl = %4.2f", APF);

// Result
// The ionic packing factor of fcc KCl = 0.56
```

Scilab code Exa 1.7 The number of atoms in unit cells of diamond and graphite

```
// Scilab Code Ex 1.7 : Page-20 (2006)
clc; clear;
N = 6.023e+23;   // Avogadro's number
M = 12.01e-03;   // Atomic weight of diamond/graphite, kg

// For diamond
```

```
a = 3.568e-010;      // Lattice parameter of diamond, m
rho = 3.518e+03;     // Density of diamond, kg per metre cube
n = a^3*rho*N/M;     // Number of atoms in the unit cell of diamond structure
printf("\nThe number of atoms in the unit cell of diamond structure = %1d", n);

// For graphite
a = 2.451e-010;      // First lattice parameter of graphite, m
c = 6.701e-010;      // Third lattice parameter of graphite, m
rho = 2.2589e+03;     // Density of graphite, kg per metre cube
V = 3*sqrt(3)*a^2*c/2;   // Volume of hexagonal unit cell of graphite, metre cube
n = V*rho*N/M;     // Number of atoms in the unit cell of graphite structure
printf("\nThe number of atoms in the unit cell of graphite structure = %2d", ceil(n));

// Result
// The number of atoms in the unit cell of diamond structure = 8
// The number of atoms in the unit cell of graphite structure = 12
```

Scilab code Exa 1.8 Densities of si and GaAs

```
// Scilab Code Ex 1.8 :Page-21 (2006)
clc; clear;
N = 6.023e+23;   // Avogadro's number

```

```
// For silicon crystallized into diamond structure
a = 5.43e-08;    // Lattice parameter of Si, cm
M = 28.1;    // Atomic mass of Si, g/mol
n = 8/a^3;         // Number of atoms per unit volume, atoms per cm cube
d = n*M/N;       // Density of Si crytal, g/cm
printf("\nThe density of crystallized Si = %4.2f gram per cm cube", d);

// For GaAs crystallized into Zinc Blende structure
a = 5.65e-08;    // Lattice parameter of GaAs, cm
M_Ga = 69.7;     // Atomic weight of Ga, g/mol
M_As = 74.9;     // Atomic weight of As, g/mol
M = M_Ga + M_As;    // Atomic weight of GaAs, g/mol
n = 4/a^3;         // Number of atoms per unit volume, atoms per cm cube
d = n*M/N;       // Density of Si crytal, g/cm
printf("\nThe density of crystallized GaAs = %5.3f gram per cm cube", d);

// Result
// The density of crystallized Si = 2.33 gram per cm cube
// The density of crystallized GaAs = 5.324 gram per cm cube 12
```

Scilab code Exa 1.9 Lattice parameters of GaP and GaAs

```
// Scilab Code Ex 1.9 :Page-21 (2006)
clc; clear;
N = 6.023e+23;  // Avogadro's number

r1 = 0.122e-09;      // Ionic radii of Ga, m
r2 = 0.125e-09;      // Ionic radii of As, m
r3 = 0.11e-09;      // Ionic radii of P, m
```

```

// For GaP
r = r1 + r3;        // Interatomic separation between Ga and P atoms, m
a = 4*r/3^(1/2);    // Lattice parameter of GaP structure, m
printf("\nThe lattice parameter of GaP structure = %5.3f angstrom", a/1e-10);

// For GaAs
r = r1 + r2;        // Interatomic separation between Ga and As atoms, m
a = 4*r/3^(1/2);    // Lattice parameter of GaP structure, m
printf("\nThe lattice parameter of GaAs structure = %4.2f angstrom", a/1e-10);

// Result
// The lattice parameter of GaP structure = 5.358 angstrom
// The lattice parameter of GaAs structure = 5.70 angstrom
```

Scilab code Exa 1.10 Crystal structures of some ionic compounds

```
// Scilab Code Ex 1.10 : Page-24 (2006)
clc; clear;
function str = structure(r_ratio)
    if r_ratio > 0.732 then
        str = 'Caesium Chloride';
    else if  r_ratio < 0.732 & r_ratio > 0.414 then
            str = 'Rock Salt';
        else if  r_ratio < 0.414 then
                str = 'Rutile'
             end
        end
    end
endfunction

crystal = cell(6,2);    // Declare cells of 6 rows
   and 2 columns
crystal(1,1).entries = ’I’;
crystal(1,2).entries = 2.19;        // Ionic radius
   of I, angstrom
crystal(2,1).entries = ’Cl’;
crystal(2,2).entries = 1.81;        // Ionic radius
   of Cl, angstrom
crystal(3,1).entries = ’Na’;
crystal(3,2).entries = 0.95;        // Ionic radius
   of Na, angstrom
crystal(4,1).entries = ’Cs’;
crystal(4,2).entries = 1.69;        // Ionic radius
   of Cs, angstrom
crystal(5,1).entries = ’Mg’;
crystal(5,2).entries = 0.99;        // Ionic radius
   of Mg2+, angstrom
crystal(6,1).entries = ’O’;
crystal(6,2).entries = 1.40;        // Ionic radius
   of O2−, angstrom

printf(”\nThe crystal structure of %s%s with radius
   ratio = %6.4f is %s”, crystal(3,1).entries,
   crystal(1,1).entries, crystal(3,2).entries/
   crystal(1,2).entries, structure(crystal(3,2).
   entries/crystal(1,2).entries));

printf(”\nThe crystal structure of %s%s with radius
   ratio = %6.4f is %s”, crystal(3,1).entries,
   crystal(2,1).entries, crystal(3,2).entries/
   crystal(2,2).entries, structure(crystal(3,2).
   entries/crystal(2,2).entries));

printf(”\nThe crystal structure of %s%s with radius
   ratio = %6.4f is %s”, crystal(4,1).entries,
   crystal(2,1).entries, crystal(4,2).entries/
   crystal(2,2).entries, structure(crystal(4,2).
   entries/crystal(2,2).entries));

printf(”\nThe crystal structure of %s%s with radius
   ratio = %6.4f is %s”, crystal(4,1).entries,
   crystal(1,1).entries, crystal(4,2).entries/
   crystal(1,2).entries, structure(crystal(4,2).
   entries/crystal(1,2).entries));

printf(”\nThe crystal structure of %s%s with radius
   ratio = %6.4f is %s”, crystal(5,1).entries,
   crystal(6,1).entries, crystal(5,2).entries/
   crystal(6,2).entries, structure(crystal(5,2).
   entries/crystal(2,2).entries));

// Result
//The crystal structure of NaI with radius ratio =
   0.4338 is Rock Salt
//The crystal structure of NaCl with radius ratio =
   0.5249 is Rock Salt
//The crystal structure of CsCl with radius ratio =
   0.9337 is Caesium Chloride
//The crystal structure of CsI with radius ratio =
   0.7717 is Caesium Chloride
//The crystal structure of MgO with radius ratio =
   0.7071 is Rock Salt
```

Scilab code Exa 1.11 Maximum radius of the sphere to fit into void between two bcc unit cells

```
// Scilab Code Ex 1.11 :Page-25 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom
```

```
to be unity, m
// For bcc Structure,
a = 4*R/sqrt(3);    // Lattice parameter of bcc crystal, m
// We have R+r = a/2, solving for r
r = a/2-R   // Relation between radius of the void and radius of the atom, m
printf("\nThe maxiumum radius of the sphere that can fit into void between two bcc unit cells = %5.3fR", r);

// Result
// The maxiumum radius of the sphere that can fit into void between two bcc unit cells = 0.155R
```

Scilab code Exa 1.12 Maximum radius of the sphere to fit into void between two fcc unit cells

```
// Scilab Code Ex 1.12 :Page-25 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom to be unity, m
// For fcc Structure,
a = 4*R/sqrt(2);    // Lattice parameter of fcc crystal, m
// We have R+r = a/2, solving for r
r = a/2-R   // Relation between radius of the void and radius of the atom, m
printf("\nThe maxiumum radius of the sphere that can fit into void between two fcc unit cells = %5.3fR", r);

// Result
// The maxiumum radius of the sphere that can fit into void between two fcc unit cells = 0.414R
```

Scilab code Exa 1.13 Radius of largest void in the bcc lattice

```
// Scilab Code Ex 1.13 :Page-26 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom to be unity, m
// For bcc Structure,
a = 4*R/sqrt(3);    // Lattice parameter of bcc crystal, m
// We have (R+r)ˆ2 = (a/2)ˆ2+(a/4)ˆ2, solving for r
r = sqrt(5)*a/4-R   // Relation between radius of the void and radius of the atom, m
printf(”\nThe radius of largest void in the bcc lattice = %4.2fR”, r);

// For fcc Structure,
a = 4*R/sqrt(2);    // Lattice parameter of fcc crystal, m
// We have (R+r)ˆ2 = (a/2)ˆ2+(a/4)ˆ2, solving for r
r_fcc = a/2-R   // Relation between radius of the void and radius of the atom, m
printf(”\nThe radius of largest void in the fcc lattice is %4.2f times larger than that in the bcc lattice”, r_fcc/r);

// Result
// The radius of largest void in the bcc lattice = 0.29R
// The radius of largest void in the fcc lattice is 1.42 times larger than that in the bcc lattice
```

Scilab code Exa 1.14 Radius of void for carbon atoms in iron

```
// Scilab Code Ex 1.14 :Page-26 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom to be unity, m

// For bcc Structure,
a = 4*R/sqrt(3);    // Lattice parameter of bcc crystal, m
// We have (R+r)^2 = (a/2)^2+(a/4)^2, solving for r
r = a/2-R   // Relation between radius of the void and radius of the atom, m
printf("\nThe radius of void for carbon atoms in iron = %5.3fR", r);

// Result
//The radius of void for carbon atoms in iron = 0.155R
```

Scilab code Exa 1.15 Radius of triangular void

```
// Scilab Code Ex 1.15 :Page-27 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom to be unity, m
// From the right triangle LMO, LM/LO = R/(R + r) = cosd(30), solving for r
r = poly(0, 'r');
r = roots(R/cosd(30)-R-r);
printf("\nThe radius of triangular void = %5.3fR", r);

// Result
// The radius of triangular void = 0.155R
```

Scilab code Exa 1.16 Radius ratio of tetrahedral void

```
// Scilab Code Ex 1.16 :Page-27 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom to be unity, m
// From the right triangle LMN similar to trinagle LPO, LM/LO = R/(R + r) = LP/LO = sqrt(2/3), solving for r
r = poly(0, 'r');
r = roots(R/sqrt(2/3)-R-r);
printf("\nThe radius ratio of tetragonal void = %5.3f", r/R);

// Result
// The radius ratio of tetragonal void = 0.225
```

Scilab code Exa 1.17 Radius ratio of octahedral void

```
// Scilab Code Ex 1.17 :Page-28 (2006)
clc; clear;
R = 1;  // For simplicity we assume radius of atom to be unity, m
// From the isosceles right triangle LMN, LM/LO = (R + r)/R = sqrt(2)/1, solving for r
r = poly(0, 'r');
r = roots(R*sqrt(2)-R-r);
printf("\nThe radius ratio of octahedral void = %5.3f", r/R);

// Result
// The radius ratio of octahedral void = 0.414
```

Scilab code Exa 1.18 Miller indices of the crystal plane

```
// Scilab Code Ex 1.18 Page-32 (2006)
clc; clear;
p = 3; q = -3; r = 3/2; // Coefficients of intercepts along three axes
h = 1/p;          // Reciprocate the first coefficient
k = 1/q;          // Reciprocate the second coefficient
l = 1/r;          // Reciprocate the third coefficient
mul_fact = double(lcm(int32([p,q,r]))); // Find l.c.m. of m,n and p
h = h*mul_fact;     // Clear the first fraction
k = k*mul_fact;     // Clear the second fraction
l = l*mul_fact;     // Clear the third fraction
printf("\nThe required miller indices are : (%d %d %d) ", h,k,l);

// Result
// The required miller indices are : (1 -1 2)
```

Scilab code Exa 1.19 Miller indices of planes cutting axes of an orthorhombic crystal

```
// Scilab Code Ex 1.19 Page-32 (2006)
clc; clear;
p = 2; q = 3; r = 4; // Coefficients of intercepts along three axes
h = 1/p;          // Reciprocate the first coefficient
k = 1/q;          // Reciprocate the second coefficient
```

```
l = 1/r;          // Reciprocate the third coefficient
mul_fact = double(lcm(int32([p,q,r]))); // Find l.c.m. of m,n and p
h = h*mul_fact;    // Clear the first fraction
k = k*mul_fact;    // Clear the second fraction
l = l*mul_fact;    // Clear the third fraction
printf("\nThe required miller indices are : (%d %d %d) ", h,k,l);

// Result
// The required miller indices are : (6 4 3)
```

Scilab code Exa 1.20 Miller indices of set of parallel planes

```
// Scilab Code Ex 1.20 Page-32 (2006)
clc; clear;
p = 4; q = 4; r = %inf; // Coefficients of intercepts along three axes
h = 1/p;          // Reciprocate the first coefficient
k = 1/q;          // Reciprocate the second coefficient
l = 1/r;          // Reciprocate the third coefficient
mul_fact = double(lcm(int32([p,q]))); // Find l.c.m. of m,n and p
h = h*mul_fact;    // Clear the first fraction
k = k*mul_fact;    // Clear the second fraction
l = l*mul_fact;    // Clear the third fraction
printf("\nThe required miller indices are : (%d %d %d) ", h,k,l);

// Result
// The required miller indices are : (1 1 0)
```

Scilab code Exa 1.21 Miller indices of planes with given intercepts

```
// Scilab Code Ex 1.21 Page-32 (2006)
clc; clear;
a = 0.424; b = 2; c = 0.367;    // Intercepts on planes along three axes, m
// Here pa = 0.424; qb = 2; rc = 0.183, solving for p, q and r, we have
p = 0.424/a; q = 2/b; r = 0.183/c; // Coefficients of intercepts along three axes
h = 1/p;        // Reciprocate the first coefficient
k = 1/q;        // Reciprocate the second coefficient
l = 1/r;        // Reciprocate the third coefficient
printf("\nThe required miller indices are : (%d %d %d) ", h,k,l);

// Result
// The required miller indices are : (1 1 2)
```

Scilab code Exa 1.22 Interplanar spacing in cubic fcc crystal

```
// Scilab Code Ex 1.22 Page-33 (2006)
clc; clear;
r = 1.746e-010;     // Atomic radius of lead atom, angstrom
a = 4*r/sqrt(2);    // Interatomic spacing, m
h = 1; k = 0; l = 0; // Miller Indices for planes in a cubic crystal
d_100 = a/(h^2+k^2+l^2)^(1/2);  // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (100) planes = %4.2f angstrom", d_100/1e-010);

```

```
h = 1; k = 1; l = 0; // Miller Indices for planes in a cubic crystal
d_110 = a/(h^2+k^2+l^2)^(1/2);  // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (110) planes = %5.3f angstrom", d_110/1e-010);

h = 1; k = 1; l = 1; // Miller Indices for planes in a cubic crystal
d_111 = a/(h^2+k^2+l^2)^(1/2);  // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (111) planes = %4.2f angstrom", d_111/1e-010);

// Result
// The interplanar spacing between consecutive (100) planes = 4.94 angstrom
// The interplanar spacing between consecutive (110) planes = 3.492 angstrom
// The interplanar spacing between consecutive (111) planes = 2.85 angstrom
```

Scilab code Exa 1.23 Wavelength of K alpha radiation of copper using Bohr atom model

```
// Scilab Code Ex 1.23 Page-34 (2006)
clc; clear;
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
h = 6.626e-034;     // Planck's constant, Js
c = 3.0e+08;    // Speed of light, m/s
E_K = 13.6*29^2;    // Energy of electron in the K-shell
E_L = 13.6*29^2/4;  // Energy of electron in the L-
```

```
shell
// As E_K - E_L = h*c/lambda, solving for lambda
lambda = h*c/((E_K - E_L)*e);        // Wavelength of K_alpha radiation of tungsten, m
printf("\nThe wavelength of K_alpha radiation of Cu = %5.3f angstrom", lambda/1e-010);

// Result
// The wavelength of K_alpha radiation of Cu = 1.448 angstrom
```

Scilab code Exa 1.24 Wavelength of K alpha radiation of tungesten

```
// Scilab Code Ex 1.24 Page-35 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
h = 6.626e-034;      // Planck's constant, Js
c = 3.0e+08;    // Speed of light, m/s
E_K = 13.6*74^2;     // Energy of electron in the K-shell
E_L = 13.6*74^2/4;   // Energy of electron in the L-shell
// As E_K - E_L = h*c/lambda, solving for lambda
lambda = h*c/((E_K - E_L)*e);        // Wavelength of K_alpha radiation of tungsten, m
printf("\nThe wavelength of K_alpha radiation of tungsten = %4.2e angstrom", lambda/1e-010);

// Result
// The wavelength of K_alpha radiation of tungsten = 2.22e-01 angstrom
```

Scilab code Exa 1.25 Lattice constants of copper palladium alloy in different proportions

```
// Scilab Code Ex 1.25 Page−35 (2006)
clc; clear;
a_Cu = 3.61;    // Lattice constant of Cu, angstrom
a_Pd = 3.89;    // Lattice constant of Pd, angstrom

// For x = 20% of Pd
x = 0.20;   // Percentage of Pd in Cu–Pd alloy
a_Cu_Pd = ((1-x)*a_Cu + x*a_Pd);
printf(”\nFor %2d percent of Pd in Cu–Pd alloy , a = %4.2f angstrom”, x*100, a_Cu_Pd);

// For x = 40% of Pd
x = 0.40;   // Percentage of Pd in Cu–Pd alloy
a_Cu_Pd = ((1-x)*a_Cu + x*a_Pd);
printf(”\nFor %2d percent of Pd in Cu–Pd alloy , a = %5.3f angstrom”, x*100, a_Cu_Pd);

// For x = 60% of Pd
x = 0.60;   // Percentage of Pd in Cu–Pd alloy
a_Cu_Pd = ((1-x)*a_Cu + x*a_Pd);
printf(”\nFor %2d percent of Pd in Cu–Pd alloy , a = %5.3f angstrom”, x*100, a_Cu_Pd);

// For x = 80% of Pd
x = 0.80;   // Percentage of Pd in Cu–Pd alloy
a_Cu_Pd = ((1-x)*a_Cu + x*a_Pd);
printf(”\nFor %2d percent of Pd in Cu–Pd alloy , a = %5.3f angstrom”, x*100, a_Cu_Pd);

// Result
// For 20 percent of Pd in Cu–Pd alloy , a = 3.67 angstrom
// For 40 percent of Pd in Cu–Pd alloy , a = 3.722 angstrom
// For 60 percent of Pd in Cu–Pd alloy , a = 3.778
```

```
angstrom
// For 80 percent of Pd in Cu-Pd alloy , a = 3.834 angstrom
```

Scilab code Exa 1.26 Amount of required Rh in Pt to change the unit cell volume

```
// Scilab Code Ex 1.26 Page-36 (2006)
clc; clear;
a_Rh = 3.80;     // Lattice constant of Rh, angstrom
a_Pt = 3.92;     // Lattice constant of Pt, angstrom
a_Pt_Rh = 3.78;     // Lattice constant of unit cell of Pt-Rh alloy , angstrom
V = (a_Pt*1e-08)^3; // Volume of unit cell of Pt, metre cube
V_90 = 0.9*V;    // 90 percent of the cell volume of Pt, metre cube

// For x = 20% of Rh in Pt-Rh alloy , we have
// a_Pt_Rh = ((1-x)*a_Pt + x*a_Rh), solving for x
x = poly(0, 'x');
x = roots (a_Pt_Rh - a_Pt + x*a_Pt - x*a_Rh); // Amount of required Rh in Pt to change the unit cell volume
printf("\nThe amount of Rh required in Pt to change the unit cell volume = %4.2f percent", x);

// Result
// The amount of Rh required in Pt to change the unit cell volume = 1.17 percent
```

Scilab code Exa 1.27 Percent volume change with the structural change

```
// Scilab Code Ex 1.27 :Page-36 (2006)
clc; clear;
r_bcc = 0.126;      // Atomic radius of the iron atoms in the bcc structure, nm
r_fcc = 0.129;       // Atomic radius of the iron atoms in the fcc structure, nm
a_bcc = 4*r_bcc/sqrt(3);
a_fcc = 4*r_fcc/sqrt(2);
V_bcc = 2*a_bcc^3;       // Volume of bcc unit cell, nm cube
V_fcc = a_fcc^3;       // Volume of fcc unit cell, nm cube
delta_V = V_fcc - V_bcc;     // Change in volume from bcc to fcc structure, nm cube
V = V_bcc;
V_frac = delta_V/V;      // Fractional change in volume from bcc to fcc structure

printf("\nThe percentage change in volume from bcc to fcc structure = %3.1f percent", V_frac*100);

// Result
// The percentage change in volume from bcc to fcc structure = -1.4 percent
```

Chapter 2

Bonding in Solids

Scilab code Exa 2.1 Binding energy of KCl

```
// Scilab Code Ex2.1 : Page-62 (2006)
clc; clear;
epsilon_0 = 8.854e-012; // Absolute electrical permittivity of free space, F/m
e = 1.6e-019;   // Energy equivalent of 1 eV, eV/J
r = 3.147e-010;     // Nearest neighbour distance for KCl, m
n = 9.1;    // Repulsive exponent of KCl
A = 1.748;  // Madelung constant for lattice binding energy
E = A*e^2/(4*%pi*epsilon_0*r)*(n-1)/n/e;     // Binding energy of KCl, eV
printf("\nThe binding energy of KCl = %5.3f eV", E);

// Result
// The binding energy of KCl = 7.110 eV
```

Scilab code Exa 2.2 Lattice energy of NaCl

```
// Scilab Code Ex2.2 : Page-62 (2006)
clc; clear;
epsilon_0 = 8.854e-012; // Absolute electrical permittivity of free space, F/m
N = 6.023e+023;     // Avogadro's number
e = 1.6e-019;   // Energy equivalent of 1 eV, eV/J
a0 = 5.63e-010;     // Lattice parameter of NaCl, m
r0 = a0/2;      // Nearest neighbour distance for NaCl, m
n = 8.4;    // Repulsive exponent of NaCl
A = 1.748;  // Madelung constant for lattice binding energy
E = A*e^2/(4*%pi*epsilon_0*r0)*(n-1)/n/e;     // Binding energy of NaCl, eV
printf("\nThe binding energy of NaCl = %5.3f kcal/mol", E*N*e/(4.186*1e+03));

// Result
// The binding energy of NaCl = 181.101 eV
```

Scilab code Exa 2.3 Nearest neighbour distance of KCl

```
// Scilab Code Ex2.3 : Page-62 (2006)
clc; clear;
epsilon_0 = 8.854e-012; // Absolute electrical permittivity of free space, F/m
N = 6.023e+023;     // Avogadro's number
e = 1.6e-019;   // Energy equivalent of 1 eV, eV/J
E = 162.9e+03;  // Binding energy of KCl, cal/mol
n = 8.6;    // Repulsive exponent of KCl
A = 1.747;  // Madelung constant for lattice binding energy
// As lattice binding energy, E = A*e^2/(4*%pi*epsilon_0*r0)*(n-1)/n, solving for r0
r0 = A*N*e^2/(4*%pi*epsilon_0*E*4.186)*(n-1)/n;
```

```
 // Nearest neighbour distance of KCl, m
printf("\nThe nearest neighbour distance of KCl = %4.2f angstorm", r0/1e-010);

// Result
// The nearest neighbour distance of KCl = 3.14 angstorm
```

Scilab code Exa 2.4 Nearest distance of CsCl

```
// Scilab Code Ex2.4 : Page-63 (2006)
clc; clear;
epsilon_0 = 8.854e-012; // Absolute electrical permittivity of free space, F/m
N = 6.023e+023;      // Avogadro's number
e = 1.6e-019;    // Energy equivalent of 1 eV, eV/J
E = 152e+03;   // Binding energy of CsCl, cal/mol
n = 10.6;     // Repulsive exponent of CsCl
A = 1.763;   // Madelung constant for lattice binding energy
// As lattice binding energy, E = A*e^2/(4*%pi*epsilon_0*r0)*(n-1)/n, solving for r0
r0 = A*N*e^2/(4*%pi*epsilon_0*E*4.186)*(n-1)/n;     // Nearest neighbour distance of CsCl, m
printf("\nThe nearest neighbour distance of CsCl = %4.2f angstrom", r0/1e-010);

// Result
// The nearest neighbour distance of CsCl = 3.48 angstrom
```

Scilab code Exa 2.5 Repulsive exponent in NaI

```
// Scilab Code Ex2.5 : Page-63 (2006)
clc; clear;
epsilon_0 = 8.854e-012; // Absolute electrical permittivity of free space, F/m
N = 6.023e+023;     // Avogadro's number
e = 1.6e-019;   // Energy equivalent of 1 eV, eV/J
r0 = 6.46e-010;     // Nearest neighbour distance of NaI
E = 157.1e+03;  // Binding energy of NaI, cal/mol
A = 1.747;  // Madelung constant for lattice binding energy
// As lattice binding energy, E = -A*e^2/(4*%pi*epsilon_0*r0)*(n-1)/n, solving for n
n = 1/(1+(4.186*E*4*%pi*epsilon_0*r0)/(N*A*e^2));  // Repulsive exponent of NaI
printf("\nThe repulsive exponent of NaI = %5.3f", n);

// Result
// The repulsive exponent of NaI = 0.363
```

Scilab code Exa 2.6 Compressibility of solid

```
// Scilab Code Ex2.6 : Page-63 (2006)
clc; clear;
e = 1.6e-019;   // Energy equivalent of 1 eV, eV/J
a0 = 2.815e-010;    // Nearest neighbour distance of solid
A = 1.747;  // Madelung constant for lattice binding energy
n = 8.6;    // The repulsive exponent of solid
c = 2;  // Structural factor for rocksalt
// As n = 1 + (9*c*a0^4)/(K0*e^2*A), solving for K0
K0 = 9*c*a0^4/((n-1)*e^2*A);        // Compressibility of solid, metre square per newton
```

```
printf("\nThe compressibility of the solid = %5.3e metre square per newton", K0);

// Result
// The compressibility of the solid = 3.325e-001 metre square per newton (Answer Given in the textbook is wrong)
```

Scilab code Exa 2.7 Percentage ionic character present in a solid

```
// Scilab Code Ex2.7 : Page-69 (2006)
clc; clear;
chi_diff = 1;    // Electronegativity difference between the constituent of elements of solid
percent_ion = 100*(1-exp(-(0.25*chi_diff^2)));  // Percentage ionic character present in solid given by Pauling
printf("\nThe percentage ionic character present in solid = %2d percent ", percent_ion);

// Result
// The percentage ionic character present in solid = 22 percent
```

Scilab code Exa 2.8 Fractional ionicity of compounds

```
// Scilab Code Ex2.8 : Page-69 (2006)
clc; clear;
A = cell(2,3);    // Declare a cell of 3X2
A(1,1).entries = 'GaAs';     // First compound name
A(1,2).entries = 4.3;        // Homopolar gap of first compound, eV
```

```
A(1,3).entries = 2.90;        // Ionic gap of first compound, eV
A(2,1).entries = 'CdTe';      // Second compound name
A(2,2).entries = 3.08;         // Homopolar gap of second compound, eV
A(2,3).entries = 4.90;        // Ionic gap of second compound, eV
printf("\nThe fractional ionicity of the compounds are given in the last column of the following table:");
printf("\nCompound      Eh      C        fi");
for i = 1:1:2
printf("\n%s           %3.1f      %4.2f     %5.3f", A(i,1).entries, A(i,2).entries, A(i,3).entries, A(i,3).entries^2/(A(i,2).entries^2+A(i,3).entries^2));  // Philips and Vanvechten model of fractional ionicity
end

// Result
// The fractional ionicity of the compounds are given in the last column of the following table:
// Compound       Eh       C         fi
// GaAs            4.3      2.90      0.313
// sCdTe            3.1      4.90      0.717
```

Chapter 3

Specific Heat of Solids and Lattice Vibrations

Scilab code Exa 3.1 Grunesien parameter for Pb

```
// Scilab Code Ex3.1: Page-79 (2006)
clc; clear;
V0 = 9.1e-05;    // Atomic volume of Pb, metre cube
   per kg
K = 2.3e-011;    // Compressibility of Pb, metre
   square per newton
alpha = 86e-06; // Coefficient of thermal expansion,
    per K
Cv = 1.4e+02;    // Specific heat at constant volume,
    J/kg
gama = alpha*V0/(K*Cv);     // Grunesien parameter
   for Pb
printf("\nThe Grunesien parameter for Pb = %3.1f",
   gama);

// Result
// The Grunesien parameter for Pb = 2.4
```

Scilab code Exa 3.2 Heat capacity of Cu

```
// Scilab Code Ex3.2: Page-79 (2006)
clc; clear;
V0 = 11e-05;    // Atomic volume of Cu, metre cube per kg
K = 0.75e-011;    // Compressibility of Cu, metre square per newton
alpha = 49e-06; // Coefficient of thermal expansion, per K
gama = 1.9;      // The Grunesien parameter for Cu = 2.4
Cv = alpha*V0/(K*gama);    // Specific heat of Cu at constant volume, J/kg
printf(”\nThe specific heat capacity of Cu = %3.1e J/kg”, Cv);

// Result
// The specific heat capacity of Cu = 3.8e+02 J/kg
```

Scilab code Exa 3.3 Debye cut off frequency of Al

```
// Scilab Code Ex3.3: Page-88 (2006)
clc; clear;
N = 6.02e+26;   // Avogadro's number, per kmole
C_t = 6.32e+03;      // Velocity of transverse wave, m/s
C_l = 3.1e+03;      // Velocity of longitudinal wave, m/s
rho = 2.7e+03;      // Density of Al, kg per metre cube
M = 26.97;   // Atomic weight of Al, gram per mol
```

```
V = M/rho;  // Atomic volume of Al, metre cube
f_c = (9*N/(4*%pi*V*(1/C_t^3+2/C_l^3)))^(1/3);
printf("\nThe Debye cut-off frequency of Al = %4.2e per sec", f_c);

// Result
// The Debye cut-off frequency of Al = 8.47e+012 per sec
```

Scilab code Exa 3.4 Specific heat capacity of diamond

```
// Scilab Code Ex3.4: Page-89 (2006)
clc; clear;
N = 6.02e+23;  // Avogadro's number, per mole
k = 1.38e-023;  // Boltzmann constant, J/K
R = N*k;    // Molar gas constant, J/mol/K
theta_D = 2230;         // Debye temperature for diamond, K
T = 300;         // Room temperature, K
C_v = 12/5*(%pi^4*R)*(T/theta_D)^3; // Specific heat capacity per unit volume of diamond, J/mol-K
printf("\nThe heat capacity per unit volume of diamond = %4.2f J/mol-K", C_v);

// Result
// The heat capacity per unit volume of diamond = 4.73 J/mol-K
```

Scilab code Exa 3.5 Debye cut off frequency of Be

```
// Scilab Code Ex3.5: Page-89 (2006)
clc; clear;
k = 1.38e-023;  // Boltzmann constant, J/K
```

```
theta_D = 1440;         // Debye temperature for Be, K
h = 6.626e-034;      // Planck's constant, Js
f_D = k*theta_D/h;       // Debye cut off frequency of Be, Hz
printf("\nThe Debye cut off frequency of Be = %g per sec", f_D);

// Result
// The Debye cut off frequency of Be = 2.99909e+013 per sec
```

Scilab code Exa 3.6 Electronic and lattice heat capacities of Cu

```
// Scilab Code Ex3.6: Page-89 (2006)
clc; clear;
N = 6.023e+023; // Avogadro's number, per kmol
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;  // Boltzmann constant, J/K
R = N*k;    // Molar gas constant, J/kmol/K
E_F = 7;    // Fermi energy of Cu, eV
theta_D = 348;  // Debye temperature of Cu, K
T = 300;    // Room temperature, K
T_F = E_F/k;    // Fermi temperature of Cu, K
C_e = %pi^2/2*R*1e+03*(T/(T_F*e));    // Electronic heat capacity of Cu, J/kmol/K
C_l = 12/5*(%pi^4*R)*(T/theta_D)^3;     // Lattice heat capacity of Cu, J/kmol/K
printf("\nThe electronic heat capacity of Cu = %3d J/kmol/K", round(C_e));
printf("\nThe lattice heat capacity of Cu = %4.2e J/mol/K", C_l);

// Result
// The electronic heat capacity of Cu = 152 J/kmol/K
// The lattice heat capacity of Cu = 1.24e+003 J/mol/K
```

Scilab code Exa 3.7 Heat capacities of Cu at a given temperature

```
// Scilab Code Ex3.7: Page-90 (2006)
clc; clear;
N = 6.023e+023; // Avogadro's number, per kmol
e = 1.602e-019;    // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;  // Boltzmann constant, J/K
R = N*k;     // Molar gas constant, J/kmol/K
E_F = 7;     // Fermi energy of Cu, eV
theta_D = 348;  // Debye temperature of Cu, K
T = 0.01;     // Room temperature, K
T_F = E_F/k;     // Fermi temperature of Cu, K
C_e = %pi^2/2*R*(T/(T_F*e));     // Electronic heat capacity of Cu, J/mol/K
C_l = 12/5*(%pi^4*R)*(T/theta_D)^3;      // Lattice heat capacity of Cu, J/kmol/K
printf(”\nThe electronic heat capacity of Cu = %4.2e J/mol/K”, C_e);
printf(”\nThe lattice heat capacity of Cu = %3.1e J/mol/K”, C_l);

// Result
// The electronic heat capacity of Cu = 5.05e-006 J/mol/K
// The lattice heat capacity of Cu = 4.6e-011 J/mol/K
```

Scilab code Exa 3.8 Electronic specific heat of Na at 20 K

```
// Scilab Code Ex3.8: Page-90 (2006)
```

```
clc; clear;
N = 6.023e+023; // Avogadro's number, per kmol
e = 1.602e-019;   // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;  // Boltzmann constant, J/K
R = N*k;    // Molar gas constant, J/kmol/K
E_F = 3.2;    // Fermi energy of Cu, eV
theta_D = 150;  // Debye temperature of Cu, K
T = 20;    // Given temperature, K
T_F = E_F/k;    // Fermi temperature of Cu, K
C_e = %pi^2/2*R*(T/(T_F*e));    // Electronic heat capacity of Cu, J/mol/K
C_l = 12/5*(%pi^4*R)*(T/theta_D)^3;     // Lattice heat capacity of Cu, J/kmol/K
printf(”\nThe electronic heat capacity of Na = %5.3e J/mol/K”, C_e);
printf(”\nThe lattice heat capacity of Na = %6.4e J/mol/K”, C_l);

// Result
// The electronic heat capacity of Na = 2.208e-002 J/mol/K
// The lattice heat capacity of Na = 4.6059e+000 J/mol/K
```

Scilab code Exa 3.9 Lattice specific heat of Hf

```
// Scilab Code Ex3.9: Page-91 (2006)
clc; clear;
N = 6.023e+023; // Avogadro's number, per kmol
e = 1.602e-019;   // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;  // Boltzmann constant, J/K
R = N*k;    // Molar gas constant, J/kmol/K
E_F = 3.2;    // Fermi energy of Hf, eV
theta_D = 242;  // Debye temperature of Hf, K
T_F = E_F/k;    // Fermi temperature of Hf, K
```

```
T = [300, 200, 100, 10, 5]; // Declare a vector of 5 temperature values, K
printf("\n________________________");
printf("\nT(K)        C_l (J/kmol/K)");
printf("\n________________________")
for i = 1:1:5
    C_l = 12/5*(%pi^4*R)*(T(i)/theta_D)^3;     // Lattice heat capacity of Hf, J/kmol/K
    printf("\n%3d        %8.3f", T(i), C_l);
end
printf("\n________________________")

// Result
// -------------------------
// T(K)        C_l (J/kmol/K)
// -------------------------
// 300         3701.863
// 200         1096.848
// 100          137.106
//  10            0.137
//   5            0.017
// -------------------------
```

Scilab code Exa 3.10 Temperature at which lattice specific heat equals electronic specific heat

```
// Scilab Code Ex3.10: Page-91 (2006)
clc; clear;
N = 6.023e+023; // Avogadro's number, per kmol
e = 1.602e-019;    // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;  // Boltzmann constant, J/K
R = N*k;     // Molar gas constant, J/kmol/K
E_F = 7;     // Fermi energy of Hf, eV
theta_D = 343;  // Debye temperature of Hf, K
T_F = E_F/k;     // Fermi temperature of Hf, K
```

```
// As C_l = 12/5*(%pi^4*R)*(T/theta_D)^3 and C_e = %pi^2/2*R*(T/(T_F*e)) so that
// For C_l = C_e, we have
T = sqrt((%pi^2/2*R*1/(T_F*e))/(12/5*%pi^4*R)*theta_D^3); // Required temperature when C_l = C_e, K
printf("\nThe temperature at which lattice specific heat equals electronic specific heat for Cu = %4.2f K", T);

// Result
// The temperature at which lattice specific heat equals electronic specific heat for Cu = 3.24 K
```

Scilab code Exa 3.11 Debye temperature of Al

```
// Scilab Code Ex3.11: Page-92 (2006)
clc; clear;
C11 = 1.08e+12, C12 = 0.62e+12, C44 = 0.28e+12; // Elastic constants of Al, dynes/cm square
a = 4.05e-08;   // Lattice constant for Al cubic structure, cm
rho = 2.70;     // g/cm cube
k = 1.38e-023;  // Boltzmann constant, J/K
h = 6.626e-034; // Planck's constant, Js
s = 4;  // Number of atoms in Al unit cell
Va = a^3;   // Volume of unit cell, cm cube
theta_D = (3.15/(8*%pi)*(h/k)^3*s/(rho^(3/2)*Va)*(C11-C12)^(1/2)*(C11+C12+2*C44)^(1/2)*C44^(1/2))^(1/3);
printf("\nThe Debye temperature of Al = %3d K", theta_D);

// Result
// The Debye temperature of Al = 466 K
```

Scilab code Exa 3.12 Debye temperatures of Cu and Na

```
// Scilab Code Ex3.12: Page-93 (2006)
clc; clear;
k = 1.38e-023;  // Boltzmann constant, J/K
h = 6.626e-034; // Planck's constant, Js
A = cell(2,8);  // Declare a matrix of 2X8
A(1,1).entries = 'Cu';
A(1,2).entries = 1.684e+012;
A(1,3).entries = 1.214e+012;
A(1,4).entries = 0.754e+012;
A(1,5).entries = 4;
A(1,6).entries = 3.61e-08;
A(1,7).entries = 8.96;
A(2,1).entries = 'Na';
A(2,2).entries = 0.055e+012;
A(2,3).entries = 0.047e+012;
A(2,4).entries = 0.049e+012;
A(2,5).entries = 2;
A(2,6).entries = 4.225e-08;
A(2,7).entries = 0.971;

// For Cu
Va = A(1,6).entries^3;    // Volume of unit cell, cm cube
A(1,8).entries = (3.15/(8*%pi)*(h/k)^3*A(1,5).entries/(A(1,7).entries^(3/2)*Va)*(A(1,2).entries-A(1,3).entries)^(1/2)*(A(1,2).entries+A(1,3).entries+2*A(1,4).entries)^(1/2)*A(1,4).entries^(1/2))^(1/3);

// For Na
Va = A(2,6).entries^3;    // Volume of unit cell, cm cube
```

```
A(2,8).entries = (3.15/(8*%pi)*(h/k)^3*A(2,5).entries/(A(2,7).entries^(3/2)*Va)*(A(2,2).entries-A(2,3).entries)^(1/2)*(A(2,2).entries+A(2,3).entries+2*A(2,4).entries)^(1/2)*A(2,4).entries^(1/2))^(1/3);

printf("\n________________________________________");
printf("\nMetal       C11       C12       C44       thetaD")
printf("\n________________________________________");
for i = 1:1:2
    printf("\n%s          %5.3f    %5.3f    %5.3f    %3d", A(i,1).entries, A(i,2).entries/1e+12, A(i,3).entries/1e+12, A(i,4).entries/1e+12, A(i,8).entries);
end
printf("\n________________________________________");

// Result
// ----------------------------------------
// Metal       C11       C12       C44       thetaD
// ----------------------------------------
// Cu          1.684     1.214     0.754     380
// Na          0.055     0.047     0.049     150
// ----------------------------------------
```

Scilab code Exa 3.13 Debye temperature as a function of temperature

```
// Scilab Code Ex3.13: Page-93 (2006)
clc; clear;
k = 1.38e-023;  // Boltzmann constant, J/K
h = 6.626e-034; // Planck's constant, Js
A = cell(4,5);  // Declare a matrix of 4X5
```

```
A(1,1).entries = 300;
A(1,2).entries = 0.878e+010;
A(1,3).entries = 0.483e+010;
A(1,4).entries = 0.448e+010;
A(2,1).entries = 200;
A(2,2).entries = 0.968e+010;
A(2,3).entries = 0.508e+010;
A(2,4).entries = 0.512e+010;
A(3,1).entries = 100;
A(3,2).entries = 1.050e+010;
A(3,3).entries = 0.540e+010;
A(3,4).entries = 0.579e+010;
A(4,1).entries = 20;
A(4,2).entries = 1.101e+010;
A(4,3).entries = 0.551e+010;
A(4,4).entries = 0.624e+010;
s = 2;      // Number of atoms in a unit cell
a = 4.225e-10;  // Lattice parameter of Na, m
rho = 0.971e+03;    // Density of Na, kg/metre-cube
Va = a^3;   // Volume of unit cell , metre cube
printf("\n________________________________________");
printf("\nT         C11       C12       C44       thetaD")
printf("\n________________________________________");
for i=1:1:4
   A(i,5).entries = (3.15/(8*%pi)*(h/k)^3*s/(rho^(3/2)*Va)*(A(i,2).entries-A(i,3).entries)^(1/2)*(A(i,2).entries+A(i,3).entries+2*A(i,4).entries)^(1/2)*A(i,4).entries^(1/2))^(1/3);
printf("\n%3d      %5.3f    %5.3f    %5.3f    %3d", A(i,1).entries, A(i,2).entries/1e+10, A(i,3).entries/1e+10, A(i,4).entries/1e+10, A(i,5).entries);
end
printf("\n________________________________________");

// Result
// ----------------------------------------
// T        C11      C12      C44      thetaD
// ----------------------------------------
// 300      0.878    0.483    0.448    197
// 200      0.968    0.508    0.512    210
// 100      1.050    0.540    0.579    222
//  20      1.101    0.551    0.624    229
// ----------------------------------------
// The theta values given in the textbook are wrong
```

Scilab code Exa 3.14 Variation of Gruneisen frequency and Debye temperature for Lu with pressure

```
// Scilab Code Ex3.12: Page-93 (2006)
clc; clear;
Lu = cell(6,5);  // Declare a matrix of 6X5
Lu(1,1).entries = 0;
Lu(1,2).entries = 5.58;
Lu(1,3).entries = 3.517;
Lu(1,5).entries = 0.750;
Lu(2,1).entries = 36;
Lu(2,2).entries = 5.409;
Lu(2,3).entries = 3.440;
Lu(2,5).entries = 0.560;
Lu(3,1).entries = 103;
Lu(3,2).entries = 5.213;
Lu(3,3).entries = 3.341;
Lu(3,5).entries = 0.492;
Lu(4,1).entries = 157;
Lu(4,2).entries = 5.067;
Lu(4,3).entries = 3.259;
Lu(4,5).entries = 0.388;
Lu(5,1).entries = 191;
Lu(5,2).entries = 4.987;
Lu(5,3).entries = 3.217;
Lu(5,5).entries = 0.357;
Lu(6,1).entries = 236;
Lu(6,2).entries = 4.921;
Lu(6,3).entries = 3.179;
Lu(6,5).entries = 0.331;
V0 = 3*sqrt(3)/2*Lu(1,3).entries^2*Lu(1,2).entries;
V = zeros(6);   // Declare volume array
printf("\
   n-----------------------------------------------------------------
   ");
printf("\nP(kbar)     c(angstrom)      a(angstrom)
   gamma_G        nu_G    ");
printf("\
   n-----------------------------------------------------------------
   ");
for i=1:1:6
   V(i) = 3*sqrt(3)/2*Lu(i,3).entries^2*Lu(i,2).
      entries;
   Lu(i,4).entries = Lu(i,5).entries*V(i)/V0+2/3*(1-
      V(i)/V0)^(1/2);
printf("\n%3d         %5.3f              %5.3f
   %5.3f         %5.3f", Lu(i,1).entries, Lu(i,2).
   entries, Lu(i,3).entries, Lu(i,4).entries, Lu(i
   ,5).entries);
end
printf("\
   n-----------------------------------------------------------------
   ");

cnt = 0;
printf("\n------------------------");
printf("\nP(kbar)        Theta_D(K)");
printf("\n------------------------");
for i=1:1:6
    theta_D = exp(integrate('-1*Lu(i,5).entries*exp(
       x)/V0-2/3*(1-exp(x)/V0)^(1/2)', 'x', -0.8+cnt
       , log(V(i)/1000000)));
    cnt = cnt + 0.01;
    printf("\n%3d              %3.0f", Lu(i,1).entries, theta_D);
end
printf("\n________________________");

// Result
// ---------------------------------------------------------------
// P(kbar)    c(angstrom)     a(angstrom)    gamma_G       nu_G
// ---------------------------------------------------------------
//   0          5.580              3.517           0.750         0.750
//  36          5.409              3.440           0.699         0.560
// 103          5.213              3.341           0.679         0.492
// 157          5.067              3.259           0.615         0.388
// 191          4.987              3.217           0.602         0.357
// 236          4.921              3.179           0.591         0.331
// ---------------------------------------------------------------
// ________________________
// P(kdbar)         Theta_D(K)
// ________________________
//   0                 185
//  36                 195
// 103                 210
// 157                 222
// 191                 230
// 236                 237
// ------------------------
```

Scilab code Exa 3.15 Lindemann rule to calculate the Debye temperature

```
// Scilab Code Ex3.15: Page-94 (2006)
clc; clear;
T_M = 1356;      // Melting temperature of Cu, K
V = 7.114;       // Atomic volume of Cu, cm cube per g-atom
M = 63.5;        // atomic weight of Cu, g/mole
K = 138.5;       // Lindemann constant
theta_M = K*(T_M/M)^(1/2)*(1/V)^(1/3); // Debye temperature by Lindemann method, K

printf("\nThe Debye temperature by Lindemann method = %3d K", ceil(theta_M));
printf("\nThe values obtained from other methods are:");
printf("\ntheta_s = 342 K;      theta_R = 336 K;      theta_E = 345 K");

// Result
// The Debye temperature by Lindemann method = 333 K
// The values obtained from other methods are:
// theta_s = 342 K;      theta_R = 336 K;      theta_E = 345 K
```

Scilab code Exa 3.16 Frequency of vibration of ions in InSb crystal

```
// Scilab Code Ex3.16: Page-100 (2006)
clc; clear;
N_A = 6.023e+023;        // Avogadro's number
c = 3.0e+08;     // Speed of light, m/s
```

```
epsilon_0 = 15; // Dielectric constant of the medium
m = 2.0e-022;         // Mass of ion, g
e = 4.8e-010;    // Charge on the ion, C
rho = 7;    // Average density of solid, g/cc
A = 120;    // Average atomic weight of solid, g
N = rho/A*N_A;  // Number of ions per cc, per cm cube
f_P = 1/(2*%pi)*sqrt(4*%pi*N*e^2/(m*epsilon_0));        // Plasma frequency of vibrating ions in the crystal, Hz
lambda_P = c/f_P;    // Plasma wavelength of vibrating ions in the crystal, cm
printf("\nThe plasma frequency of vibrating ions in InSb crystal = %3.1e Hz", f_P);
printf("\nThe plasma wavelength of vibrating ions in InSb crystal = %3d micron", lambda_P/1e-06);
printf("\nThe calculated frequency lies in the infrared region.")

// Result
// The plasma frequency of vibrating ions in InSb crystal = 9.3e+011 Hz
// The plasma wavelength of vibrating ions in InSb crystal = 323 micron
// The calculated frequency lies in the infrared region.
```

Scilab code Exa 3.17 Debye temperature for diamond

```
// Scilab Code Ex3.17: Page-103 (2006)
clc; clear;
h = 6.624e-034;      // Planck's constant, Js
k = 1.38e-023;  // Boltzmann constant, J/mol/K
q = 1.486e+011;      // Young's modulus of diamond, N/metre-square
```

```
rho = 3500;      // Density of diamond, kg/metre−cube
c = sqrt(q/rho);     // Speed of transverse wave through diamond, m/s
m = 12*1.66e-027;      // Atomic weight of carbon, kg
theta_D = (h/k)*c*(3*rho/(4*%pi*m))^(1/3);     // Debye temperature for diamond, K
printf(”\nThe Debye temperature for diamond = %4d K”, theta_D);

// Result
// The Debye temperature for diamond = 1086 K
```

Chapter 4

Free Electron Theory of Metals

Scilab code Exa 4.1 Collision time for an electron in monovalent Cu

```
// Scilab Code Ex4.1: Page-112 (2006)
clc; clear;
m = 9.1e-031;    // Mass of an electron , kg
e = 1.6e-019;    // Charge on an electron , C
n = 8.5e+028;    // Concentration of electron in Cu, per metre cube
rho = 1.7e-08;   // Resistivity of Cu, ohm-m
t = m/(n*e^2*rho);    // Collision time for an electron in monovalent Cu, s
printf("\nThe collision time for an electron in monovalent Cu = %3.1e s", t);

// Result
// The collision time for an electron in monovalent Cu = 2.5e-014 s
```

Scilab code Exa 4.2 Relaxation time and mean free path at 0K

```
// Scilab Code Ex4.2: Page-112 (2006)
clc; clear;
m = 9.1e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Charge on an electron, C
n = 1e+029;    // Concentration of electron in material, per metre cube
rho = 27e-08;   // Resistivity of the material, ohm-m
tau = m/(n*e^2*rho);   // Collision time for an electron in the material, s
v_F = 1e+08;     // Velocity of free electron, cm/s
lambda = v_F*tau;    // Mean free path of electron in the material, cm
printf("\nThe collision time for an electron in monovalent Cu = %3.1e s", tau);
printf("\nThe mean free path of electron at 0K = %3.1e cm", lambda);

// Result
// The collision time for an electron in monovalent Cu = 1.3e-015 s
// The mean free path of electron at 0K = 1.3e-007 cm
```

Scilab code Exa 4.3 Free electron density and electrical conductivity of monovalent Cu

```
// Scilab Code Ex4.3: Page-112 (2006)
clc; clear;
m = 9.1e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Charge on an electron, C
r = 1.28e-010;   // Atomic radius of cupper, m
a = 4*r/sqrt(2);     // Lattice parameter of fcc structure of Cu, m
V = a^3;     // Volume of unit cell of Cu, metre cube
n = 4/V;     // Number of atoms per unit volume of Cu
```

```
, per metre cube
tau = 2.7e-04;  // Relaxation time for an electron in monovalent Cu, s
sigma = n*e^2*tau/m;    // Electrical conductivity of Cu, mho per cm
printf("\nThe free electron density in monovalent Cu = %5.3e per metre cube", n);
printf("\nThe electrical conductivity of monovalent Cu = %5.3e mho per cm", sigma);

// Result
// The free electron density in monovalent Cu = 8.429e+028 per metre cube
// The electrical conductivity of monovalent Cu = 6.403e+017 mho per cm
```

Scilab code Exa 4.4 Energy difference between two levels for the free electrons

```
// Scilab Code Ex4.4: Page-118 (2006)
clc; clear;
m = 9.1e-031;   // Mass of an electron, kg
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
h = 6.625e-034;     // Planck's constant, Js
L = 10e-03;     // Length of side of the cube, m
// For nth level
nx = 1, ny = 1, nz = 1;     // Positive integers along three axis
En = h^2/(8*m*L^2)*(nx^2+ny^2+nz^2)/e;      // Energy of nth level for electrons, eV
// For (n+1)th level
nx = 2, ny = 1, nz = 1;     // Positive integers along three axis
En_plus_1 = h^2/(8*m*L^2)*(nx^2+ny^2+nz^2)/e;   // Energy of (n+1)th level for electrons, eV
```

```
delta_E = En_plus_1 - En;        // Energy difference between two levels for the free electrons
printf("\nThe energy difference between two levels for the free electrons = %4.2e eV", delta_E);

// Result
// The energy difference between two levels for the free electrons = 1.13e-014 eV
```

Scilab code Exa 4.5 Probability of the electron in tungsten at room temperature

```
// Scilab Code Ex4.5: Page-119 (2006)
clc; clear;
T = 300;     // Room temperature of tungsten, K
k = 1.38e-023;  // Boltzmann constant, J/mol/K
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
E_F = 4.5*e;    // Fermi energy of tungsten, J
E = E_F-0.1*E_F; // 10% energy below Fermi energy, J
f_T = 1/(1+exp((E-E_F)/(k*T)));      // Probability of the electron in tungsten at room temperature at an nergy 10% below the Fermi energy
printf("\nThe probability of the electron at an energy 10 percent below the Fermi energy in tungsten at 300 K = %4.2f", f_T);
E = 2*k*T+E_F; // For energy equal to 2kT + E_F
f_T = 1/(1+exp((E-E_F)/(k*T)));      // Probability of the electron in tungsten at an energy 2kT above the Fermi energy
printf("\nThe probability of the electron at an energy 2kT above the Fermi energy = %6.4f", f_T);

// Result
// The probability of the electron at an energy 10 percent below the Fermi energy in tungsten at 300
```

```
K = 1.00
// The probability of the electron at an energy 2kT above the Fermi energy = 0.1192
```

Scilab code Exa 4.6 Fermi energy of a monovalent bcc solid

```
// Scilab Code Ex4.6: Page-121 (2006)
clc; clear;
h = 6.625e-034; // Planck's constant, Js
h_cross = h/(2*%pi);    // Reduced Planck's constant, Js
m = 9.1e-031;   // Mass of an electron, kg
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
a = 5.34e-010;      // Lattice constant of monovalent bcc lattice, m
V = a^3;    // Volume of bcc unit cell, metre cube
n = 2/V;    // Number of atoms per metre cube
E_F = h_cross^2/(2*m*e)*(3*%pi^2*n)^(2/3);    // Fermi energy of monovalent bcc solid, eV

printf("\nThe Fermi energy of a monovalent bcc solid = %5.3f eV", E_F);

// Result
// The Fermi energy of a monovalent bcc solid = 2.034
```

Scilab code Exa 4.7 Number of states at Fermi energy

```
// Scilab Code Ex4.7: Page-121 (2006)
clc; clear;
h = 6.625e-034; // Planck's constant, Js
```

```
h_cross = h/(2*%pi);     // Reduced Planck's constant, Js
m = 9.11e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
V = 1e-05;     // Volume of cubical box, metre cube
E_F = 5*e;  // Fermi energy, J
D_EF = V/(2*%pi^2)*(2*m/h_cross^2)^(3/2)*E_F^(1/2)*e;      // Density of states at Fermi energy, states/eV
printf("\nThe density of states at Fermi energy = %4.2e states/eV", D_EF);

// Result
// The density of states at Fermi energy = 1.52e+023 states/eV
```

Scilab code Exa 4.8 Energy separation between adjacent energy levels of Mg and Ca

```
// Scilab Code Ex4.8: Page-121 (2006)
clc; clear;
h = 6.626e-034; // Planck's constant, Js
h_cross = h/(2*%pi);     // Reduced Planck's constant, Js
m = 9.1e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
V = 1e-06;     // Volume of cubical box, metre cube
E_F = 7.13*e;   // Fermi energy for Mg, J
D_EF = V/(2*%pi^2)*(2*m/h_cross^2)^(3/2)*E_F^(1/2);        // Density of states at Fermi energy for Cs, states/eV
E_Mg = 1/D_EF;   // The energy separation between adjacent energy levels of Mg, J
printf("\nThe energy separation between adjacent energy levels of Mg = %5.3e eV", E_Mg/e);
```

```
E_F = 1.58*e;  // Fermi energy for Cs, J
D_EF = V/(2*%pi^2)*(2*m/h_cross^2)^(3/2)*E_F^(1/2);       // Density of states at Fermi energy for Mg, states/eV
E_Mg = 1/D_EF;  // The energy separation between adjacent energy levels of Cs, J
printf("\nThe energy separation between adjacent energy levels of Cs = %5.3e eV", E_Mg/e);

// Result
// The energy separation between adjacent energy levels of Mg = 5.517e-023 eV
// The energy separation between adjacent energy levels of Cs = 1.172e-022 eV
```

Scilab code Exa 4.9 Fermi momentum of sodium

```
// Scilab Code Ex4.9: Page-122 (2006)
clc; clear;
m = 9.1e-031;   // Mass of an electron, kg
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
E_F = 3.2*e;    // Fermi energy of sodium, J
P_F = sqrt(E_F*2*m);    // Fermi momentum of sodium, kg-m/s
printf("\nThe Fermi momentum of sodium = %5.3e kg-m/sec", P_F);

// Result
// The Fermi momentum of sodium = 9.653e-025 kg-m/sec
```

Scilab code Exa 4.10 Change in Fermi energy with temperature

```
// Scilab Code Ex4.10: Page-122 (2006)
clc; clear;
k = 1.38e-023;   // Boltzmann constant, J/mol/K
T = 500;     // Rise in temperature of Al, K
EF_0 = 11.63;     // Fermi energy of Al, eV
EF_T = EF_0*(1-%pi^2/12*(k*T/EF_0)^2);   // Change in Fermi energy of Al with temperature, eV
printf("\nThe change in Fermi energy of Al with tempertaure rise of 500 degree celsius = %5.2f eV", EF_T);

// Result
// The change in Fermi energy of Al with tempertaure rise of 500 degree celsius = 11.63 eV
```

Scilab code Exa 4.11 Electrical conductivity of Pb and Ag

```
// Scilab Code Ex4.11: Page-122 (2006)
clc; clear;
m = 9.1e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Charge on an electron, C
lambda = 1.0e-09;    // Mean free path of electron in metal, m
v = 1.11e+05;     // Average velocity of the electron in metal, m/s

// For Lead
n = 13.2e+028;   // Electronic concentration of Pb, per metre cube
sigma = n*e^2*lambda/(m*v);      // Electrical conductivity of lead, mho per metre
printf("\nThe electrical conductivity of lead = %4.2e mho per metre", sigma);

// For Silver
```

```
n = 5.85e+28;  // Electronic concentration of Ag, per metre cube
sigma = n*e^2*lambda/(m*v);     // Electrical conductivity of Ag, mho per metre
printf("\nThe electrical conductivity of silver = %4.2e mho per metre", sigma);

// Result
// The electrical conductivity of lead = 3.35e+007 mho per metre
// The electrical conductivity of silver = 1.48e+007 mho per metre
```

Scilab code Exa 4.12 Lorentz number

```
// Scilab Code Ex4.12: Page-125 (2006)
clc; clear;
k = 1.38e-023;  // Boltzmann constant, J/mol/K
e = 1.6e-019;   // Charge on an electron, C
L = %pi^2/3*(k/e)^2;    // Lorentz number, watt-ohm/degree-square
printf("\nThe Lorentz number = %4.2e watt-ohm/degree-square", L);

// Result
// The Lorentz number = 2.45e-008 watt-ohm/degree-square
```

Scilab code Exa 4.13 Lorentz numbers for metals at 273 K and comparison with the given values

```
// Scilab Code Ex4.13: Page-125 (2006)
clc; clear;
```

```
A = cell(4,4);      // Declare a 4X4 cell
A(1,1).entries = 'Mg';
A(1,2).entries = 2.54e-05;
A(1,3).entries = 1.5;
A(1,4).entries = 2.32e+02;
A(2,1).entries = 'Cu';
A(2,2).entries = 6.45e-05;
A(2,3).entries = 3.85;
A(2,4).entries = 2.30e+02;
A(3,1).entries = 'Al';
A(3,2).entries = 4.0e-05;
A(3,3).entries = 2.38;
A(3,4).entries = 2.57e+02;
A(4,1).entries = 'Pt';
A(4,2).entries = 1.02e-05;
A(4,3).entries = 0.69;
A(4,4).entries = 2.56e+02;
T1 = 273;   // First temperature, K
T2 = 373;   // Second temperature, K
printf("\n-----------------------------------------------------------------");
printf("\nMetal     sigma x 1e-05    K(W/cm-K)    Lorentz number          ");
printf("\n          (mho per cm)                        (watt-ohm/deg-square)x1e-02")
printf("\n-----------------------------------------------------------------");
for i = 1:1:4
    L1 = A(i,3).entries/(A(i,2).entries*T1); L2 = A(i,4).entries;
    printf("\n%s        %4.2f            %4.2f            %4.2f            %4.2f", A(i,1).entries, A(i,2).entries/1e-05, A(i,3).entries, L1/1e+02, L2/1e+02);
end
printf("\
n-----------------------------------------------------------
");

// Result
// ------------------------------------------------------------
// Metal      sigma x 1e-05    K(W/cm-K)     Lorentz number
//            (mho per cm)                   (watt-ohm/deg-square)x1e-02
// ------------------------------------------------------------
// Mg         2.54              1.50           2.16          2.32
// Cu         6.45              3.85           2.19          2.30
// Al         4.00              2.38           2.18          2.57
// Pt         1.02              0.69           2.48          2.56
// ------------------------------------------------------------
```

Scilab code Exa 4.14 Thermal conductivity of gold at 100 K and 273 K

```
// Scilab Code Ex4.14: Page-125 (2006)
clc; clear;
A = cell(2,2);       // Declare a 2X3 cell
A(1,1).entries = 1.6e+08;   // Electrcal conductivity of Au at 100 K, mho per metre
A(1,2).entries = 2.0e-08;    // Lorentz number of Au at 100 K, volt/K-square
```

```
A(2,1).entries = 5.0e+08;  // Electrcal conductivity of Au at 273  K, mho per metre
A(2,2).entries = 2.4e-08;   // Lorentz number of Au at 273 K, volt/K-square
T1 = 100;        // First temperature, K
T2 = 273;        // Second temperature, K

printf("\n------------------------------------------------------------");
printf("\n           T = 100 K                              T = 273 K                   ");
printf("\n-----------------------------------------------------------------------");
printf("\nElectrical conductivity)  L     Electrical conductivity)  L        ");
printf("\n     mho per metre        V/K-square           mho per metre        V/K-square");
printf("\n------------------------------------------------------------");
K1 = A(1,1).entries*T1*A(1,2).entries; K2 = A(2,1).entries*T2*A(2,2).entries;
    printf("\n%3.1e                      %3.1e        %3.1e                      %3.1e", A(1,1).entries, A(1,2).entries, A(2,1).entries, A(2,2).entries);
    printf("\nK = %3d W/cm-K                                 K = %3d W/cm-K", K1, K2);
printf("\n------------------------------------------------------------");

// Result
// ------------------------------------------------------------
//            T = 100 K                              T = 273 K
// -------------------------------- ----------------------------------
// Electrical conductivity)  L  Electrical conductivity)  L
//     mho per metre       V/K-square           mho per metre       V/K-square
// ------------------------------------------------------------------------------------
// 1.6e+008                  2.0e-008      5.0e+008                 2.4e-008
// K = 320 W/cm-K                           K = 3276 W/cm-K
// ------------------------------------------------------------------------------------
```

Scilab code Exa 4.15 Hall coefficient of sodium

```
// Scilab Code Ex4.15: Page-131 (2006)
clc; clear;
e = 1.6e-019;   // Electronic charge, C
a = 0.428e-09;  // Lattice constant of Na, m
V = a^3;    // Volume of unit cell, metre cube
N = 2;  // No. of atoms per unit cell of Na
n = N/V;    // No. of electrons per metre cube, per metre cube
R_H = -1/(n*e);  // Hall coeffcient of Na, metre cube per coulomb
printf("\nThe Hall coefficient of sodium = %4.2e metre cube per coulomb", R_H);

```

```
// Result
// The Hall coefficient of sodium = -2.45e-010 metre cube per coulomb
```

Scilab code Exa 4.16 Hall coefficient of beryllium

```
// Scilab Code Ex4.16: Page-131 (2006)
clc; clear;
e = 1.6e-019;    // Electronic charge, C
n = 24.2e+028;     // No. of electrons per metre cube, per metre cube
R_H = -1/(n*e);  // Hall coeffcient of Be, metre cube per coulomb
printf("\nThe Hall coefficient of beryllium = %4.2e metre cube per coulomb", R_H);

// Result
// The Hall coefficient of beryllium = -2.58e-011 metre cube per coulomb
```

Scilab code Exa 4.17 Electronic concentration of silver from Hall coefficient

```
// Scilab Code Ex4.17: Page-131 (2006)
clc; clear;
e = 1.6e-019;    // Electronic charge, C
R_H = -8.4e-011;  // Hall coeffcient of Ag, metre cube per coulomb
n = -3*%pi/(8*R_H*e);     // Electronic concentration of Ag, per metre cube
printf("\nThe electronic concentration of Ag = %3.1e per metre cube", n);

```

```
// Result
// The electronic concentration of Ag = 8.8e+028 per metre cube
```

Scilab code Exa 4.18 Resistivity of a metal using Matthiessen rule

```
// Scilab Code Ex4.18: Page-134 (2006)
clc; clear;
// We have from Mattheissen rule, rho = rho_0 + alpha*T1
T1 = 300;   // Initial temperature, K
T2 = 1000;  // Final temperature, K
rho = 1e-06;    // Resistivity of the metal, ohm-m
delta_rho = 0.07*rho;       // Increase in resistivity of metal, ohm-m
alpha = delta_rho/(T2-T1);  // A constant, ohm-m/K
rho_0 = rho - alpha*T1;     // Resistivity at room temperature, ohm-m
printf("\nThe resistivity at room temperature = %4.2e ohm-m", rho);

// Result
// The resistivity at room temperature = 1.00e-006 ohm-m
```

Scilab code Exa 4.19 Resitivity of Ge at 20 degree celsius

```
// Scilab Code Ex4.19: Page-134 (2006)
clc; clear;
// We have from Mattheissen rule, rho = rho_0 + alpha*T1
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;  // Boltzmann constant, J/mol/K
```

```
rho_40 = 0.2;   // Resistivity of Ge at 40 degree celsius, ohm-m
E_g = 0.7;  // Bandgap for Ge, eV
T1 = 20+273;    // Second temperature, K
T2 = 40 + 273;  // First temperature, K
rho_20 = rho_40*exp(E_g*e/(2*k)*(1/T1-1/T2));    // Resistivity of Ge at 20 degree celsius, ohm-m
printf("\nThe resistivity of Ge at 20 degree celsius = %3.1f ohm-m", rho_20);

// Result
// The resistivity of Ge at 20 degree celsius = 0.5 ohm-m
```

Scilab code Exa 4.20 Solid radius and Fermi level quantities for Li

```
// Scilab Code Ex4.20: Page-135 (2006)
clc; clear;
rs_a0_ratio = 3.25;     // Ratio of solid radius to the lattice parameter
E_F = 50.1*(rs_a0_ratio)^(-2);  // Fermi level energy of Li, eV
T_F = 58.2e+04*(rs_a0_ratio)^(-2);  // Fermi level temperature of Li, K
V_F = 4.20e+08*(rs_a0_ratio)^(-1);  // Fermi level velocity of electron in Li, cm/sec
K_F = 3.63e+08*(rs_a0_ratio)^(-1);
printf("\nE_F = %4.2f eV", E_F);
printf("\nT_F = %4.2e K", T_F);
printf("\nV_F = %4.2e cm/sec", V_F);
printf("\nK_F = %4.2e per cm", K_F);

// Result
// E_F = 4.74 eV
// T_F = 5.51e+004 K
// V_F = 1.29e+008 cm/sec
// K_F = 1.12e+008 per cm
```

Scilab code Exa 4.21 Fermi energy for yittrium

```
// Scilab Code Ex4.21: Page-135 (2006)
clc; clear;
n = 6.04e+022;  // Concentration of electrons in yittrium, per metre cube
r_s = (3/(4*%pi*n))^(1/3)/1e-08;    // Radius of the solid, angstrom
a0 = 0.529; // Lattice parameter of yittrium, angstrom
rs_a0_ratio = r_s/a0;   // Solid radius to lattice parameter ratio
E_F = 50.1*(rs_a0_ratio)^(-2);  // Fermi level energy of Y, eV
printf("\nThe Fermi energy of yittrium = %5.3f eV", E_F);
Ryd = 13.6;     // Rydberg energy constant, eV
E_bs = 0.396*Ryd;   // Band structure energy value of Y, eV
printf("\nThe band structure value of E_F = %5.3f eV is in close agreement with the calculated value of %5.3f eV", E_bs, E_F);

// Result
// The Fermi energy of yittrium = 5.608 eV
// The band structure value of E_F = 5.386 eV is in close agreement with the calculated value of 5.608 eV
```

Scilab code Exa 4.22 Plasmon energy of Al using free electron gas parameter

```
// Scilab Code Ex4.22: Page-137 (2006)
clc; clear;
rs_a0_ratio = 2.07;    // Solid radius to lattice parameter ratio for Al
E_F = 50.1*(rs_a0_ratio)^(-2);   // Fermi level energy of Y, eV
// According to Jellium model, h_cross*omega_P = E = 47.1 eV *(rs_a0_ratio)^(-3/2)
E = 47.1*(rs_a0_ratio)^(-3/2);       // Plasmon energy of Al, eV
printf("\nThe plasmon energy of Al = %4.2f eV", E);
printf("\nThe experimental value is 15 eV");

// Result
// The plasmon energy of Al = 15.81 eV
// The experimental value is 15 eV
```

Scilab code Exa 4.23.1 Occupation probability of an electron at a given temperature

```
// Scilab Code Ex4.1a: Page-137 (2006)
clc; clear;
E_F = 1;     // For simplicity assume Fermi energy to be unity, eV
k = 1.38e-023;       // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
dE = 0.1;          // Exces energy above Fermi level, eV
T = 300;          // Room temperature, K
E = E_F + dE;     // Energy of the level above Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);     // Occupation
```

```
probability of the electron at 0.1 eV above E_F
printf("\nAt 300 K:");
printf("\n==========");
printf("\nThe occupation probability of electron at %3.1f eV above Fermi energy = %7.5f", dE, f_E);
E = E_F - dE;     // Energy of the level below Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);    // Occupation probability of the electron at 0.1 eV below E_F
printf("\nThe occupation probability of electron at %3.1f eV below Fermi energy = %7.5f", dE, f_E);

T = 1000;          // New temperature, K
printf("\n\nAt 1000 K:");
printf("\n==========");
E = E_F + dE;     // Energy of the level above Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);    // Occupation probability of the electron at 0.1 eV above E_F
printf("\nThe occupation probability of electron at %3.1f eV above Fermi energy = %4.2f", dE, f_E);
E = E_F - dE;     // Energy of the level below Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);    // Occupation probability of the electron at 0.1 eV below E_F
printf("\nThe occupation probability of electron at %3.1f eV below Fermi energy = %4.2f", dE, f_E);

// Result
// At 300 K:
// ==========
// The occupation probability of electron at 0.1 eV above Fermi energy = 0.02054
// The occupation probability of electron at 0.1 eV below Fermi energy = 0.97946

// At 1000 K:
// ==========
// The occupation probability of electron at 0.1 eV above Fermi energy = 0.24
// The occupation probability of electron at 0.1 eV below Fermi energy = 0.76
```

Scilab code Exa 4.23.2 Variation of occupation probability with temperature

```
// Scilab Code Ex4.2a: Page-138 (2006)
clc; clear;
f_E = 0.01;     // Occupation probability of electron
E_F = 1;    // For simplicity assume Fermi energy to be unity, eV
k = 1.38e-023;      // Boltzmann constant, J/mol/K
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
dE = 0.5;       // Exces energy above Fermi level, eV
E = E_F + dE;    // Energy of the level above Fermi level, eV
// We have, f_E = 1/(exp((E-E_F)*e/(k*T))+1), solving for T
T = (E-E_F)*e/k*1/log(1/f_E-1);   // Temperature at which the electron will have energy 0.1 eV above the Fermi energy, K
printf(”\nThe temperature at which the electron will have energy %3.1f eV above the Fermi energy = %4d K”, dE, T);

// Result
// The temperature at which the electron will have energy 0.5 eV above the Fermi energy = 1261 K
```

Scilab code Exa 4.23.3 Average energy and speed of free electron in metal

```
// Scilab Code Ex4.3a: Page-139 (2006)
clc; clear;
E_F = 10;    // Fermi energy of electron in metal, eV
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
m = 9.1e-031;    // Mass of an electron, kg
E_av = 3/5*E_F;      // Average energy of free
   electron in metal at 0 K, eV
V_F = sqrt(2*E_av*e/m);    // Speed of free electron
   in metal at 0 K, eV
printf("\nThe average energy of free electron in
   metal at 0 K = %1d eV", E_av);
printf("\nThe speed of free electron in metal at 0 K
    = %4.2e m/s", V_F);

// Result
// The average energy of free electron in metal at 0
    K = 6 eV
// The speed of free electron in metal at 0 K = 1.45
   e+006 m/s
```

Scilab code Exa 4.23.4 Temperature dependence of occupation probability

```
// Scilab Code Ex4.4a: Page-139 (2006)
clc; clear;
f_E = 0.1;      // Occupation probability of electron
E_F = 5.5;     // Fermi energy of Cu, eV
k = 1.38e-023;       // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
dE = 0.05*E_F;          // Exces energy above Fermi
   level, eV
E = E_F + dE;     // Energy of the level above Fermi
   level, eV
```

```
// We have, f_E = 1/(exp((E-E_F)*e/(k*T))+1), solving for T
T = (E-E_F)*e/k*1/log(1/f_E-1);    // Temperature at which the electron will have energy 0.1 eV above the Fermi energy, K
printf("\nThe temperature at which the electron will have energy %1d percent above the Fermi energy %4d K", dE/E_F*100, T);


// Result
// The temperature at which the electron will have energy 5 percent above the Fermi energy 1451 K (The answer given in the textbook is wrong)
```

Scilab code Exa 4.23.5 Fermi velocity of Potassium

```
// Scilab Code Ex4.5a: Page-139 (2006)
clc; clear;
T_F = 24600;    // Fermi temperature of potassium, K
k = 1.38e-023;   // Boltzmann constant, J/mol/K
m = 9.1e-031;    // Mass of an electron, kg
E_F = k*T_F;     // Fermi energy of potassium, eV
v_F = sqrt(2*k*T_F/m);       // Fermi velocity of potassium, m/s
printf("\nThe Fermi velocity of potassium = %5.3e m/s", v_F);

// Result
// The Fermi velocity of potassium = 8.638e+005 m/s
```

Scilab code Exa 4.23.6 Energy level of Cu for given occupation probability

```
// Scilab Code Ex4.6a: Page-139 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
E_F = 7.0;   // Fermi energy of Cu, eV
f_E = 0.9;   // Occupation probability of Cu
k = 1.38e-023;   // Boltzmann constant, J/mol/K
T = 1000;    // Given temperature, K
// We have, f_E = 1/(exp((E-E_F)*e/(k*T))+1), solving for E
E = k*T*log(1/f_E-1) + E_F*e;      // Energy level of Cu for 10% occupation probability at 1000 K, J
printf("\nThe energy level of Cu for 10 percent occupation probability at 1000 K = %4.2f eV", E/e);

// Result
// The energy level of Cu for 10 percent occupation probability at 1000 K = 6.81 eV
```

Scilab code Exa 4.23.7 Electronic concentration in cesium

```
// Scilab Code Ex4.7a: Page-140 (2006)
clc; clear;
m = 9.1e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Electronic charge, C
h = 6.626e-034;  // Planck's constant, Js
E_F = 1.55;   // Fermi energy of Cu, eV
n = %pi/3*(8*m/h^2)^(3/2)*(E_F*e)^(3/2);     // Electronic concentration in cesium, electrons/cc
printf("\nThe electronic concentration in cesium = %5.3e electrons/cc", n);

// Result
// The electronic concentration in cesium = 8.733e+027 electrons/cc
```

Scilab code Exa 4.23.8 Fermi temperature corresponding to Fermi energy

```
// Scilab Code Ex4.8a: Page-141 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
E_F = 7;   // Fermi energy, eV
k = 1.38e-023;   // Boltzmann constant, J/mol/K
T_F = E_F*e/k;     // Fermi temperature, K
printf("\nThe Fermi temperature corresponding to
   Fermi energy = %5.3e K", T_F);

// Result
// The Fermi temperature corresponding to Fermi
   energy = 8.116e+004 K
```

Scilab code Exa 4.23.9 Density of states for the electron in a cubical box

```
// Scilab Code Ex4.9a: Page-141 (2006)
clc; clear;
m = 9.1e-031;    // Mass of the electron, kg
h = 6.626e-034;  // Planck's constant, Js
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
h_cross = h/(2*%pi);     // Reduced Planck's constant
   , Js
s = 0.01;    // Side of the box, m
E = 2;   // Energy range of the electron in the box,
   eV
V = s^3;     // Volume of the box, metre cube
I = integrate("E^(1/2)", 'E', 0, 2);     // Definite
   integral over E
```

```
D_E = V/(2*%pi^2)*(2*m/h_cross^2)^(3/2)*I*e^(3/2);    // Density of states for the electron in a cubical box, states
printf("\nThe density of states for the electron in a cubical box = %5.3e states", D_E);

// Result
// The density of states for the electron in a cubical box = 1.280e+022 states
```

Scilab code Exa 4.23.10 Occupation probability of an electron above and below Fermi energy

```
// Scilab Code Ex4.10a: Page-141 (2006)
clc; clear;
E_F = 1;    // For simplicity assume Fermi energy to be unity, eV
k = 1.38e-023;      // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
dE = 0.5;        // Exces energy above Fermi level, eV
T = 300;        // Room temperature, K
E = E_F + dE;     // Energy of the level above Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);    // Occupation probability of the electron at 0.1 eV above E_F
printf("\nAt 300 K:");
printf("\n===========");
printf("\nThe occupation probability of electron at %3.1f eV above Fermi energy = %11.9f", dE, f_E);
E = E_F - dE;     // Energy of the level below Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);    // Occupation probability of the electron at 0.1 eV below E_F
printf("\nThe occupation probability of electron at
%3.1f eV below Fermi energy = %11.9f", dE, f_E);

// Result
// At 300 K:
// ==========
// The occupation probability of electron at 0.5 eV above Fermi energy = 0.000000004
// The occupation probability of electron at 0.5 eV below Fermi energy = 0.999999996
```

Scilab code Exa 4.23.11 Occupation probability at two different temperatures

```
// Scilab Code Ex4.9a: Page-141 (2006)
clc; clear;
E_F = 1;     // For simplicity assume Fermi energy to be unity, eV
k = 1.38e-023;       // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
dE = 0.2;         // Exces energy above Fermi level, eV
T = 0+273;         // Room temperature, K
E = E_F + dE;     // Energy of the level above Fermi level, eV
f_E = 1/(exp((E-E_F)*e/(k*T))+1);     // Occupation probability of the electron at 0.1 eV above E_F
printf("\nAt 273 K:");
printf("\n============");
printf("\nThe occupation probability of electron at %3.1f eV above Fermi energy = %4.2e", dE, f_E);
T = 100+273;     // Given temperature of 100 degree celsius, K
f_E = 1/(exp((E-E_F)*e/(k*T))+1);     // Occupation probability of the electron at 0.1 eV below E_F
printf("\n\nAt 373 K:");
printf("\n==============");
printf("\nThe occupation probability of electron at %3.1f eV above Fermi energy = %4.2e", dE, f_E);

// Result
// At 273 K:
// ==========
// The occupation probability of electron at 0.2 eV above Fermi energy = 2.05e-004

// At 373 K:
// ==========
// The occupation probability of electron at 0.2 eV above Fermi energy = 1.99e-003
```

Scilab code Exa 4.23.12 Concentration of free electrons and electrical conductivity in Cu

```
// Scilab Code Ex4.12a: Page-142 (2006)
clc; clear;
m = 9.1e-031;   // Mass of the electron, kg
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
r = 1.28e-010;   // Atomic radius of Cu, m
a = 4*r/sqrt(2);    // Lattice constant of Cu, m
tau = 2.7e-14;      // Relaxation time for the electron in Cu, s
V = a^3;    // Volume of the cell, metre cube
n = 4/V;    // Concentration of free electrons in monovalent copper,
sigma = n*e^2*tau/m;    // Electrical conductivity of monovalent copper, mho per m
printf("\nThe electrical conductivity of monovalent copper = %5.3e mho per cm", sigma/100);

// Result
// The electrical conductivity of monovalent copper = 6.403e+005 mho per cm
```

Scilab code Exa 4.23.13 Interelectronic energy separation between bands of Al

```
// Scilab Code Ex4.13a: Page-142 (2006)
clc; clear;
n = 18.1e+022;  // Number of electrons per unit volume, per cm cube
N = n/2;    // Pauli's principle for number of energy levels, per cm cube
E_F = 11.58;     // Fermi energy of Al, eV
E = E_F/N;     // Interelectronic energy separation between bands of Al, eV
printf("\nThe interelectronic energy separation between bands of Al = %4.2e eV", E);

// Result
// The interelectronic energy separation between bands of Al = 1.28e-022 eV
```

Scilab code Exa 4.23.14 Density of states in Cu contained in cubic metal

```
// Scilab Code Ex4.14a: Page-142 (2006)
clc; clear;
m = 9.1e-031;   // Mass of the electron, kg
h = 6.626e-034; // Planck's constant, Js
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
h_cross = h/(2*%pi);    // Reduced Planck's constant, Js
E_F = 7;    // Fermi energy of Cu, eV
V = 1e-06;  // Volume of the cubic metal, metre cube
```

```
D_EF = V/(2*%pi^2)*(2*m/h_cross^2)^(3/2)*(E_F)^(1/2)*e^(3/2);   // Density of states in Cu contained in cubic metal, states/eV
printf("\nThe density of states in Cu contained in cubic metal = %3.1e states/eV", D_EF);

// Result
// The density of states in Cu contained in cubic metal = 1.8e+022 states/eV
```

Scilab code Exa 4.23.15 Electronic energy level spacing between successive levels of Cu

```
// Scilab Code Ex4.15a: Page-143 (2006)
clc; clear;
m = 9.1e-031;    // Mass of the electron, kg
h = 6.626e-034;  // Planck's constant, Js
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
h_cross = h/(2*%pi);     // Reduced Planck's constant, Js
E_F = 7;     // Fermi energy of Cu, eV
V = 1e-06;   // Volume of the cubic metal, metre cube
D_EF = V/(2*%pi^2)*(2*m/h_cross^2)^(3/2)*(E_F)^(1/2)*e^(3/2);   // Density of states in Cu contained in cubic metal, states/eV
d = 1/(D_EF);      // Electronic energy level spacing between successive levels of Cu, eV
printf("\nThe electronic energy level spacing between successive levels of Cu = %4.2e eV", d);

// Result
// The electronic energy level spacing between successive levels of Cu = 5.57e-023 eV
```

Scilab code Exa 4.23.16 Energy band gaps of the solids

```
// Scilab Code Ex4.16a: Page-143 (2006)
clc; clear;
A = cell(4,2);  // Declare a 3X2 cell
A(1,1).entries = 'Li';      //
A(1,2).entries = -0.4039;   // Energy of outermost atomic orbital of Li, Rydberg unit
A(2,1).entries = 'Na';      //
A(2,2).entries = -0.3777;   // Energy of outermost atomic orbital of Na, Rydberg unit
A(3,1).entries = 'F';       //
A(3,2).entries = -1.2502;   // Energy of outermost atomic orbital of F, Rydberg unit
A(4,1).entries = 'Cl';      //
A(4,2).entries = -0.9067;   // Energy of outermost atomic orbital of Cl, Rydberg unit
cf = 13.6;  // Conversion factor for Rydberg to eV
printf("\n________________________________________");
printf("\nAtom                              Energy gap");
printf("\n%s%s                              %5.2f eV", A(2,1).entries, A(4,1).entries, (A(2,2).entries-A(4,2).entries)*cf);
printf("\n%s%s                               %5.2f eV", A(2,1).entries, A(3,1).entries, (A(2,2).entries-A(3,2).entries)*cf);
printf("\n%s%s                               %5.2f eV", A(1,1).entries, A(3,1).entries, (A(1,2).entries-A(3,2).entries)*cf);
printf("\n________________________________________");

```

```
// Result
// -------------------------------------
// Atom                          Energy gap
// NaCl                           7.19 eV
// NaF                            11.87 eV
// LiF                            11.51 eV
// -------------------------------------
```

Scilab code Exa 4.23.18 Solid radius and Fermi level quantities for Cu and Nb

```
// Scilab Code Ex4.18a: Page-144 (2006)
clc; clear;
// For Cu
rs_a0_ratio = 2.67;     // Ratio of solid radius to the lattice parameter
E_F = 50.1*(rs_a0_ratio)^(-2);  // Fermi level energy of Cu, eV
T_F = 58.2e+04*(rs_a0_ratio)^(-2);  // Fermi level temperature of Cu, K
V_F = 4.20e+08*(rs_a0_ratio)^(-1);  // Fermi level velocity of electron in Cu, cm/sec
K_F = 3.63e+08*(rs_a0_ratio)^(-1);
printf("\nFor Cu :");
printf("\n==========");
printf("\nE_F = %6.4f eV", E_F);
printf("\nT_F = %5.3e K", T_F);
printf("\nV_F = %7.5e cm/sec", V_F);
printf("\nK_F = %6.4e per cm", K_F);
rs_a0_ratio = 3.07;     // Ratio of solid radius to the lattice parameter
E_F = 50.1*(rs_a0_ratio)^(-2);  // Fermi level energy of Nb, eV
T_F = 58.2e+04*(rs_a0_ratio)^(-2);  // Fermi level temperature of Nb, K
```

```
V_F = 4.20e+08*(rs_a0_ratio)^(-1);  // Fermi level velocity of electron in Nb, cm/sec
K_F = 3.63e+08*(rs_a0_ratio)^(-1);
printf("\n\nFor Nb:");
printf("\n==========");
printf("\nE_F = %6.4f eV", E_F);
printf("\nT_F = %5.3e K", T_F);
printf("\nV_F = %6.4e cm/sec", V_F);
printf("\nK_F = %6.4e per cm", K_F);

// Result
// For Cu :
// ==========
// E_F = 7.0277 eV
// T_F = 8.164e+004 K
// V_F = 1.57303e+008 cm/sec
// K_F = 1.3596e+008 per cm
//
// For Nb:
// ==========
// E_F = 5.3157 eV
// T_F = 6.175e+004 K
// V_F = 1.3681e+008 cm/sec
// K_F = 1.1824e+008 per cm
```

Chapter 5

Band Theory of Solids

Scilab code Exa 5.1 Fermi energy of Na and K

```
// Scilab Code Ex5.1: Page-176 (2006)
clc; clear;
h = 6.626e-34;  // Planck's constant, Js
h_bar = h/(2*%pi);  // Reduced Planck's constant, Js
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
m = 9.1e-031;   // Mass of an electron, kg


// For Na
n_Na = 2.65e+28;    // electronic concentration of Na, per metre cube
k_F = (3*%pi^2*n_Na)^(1/3);    // Fermi wave vector, per cm
E_F = h_bar^2*k_F^2/(2*m*e);  // Fermi energy of Na, eV
printf("\nThe fermi energy of Na = %4.2f eV", E_F);
printf("\nThe band structure value of Na = %4.2f eV", 0.263*13.6);
// For K
n_K = 1.4e+28;  // electronic concentration of K, per metre cube
```

```
k_F = (3*%pi^2*n_K)^(1/3);    // Fermi wave vector, per cm
E_F = h_bar^2*k_F^2/(2*m*e);  // Fermi energy of K, eV
printf("\nThe fermi energy of K = %4.2f eV", E_F);
printf("\nThe band structure value of K = %4.2f eV", 0.164*13.6);
printf("\nThe agreement between the free electron and band theoretical values are fairly good both for Na and K");


// Result
// The fermi energy of Na = 3.25 eV
// The band structure value of Na = 3.58 eV
// The fermi energy of K = 2.12 eV
// The band structure value of K = 2.23 eV
// The agreement between the free electron and band theoretical values are fairly good both for Na and K
```

Scilab code Exa 5.3 Fermi momentum of Na

```
// Scilab Code Ex5.3: Page-177 (2006)
clc; clear;
n_Na = 2.65e+22;    // electronic concentration of Na, per cm cube
k_F = (3*%pi^2*n_Na)^(1/3);    // Fermi wave vector, per cm
printf("\nThe fermi momentum of Na = %4.2e per cm", k_F);

// Result
// The fermi momentum of Na = 9.22e+07 per cm
```

Scilab code Exa 5.5 Energy separation between adjacent energy levels

```
// Scilab Code Ex5.5: Page-177 (2006)
clc; clear;
h = 6.626e-34;  // Planck's constant, Js
h_bar = h/(2*%pi);  // Reduced Planck's constant, Js
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
m = 9.1e-031;   // Mass of an electron, kg
V = 1.0e-06; // Volume of unit cube of material, metre cube

// For Mg
E_F = 7.13*e;  // Fermi energy of Mg, J
s = 2*%pi^2/(e*V)*(h_bar^2/(2*m))^(3/2)*(E_F)^(-1/2);  // Energy separation between levels for Mg, eV
printf("\nThe energy separation between adjacent levels for Mg = %5.3e eV", s);

// For Cs
E_F = 1.58*e;  // Fermi energy of Cs, J
s = 2*%pi^2/(e*V)*(h_bar^2/(2*m))^(3/2)*(E_F)^(-1/2);  // Energy separation between levels for Cs, eV
printf("\nThe energy separation between adjacent levels for Cs = %5.3e eV", s);


// Result
// The energy separation between adjacent levels for Mg = 5.517e-23 eV
// The energy separation between adjacent levels for Cs = 1.172e-22 eV
```

Scilab code Exa 5.9 Coupling constant of superconducting lead

```
// Scilab Code Ex5.9: Page-180 (2006)
clc; clear;

gamma_expt = 7.0e-04;    // Experimental value of electronic specific heat, cal/mol/K-square
gamma_theory = 3.6e-04;    // Theoretical value of electronic specific heat, cal/mol/K-square
L = poly(0, 'L');
L = roots(gamma_expt - gamma_theory*(1 + L));
printf("\nThe electron-phonon coupling constant of superconductor = %3.1f", L);

// Result
// The electron-phonon coupling constant of superconductor = 0.9
```

Scilab code Exa 5.10 Electronic specific heat coefficient of superconductor

```
// Scilab Code Ex5.10: Page-181 (2006)
clc; clear;
N_Ef = 1.235;    // Density of states at fermi energy, electrons/atom-eV
N = 6.023e+23;   // Avogadro's number
k = 1.38e-23;    // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Charge on an electron, C
gama = %pi^2*k^2/3*(N_Ef*N/e);    // Electronic specific heat coefficient, J/g-atom-kelvin square

printf("\nThe electronic specific heat coefficient of superconductor = %5.3f mJ/g-atom-kelvin square", gama/1e-03);

```

```
// Result
// The electronic specific heat coefficient of superconductor = 2.913 mJ/g-atom-kelvin square
```

Scilab code Exa 5.11 Electron phonon coupling constant for metal

```
// Scilab Code Ex5.11: Page-181 (2006)
clc; clear;
gamma_expt = 4.84;    // Experimental value of electronic specific heat of metal, mJ/g-atom/K-square
gamma_theory = 2.991;    // Theoretical value of electronic specific heat of metal, mJ/g-atom/K-square
L = poly(0, 'L');
L = roots(gamma_expt - gamma_theory*(1 + L));
printf("\nThe electron-phonon coupling constant for metal = %5.3f", L);

// Result
// The electron-phonon coupling constant for metal = 0.618
```

Scilab code Exa 5.12 Pauli spin susceptibility of Mg

```
// Scilab Code Ex5.12: Page-181 (2006)
clc; clear;
mu_B = 9.24e-027; // Bohr's magneton, J/T
N_Ef = 0.826;    // Density of states at fermi energy, electrons/atom-eV
N = 6.023e+23;   // Avogadro's number
e = 1.6e-019;    // Energy equivalent of 1 eV, J
chi_Pauli = mu_B^2*N_Ef*N/e;
```

```
printf("\nPauli spin susceptibility of Mg = %5.2e cgs units", chi_Pauli/1e-03);

// Result
// Pauli spin susceptibility of Mg = 2.65e-07 cgs units
```

Chapter 6

Semiconductor Physics

Scilab code Exa 6.1 Crystal absorption wavelengths from energy gaps

```
// Scilab Code Ex6.1: Page-190 (2006)
clc; clear;
S = cell(4,2);      // Declare a 4X2 cell
// Enter material names
S(1,1).entries = 'Si'; S(2,1).entries = 'GaAs'; S(3,1).entries = 'GaP'; S(4,1).entries = 'ZnS';
// Enter energy band gap values
S(1,2).entries = 1.11; S(2,2).entries = 1.42; S(3,2).entries = 2.26; S(4,2).entries = 3.60;
h = 6.626e-034;     // Planck's constant, Js
c = 3e+08;      // Speed of light, m/s
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
printf("\n--------------------------------------------");
printf("\nMaterial      E_g (eV)     Critical Wavelength (micron)");
printf("\n--------------------------------------------");
for i = 1:1:4
```

```
    lambda = h*c/(S(i,2).entries*e);
    printf("\n%8s         %4.2f              %5.3f", S(i, 1).entries, S(i, 2).entries, lambda/1e-06);
end
printf("\n-----------------------------------------------------");

// Result
// -----------------------------------------------------
// Material       E_g (eV)     Critical Wavelength (micron)
// -----------------------------------------------------
//       Si          1.11               1.119
//     GaAs          1.42               0.875
//      GaP          2.26               0.550
//      ZnS          3.60               0.345
// -----------------------------------------------------
```

Scilab code Exa 6.2 Phonon energy to lift the electron from valence band to conduction band

```
// Scilab Code Ex6.2: Page-192 (2006)
clc; clear;
c = 3e+08;       // Speed of light, m/s
h = 6.626e-034;      // Planck's constant, Js
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
omega = 2e+014;       // Wave vector involved in phonon energy, rad per sec
```

```
f = omega/(2*%pi);    // Frequency of the wave, Hz
E = h*f/e;          // Phonon energy involved in Si to lift the electron, eV
printf("\nThe phonon energy involved in Si = %5.3f eV which is insufficient to lift an electron.", E);

// Result
// The phonon energy involved in Si = 0.132 eV which is insufficient to lift an electron.
```

Scilab code Exa 6.3 Densitites of Si and GaAs

```
// Scilab Code Ex6.3: Page-192 (2006)
clc; clear;
N_A = 6.023e+023;    // Avogadro's number
// For Si
A = 28.1;    // Atomic weight of Si, g/mol
a = 5.43e-08;    // Lattice constant for Si, cm
n = 8/a^3;   // Number of atoms per unit volume, atoms/cc
rho = n*A/N_A;   // Density of Si, g/cc
printf("\nThe density of Si = %4.2f atoms per cc", rho);
// For GaAs
A = 69.7+74.9;    // Atomic weight of GaAs, g/mol
a = 5.65e-08;    // Lattice constant for Si, cm
n = 4/a^3;   // Number of atoms per unit volume, atoms/cc
rho = n*A/N_A;   // Density of GaAs, g/cc
printf("\nThe density of GaAs = %5.3f atoms per cc", rho);

// Result
// The density of Si = 2.33 atoms per cc
// The density of GaAs = 5.324 atoms per cc
```

Scilab code Exa 6.4 Intrinsic carrier concentration of GaAs at 300 K

```
// Scilab Code Ex6.4: Page-196 (2006)
clc; clear;
m = 9.11e-031;   // Electron Rest Mass , kg
k = 1.38e-023;   // Boltzmann constant , J/mol/K
h = 6.626e-034;  // Planck's constant , Js
T = 300;         // Room temperature , K
m_e = 0.068*m;     // Mass of electron , kg
m_h = 0.56*m;    // Mass of hole , kg
E_g = 1.42*1.6e-019;      // Energy band gap for GaAs , J
n_i = 2*(2*%pi*k*T/h^2)^(3/2)*(m_e*m_h)^(3/4)*exp(-E_g/(2*k*T));
printf("\nThe Intrinsic carrier concentration of GaAs at 300 K = %1.0e per metre cube", n_i);

// Result
// The intrinsic carrier concentration of GaAs at 300 K = 3e+012 per metre cube
```

Scilab code Exa 6.5 Position of Fermi level of Si at room temperature

```
// Scilab Code Ex6.5: Page-197 (2006)
clc; clear;
m = 9.11e-031;   // Electron Rest Mass , kg
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
k = 1.38e-023;   // Boltzmann constant , J/mol/K
T = 300;         // Room temperature , K
m_e = 1.1*m;     // Mass of electron , kg
m_h = 0.56*m;    // Mass of hole , kg
```

```
E_g = 1.1;      // Energy band gap for GaAs, J
E_F = E_g/2+3/4*k*T/e*log(m_h/m_e);   // Position of Fermi level of Si at room temperature, eV
printf("\nThe position of Fermi level of Si at room temperature = %5.3f eV", E_F);
printf("\nThe fermi level in this case is shifted downward from the midpoint (0.55 eV) in the forbiddem gap.");

// Result
// The position of Fermi level of Si at room temperature = 0.537 eV
// The fermi level in this case is shifted downward from the midpoint (0.55 eV) in the forbiddem gap.
```

Scilab code Exa 6.6 Intrinsic resistivity of Ge at room temperature

```
// Scilab Code Ex6.6: Page-197 (2006)
clc; clear;
e = 1.6e-019;   // Electronic charge, C
n_i = 2.15e+013;    // Carrier density of Ge at room temperature, per cc
mu_e = 3900;    // Mobility of electron, cm-square/V-s
mu_h = 1900;    // Mobility of hole, cm-square/V-s
sigma_i = e*(mu_e+mu_h)*n_i;    // Intrinsic conductivity of Ge, mho per m
rho_i = 1/sigma_i;  // Intrinsic resistivity of Ge at room temperature, ohm-m
printf("\nThe intrinsic resistivity of Ge at room temperature = %2d ohm-cm", rho_i);


// Result
// The intrinsic resistivity of Ge at room
```

```
temperature = 50 ohm-cm
```

Scilab code Exa 6.7 Conductivity in CdS

```
// Scilab Code Ex6.7: Page-197 (2006)
clc; clear;
m = 9.1e-031;    // Mass of an electron , kg
e = 1.6e-019;    // Electronic charge , C
k = 1.38e-023;   // Boltzmann constant , J/mol/K
T = 30;     // Given temperature , K
n = 1e+22;     // Carrier density of CdS, per metre cube
mu = 1e-02;      // Mobility of electron , metre-square/V-s
sigma = e*mu*n;     // Conductivity of CdS, mho per m
printf("\nThe conductivity of CdS sample = %2d mho per m", ceil(sigma));
m_eff = 0.1*m;   // Effective mass of the charge carries , kg
t = m_eff*sigma/(n*e^2);   // Average time between successive collisions , s
printf("\nThe average time between successive collisions = %4.2e sec", t);
// We have 1/2*m_eff*v^2 = 3/2*k*T, solving for v
v = sqrt(3*k*T/m_eff);     // Velocity of charrge carriers , m/s
l = v*t;     // Mean free distance travelled by the carrier , m
printf("\nThe mean free distance travelled by the carrier = %4.2e m", l);

// Result
// The conductivity of CdS sample = 16 mho per m
// The average time between successive collisions = 5.69e-015 sec
```

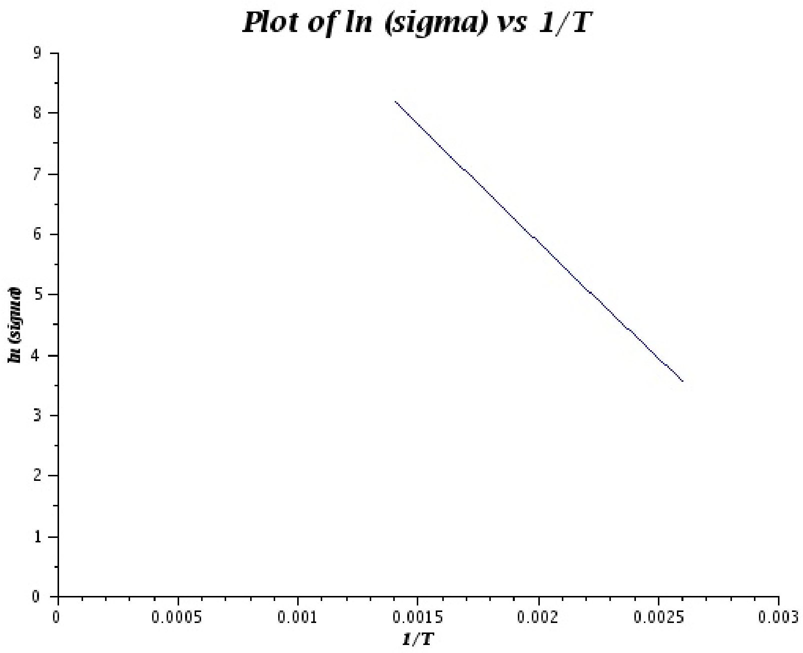

Figure 6.1: Energy Gap of Ge

```
// The mean free distance travelled by the carrier =
    6.64e-010 m
```

Scilab code Exa 6.8 Energy Gap of Ge

```
// Scilab Code Ex6.8: Page-199 (2006)
clc; clear;
k = 1.38e-023;   // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
T = [385 455 556 714];       // Temperatures of Ge, K
rho = [0.028 0.0061 0.0013 0.000274];    //
```

```
   Electrical resistivity, ohm-m
Tinv = zeros(4);    // Create an empty row matrix
   for 1/T
ln_sigma = zeros(4);  // Create the empty row matrix
    for log(sigma)
for i = 1:1:4
    Tinv(i) = 1/T(i);
    log_sigma(i) = log(1/rho(i));
end
// Plot the graph
plot(Tinv, log_sigma);
a=gca(); // Handle on axes entity
a.box="off";
a.x_location = "origin";
a.y_location = "origin";
a.x_label
a.y_label
a.title
type(a.title);
x_label=a.x_label;
x_label.text="1/T";
x_label.font_style= 5;
y_label=a.y_label;
y_label.text="ln (sigma)";
y_label.font_style= 5;
t=a.title;
t.foreground=9;
t.font_size=4;
t.font_style=5;
t.text="Plot of ln (sigma) vs 1/T";
// Calculate slope
slope = (log_sigma(2)-log_sigma(1))/(Tinv(2)-Tinv(1)
   );
E_g = abs(2*slope*k);        // Energy gap of Ge, J
printf("\nThe energy gap of Ge = %5.3f eV", E_g/e);

// Result
// The energy gap of Ge = 0.658 eV
```

Scilab code Exa 6.9 Energy gap and emission wavelength of Al doped GaAs

```
// Scilab Code Ex6.9: Page-199 (2006)
clc; clear;
h = 6.626e-34;  // Planck's constant, Js
c = 3e+08;      // Speed of light, m/s
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
x = 0.07;   // Al concentration in host GaAs
E_g = 1.424 + 1.266*x + 0.266*x^2;    // Band gap of GaAs as a function of x, eV
// As E_g = h*c/lambda, solving for lambda
lambda = h*c/(E_g*e);   // Emission wavelength of light, m
printf("\nThe energy band gap of Al doped GaAs = %4.2f eV", E_g);
printf("\nThe emission wavelength of light = %4.2f micron", lambda/1e-06);
printf("\nThe Al atoms go as substitutional impurity in the host material.");

// Result
// The energy band gap of Al doped GaAs = 1.51 eV
// The emission wavelength of light = 0.82 micron
// The Al atoms go as substitutional impurity in the host material.
```

Scilab code Exa 6.10 Energy gap of Al doped GaAs

```
// Scilab Code Ex6.10: Page-200 (2006)
clc; clear;
```

```
x = 0.38;    // Al concentration in host GaAs
E_g = 1.424 + 1.266*x + 0.266*x^2;     // Band gap of GaAs as a function of x, eV
printf("\nThe energy band gap of 38 percent Al doped in GaAs = %5.3f eV", E_g);

// Result
// The energy band gap of 38 percent Al doped in GaAs = 1.943 eV
```

Scilab code Exa 6.11 Resistivity of Ge at 20 degree celsius

```
// Scilab Code Ex6.11: Page-200 (2006)
clc; clear;
k = 1.38e-023;   // Boltzmann constant, J/mol/K
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
rho_40 = 0.2;    // Resistivity of Ge at 40 degree celsius, ohm-m
T1 = 40+273;     // Temperature at which resistivity of Ge becomes 0.2 ohm-m, K
T2 = 20+273;     // Temperature at which resistivity of Ge is to be calculated, K
E_g = 0.7;   // Band gap of Ge, eV
// As rho = exp(E_g/(2*k*T)), so for rho_20
rho_20 = rho_40*exp(E_g/(2*k/e)*(1/T2-1/T1));   // Resistivity of Ge at 20 degree celsius, ohm-m
printf("\nThe resistivity of Ge at 20 degree celsius = %3.1f ohm-m", rho_20);

// Result
// The resistivity of Ge at 20 degree celsius = 0.5 ohm-m
```

Scilab code Exa 6.12 Donor ionization energies at room temperature

```
// Scilab Code Ex6.12: Page-203 (2006)
clc; clear;
k = 1.38e-023;  // Boltzmann constant, J/mol/K
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
T = 300;        // Room temperature of the material, K
K_Si = 11.7;    // Dielectric constant of Si
K_Ge = 15.8;    // Dielectric constant of Ge
m = 9.1e-031;   // Mass of an electron, kg
m_eff = 0.2;    // Effective masses of the electron in both Si and Ge, kg
E_ion_Si = 13.6*m_eff/K_Si^2;   // Donor ionization energy of Si, eV
E_ion_Ge = 13.6*m_eff/K_Ge^2;   // Donor ionization energy of Ge, eV
E = k*T/e;      // Energy available for electrons at 300 K, eV
printf("\nThe donor ionization energy of Si = %6.4f eV", E_ion_Si);
printf("\nThe donor ionization energy of Ge = %6.4f eV", E_ion_Ge);
printf("\nThe energy available for electrons at 300 K = %5.3f eV", E);

// Result
// The donor ionization energy of Si = 0.0199 eV
// The donor ionization energy of Ge = 0.0109 eV
// The energy available for electrons at 300 K = 0.026 eV
```

Scilab code Exa 6.13 Radius of the orbit of the fifth valence electron of the acceptor impurity in Ge

```
// Scilab Code Ex6.13: Page-203 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
epsilon = 15.8;   // Dielectric constant of Ge
m = 9.1e-031;    // Mass of an electron , kg
m_e = 0.2*m;      // Effective masses of the electron in Ge, kg
a_Ge = 5.65;     // Lattice parameter of Ge, angstrom
A_d = 0.53*epsilon*(m/m_e);     // Radius of donor atom, angstrom
printf("\nThe radius of the orbits of fifth valence electron of acceptor impurity = %2d angstrom", ceil(A_d));
printf("\nThis radius is %d times the lattice constant of Ge", ceil(A_d/a_Ge));

// Result
// The radius of the orbits of fifth valence electron = 42 angstrom
// This radius is 8 times the lattice constant of Ge
```

Scilab code Exa 6.14 Mobility of electron and hole concentration in Ge

```
// Scilab Code Ex6.14: Page-203 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
tau = 1e-012;    // Life time of electron in Ge, s
m = 9.1e-031;    // Mass of an electron , kg
m_e = 0.5*m;      // Effective masses of the electron in Ge, kg
mu = e*tau/m_e;    // Mobility of electron in Ge, m-square/V-s
n_i = 2.5e+019;      // Intrinsic carrier concentration of Ge at room temperature , per metre cube
```

```
n_Ge = 5e+028;  // Concentration of Ge atoms, per metre cube
n_e = n_Ge/1e+06;     // Concentration of impurity atoms, per metre cube
// From law of mass action, n_e*n_h = n_i^2, solving for n_h
n_h = n_i^2/n_e;     // Concentration of holes, per metre cube

printf("\nThis mobility of electron in Ge = %4d cm-square/V-s", mu/1e-04);
printf("\nThis concentration of holes in Ge = %4.2e per metre cube", n_h);

// Result
// This mobility of electron in Ge = 3516 cm-square/V-s
// This concentration of holes in Ge = 1.25e+016 per metre cube
```

Scilab code Exa 6.15 Hole concentration in Ge at room temperature

```
// Scilab Code Ex6.15: Page-204 (2006)
clc; clear;
n_i = 2.5e+019;      // Intrinsic carrier concentration of Ge at room temperature, per metre cube
n_Ge = 5e+028;  // Concentration of Ge atoms, per metre cube
delta_d = 1e+06;     // Rate at which pentavalent impurity is doped in pure Ge, ppm
n_e = n_Ge/delta_d;     // Concentration of impurity atoms, per metre cube
// From law of mass action, n_e*n_h = n_i^2, solving for n_h
```

```
n_h = n_i^2/n_e;     // Concentration of holes, per metre cube

printf("\nThis concentration of holes in Ge = %4.2e per metre cube", n_h);

// Result
// This concentration of holes in Ge = 1.25e+016 per metre cube
```

Scilab code Exa 6.16 Hall effect in n type semiconductor

```
// Scilab Code Ex6.16: Page-205 (2006)
clc; clear;
e = 1.6e-019;    // Charge on an electron, C
mu = 1400e-04;   // Mobility of electron, metre-square per volt per sec
l = 300-06;     // Length of the n-type semiconductor, m
w = 100-06;     // Width of the n-type semiconductor, m
t = 20-06;      // Thickness of the n-type semiconductor, m
N_D = 4.5e+021; // Doping concentration of donor impurities, per metre-cube
V = 10;      // Biasing voltage for semiconductor, V
B_prep = 1; // Perpendicular magnetic field to which the semiconductor is subjected, tesla

// Part (a)
n = N_D;     // Electron concentration in semiconductor, per cc
R_H = -1/(n*e); // Hall Co-efficient, per C per metre cube

```

```
// Part (b)
rho = 1/(n*e*mu);        // Resistivity of semiconductor, ohm-m
R = rho*l/(w*t);     // Resistance of the semiconductor, ohm
I = V/R;     // Current through the semiconductor, A
V_H = R_H*I*B_prep/t;    // Hall voltage, V

// Part (c)
theta_H = atand(-mu*B_prep);     // Hall angle, degrees


printf("\nHall coefficient, R_H = %4.2e per C metre cube", R_H);
printf("\nHall voltage, V_H = %4.2f V", abs(V_H));
printf("\nHall angle, theta_H = %4.2f degree", theta_H);

// Result
// Hall coefficient, R_H = -1.39e-003 per C metre cube
// Hall voltage, V_H = 0.45 V
// Hall angle, theta_H = -7.97 degree
```

Chapter 8

Magnetism

Scilab code Exa 8.1 Spontaneous magnetization of iron

```
// Scilab code Ex8.1 Page:241 (2006)
clc; clear;
rho = 7.9e+03;    // Density of iron, kg per cubic meter
A = 56e-03;    // Atomic weight of iron, g/mol
N_A = 6.02e+023;    // Avogadro's number, atoms per mole
mu_B = 9.3e-024;    // Bohr magneton;    // Ampere meter square
n = rho*N_A/A;    // Total number of atoms per unit cell, per cubic meter
M = 2.2*n*mu_B;    // Spontaneous magnetization of iron, Ampere per meter
printf("\nSpontaneous magnetization of iron = %4.2e Ampere per meter", M);

// Result
// Spontaneous magnetization of iron = 1.74e+006 Ampere per meter
```

Scilab code Exa 8.2 Saturation magnetization of a ferromagnetic material

```
// Scilab code Ex8.2 Page:241 (2006)
clc; clear;
n = 3e+028;     // Spin density of electrons in a ferromagnetic material, per cubic meter
mu = 3e-023;     // spin magnetic moment of a ferromagnetic material, Square Ampere
M_s = n*mu;     // Saturation magnetization of a ferromagnetic material, Per Ampere
printf("\nSaturation magnetization of a ferromagnetic material = %1.0e ampere per meter", M_s);

// Result
// Saturation magnetization of a ferromagnetic material = 9e+005 ampere per meter
```

Scilab code Exa 8.3 Magnetic susceptibility of Lithium

```
// Scilab code Ex8.3 Page:241 (2006)
clc; clear;
h_bar = 6.58e-016;      // Planck's constant, eV.s
m = 0.511e+06;        // Mass of an electron, eV
e = 1.6e-012;        // Energy equivalent of 1 eV, erg/eV
c = 3.0e+010;        // Speed of light, cm/s
N = 4.7e+022;     // Free electron gas concentration of Lithium, per cubic cm
mu_B = 9.27e-021;     // Bohr magneton, Ampere cm-square
```

```
E_F = (h_bar*c)^2/(2*m)*(3*%pi^2*N)^(2/3);     // Fermi energy, eV
chi = 3*N*mu_B^2/(2*E_F*e);     // Magnetic susceptibility of Lithium, cgs units
printf("\nMagnetic susceptibility of Lithium = %2.0e cgs units", chi);

// Result
// Magnetic susceptibility of Lithium = 8e-007 cgs units
```

Scilab code Exa 8.4 Diamagnetic susceptibility of helium atom in ground state

```
// Scilab code Ex8.4 Page:241 (2006)
clc; clear;
a_B = 0.53e-08;     // Bohr radius, cm
N = 27e+023;     // Atomic density of He gas, per cubic cm
c = 3e+010;     // Speed of light, cm/sec
e = 1.6e-019;     // Charge of an electron, Coulomb
m = 9.1e-028;     // Mass of an electron, g
// As r_classic = e^2/(m*c^2), Classical radius of an electron
r_classic = 2.8e-013;    // Classical radius of the electron, cm
chi = -2*N*r_classic/6*a_B^2;     // Magnetic susceptibility of Helium, cgs units

printf("\nDiamagnetic susceptibility of helium atom in ground state = %3.1e emu", chi);

// Result
// Diamagnetic susceptibility of helium atom in ground state = -7.1e-006 emu
```

Scilab code Exa 8.5 Atomic radii of helium and copper from atomic susceptibilities

```
// Scilab code Ex8.5 Page:242 (2006)
clc; clear;
chiA_He = 1.9e-06;    // Atomic susceptibility of helium, cm cube per mole
chiA_Cu = 18e-06;    // Atomic susceptibility of Copper, cm cube per mole
Q_sp = 1.77e+07;    // Specific charge of an electron, emu
Ne = 9650;    // Charge of a gram ion, emu
Z_He = 2;    // Atomic number of helium atom
Z_Cu = 29;    // Atomic number of copper atom
R_He = sqrt(abs(-6*chiA_He/(Ne*Z_He*Q_sp)));    // Magnetic susceptibility of helium atom, cgs units
R_Cu = sqrt(abs(-6*chiA_Cu/(Ne*Z_Cu*Q_sp)));    // Magnetic susceptibility of copper atom, cgs units
printf(”\nAtomic radius of helium = %4.2e cm”, R_He);
printf(”\nAtomic radius of copper = %4.2e cm”, R_Cu);

// Result
// Atomic radius of helium = 5.78e-009 cm
// Atomic radius of copper = 4.67e-009 cm
```

Scilab code Exa 8.6 Atomic susceptibility of Ne atom

```
// Scilab code Ex8.6 Page:242 (2006)
clc; clear;
```

```
N = 6.039e+022;     // Atomic density of Neon gas, per cubic cm
// As r_classic = e^2/(m*c^2), Classical radius of an electron
r_classic = 2.8e-013;    // Classical radius of the electron, cm
Z = 10;     // Atomic number of helium atom
a0 = 0.53e-08;       // Bohr's radius, cm
n1 = 2, n2 = 2, n3 = 6;      // Occupation numbers for 1s, 2s and 2p states of Ne
r_sq_1s = 0.031;     // Expectation value for 1s state
r_sq_2s = 0.905;     // Expectation value for 2s state
r_sq_2p = 1.126;     // Expectation value for 2p state
mean_r_sq = n1*r_sq_1s + n2*r_sq_2s + n3*r_sq_2p;    // Mean square radius, cm-square
Chi_A = -1/6*N*Z*r_classic*mean_r_sq*a0^2;    // Magnetic susceptibility of helium atom, cgs units
printf("\nAtomic susceptibility of Ne atom = %6.4e emu/mole", Chi_A);

// Result
// Atomic susceptibility of Ne atom = -6.8302e-006 emu/mole
```

Scilab code Exa 8.7 Langevin approximation for paramagnetism

```
// Scilab code Ex8.7: Page:249 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
h = 6.626e-034; // Planck's constant, Js
h_cross = h/(2*%pi);     // Reduced Planck's constant, Js
```

```
m = 9.1e-031;    // Mass of an electron, kg
mu = e*h_cross/(2*m);     // Bohr magneton, J/T
mu_H = mu/e;     // Magnetic energy, eV
kT = 0.025;     // Energy associated with two degrees of freedom, eV
E_ratio = mu_H/kT;  // Exceptional terms in Langevin's function
printf("\nThe magnitude of mu*H/(k*T) = %3.1e", E_ratio);

// Result
// The magnitude of mu*H/(k*T) = 2.3e-003
```

Scilab code Exa 8.8 Paramagnetic susceptibility of Mg

```
// Scilab code Ex8.8 Page:249 (2006)
clc; clear;
mu = 5.78e-005;     // Bohr magneton, eV/T
NE_F = 0.826;     // Density of states at fermi level, electrons/atom-J
chi_Pauli = mu^2*NE_F/1e-004;     // Pauli diamagnetism, cgs units
chi_Core = -4.2e-06;     // Core diamagnetism, cgs units
chi_Landau = -1/3*chi_Pauli;     // Landau diamagnetism, cgs units
chi_Total = chi_Core+ chi_Pauli+chi_Landau;     // Paramagnetic susceptibility of Mg, cgs units

printf("\nThe paramagnetic susceptibility of Mg = %5.2e cgs units",chi_Total);

// Result
// The paramagnetic susceptibility of Mg = 1.42e-05 cgs units
```

Scilab code Exa 8.9 Pauli spin susceptibility and diamagnetic contribution in Aluminium

```
// Scilab code Ex8.9 Page:250 (2006)
clc; clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
mu = 9.29e-024;     // Bohr magneton, J/T
mu_0 = 1.26e-006;     // Permeability of free space,
   Sq. tesla cubic meter per joule
E_F= 11.63*e;     // Fermi energy, J
N = 6.02e+028;     // Atomic concentration, atoms per
    cubic meter
chi_Total = 2.2e-005;    // Paramagnetic
   susceptibility of Mg, S.I. units
chi_Pauli = 3*N*mu^2*mu_0/(2*E_F);     // Pauli
   diamagnetism, S.I. units
chi_dia = chi_Total - chi_Pauli;     // Diamagnetic
   contribution to magnetic susceptibility

printf("\nThe Pauli spin susceptibility of Al = %5.3
   e S.I. units", chi_Pauli);
printf("\nThe diamagnetic contribution to magnetic
   susceptibility of Al = %5.3e S.I. units", chi_dia
   );

// Result
// The Pauli spin susceptibility of Al = 5.277e-06 S
   .I. units
// The diamagnetic contribution to magnetic
   susceptibility of Al = 1.672e-05 S.I. units
```

Scilab code Exa 8.10 Pauli spin susceptibility for Na

```
// Scilab code Ex8.10 Page:250 (2006)
clc; clear;
a0 = 5.3;     // Bohr radius, nm
rs_a0_ratio = 3.93;      // Ratio of solid radius to the lattice parameter
chi_Pauli = 2.59/rs_a0_ratio;     // Pauli's spin susceptibility, cgs units

printf("\nThe Pauli spin susceptibility for Na in terms of free electron gas parameter = %4.2f", chi_Pauli);

// Result
// The Pauli spin susceptibility for Na in terms of free electron gas parameter = 0.66
```

Scilab code Exa 8.11 Effective magneton number of Mn ion

```
// Scilab code Ex8.11 Page:264 (2006)
clc; clear;
S = 2;  // Spin quantum number
J = 0;  // Total quantum number
L = 2;  // Orbital quantum number
g = 2;  // Lande splitting factor
printf("\nThe spectroscopic term value of Mn3+ ion = %d_D_%d", 2*S+1, J);
// For J = L - S
J = L - S;
mu_N = g*sqrt(J*(J+1)); // Effective magneton number
printf("\nThe effective magneton number for J = L - S is %d", mu_N);
// For J = S, L = 0 so that
L = 0;
J = L+S;
mu_N = g*sqrt(J*(J+1)); // Effective magneton number
```

```
printf("\nThe effective magneton number for J = S is %3.1f.\nIt is in agreement with the experimental value of 5.0.", mu_N);

// Result
// The spectroscopic term value of Mn3+ ion = 5_D_0
// The effective magneton number for J = L - S is 0
// The effective magneton number for J = S is 4.9.
// It is in agreement with the experimental value of 5.0.
```

Scilab code Exa 8.12 Magnetic moment of 3d electrons of Fe using Hunds rule

```
// Scilab code Ex8.12 Page:264 (2006)
clc; clear;
mu = 9.27e-024; // Bohr's magneton, J/T
N_up = 5;    // Number of electrons with spin up as per Hunds Rule
N_down = 1; // Number of electrons with spin down as per Hunds Rule
M = mu*(N_up-N_down);      // Net magnetic moment associated with six electrons in the 3d shell, J/T

printf("\nThe magnetic moment of 3d electrons of Fe using Hunds rule = %d Bohr magnetons", M/mu);

// Result
// The magnetic moment of 3d electrons of Fe using Hunds rule = 4 Bohr magnetons
```

Scilab code Exa 8.13 Magnetic moment of compounds using Hunds rule

```
// Scilab code Ex8.13 Page:264 (2006)
clc; clear;
C = cell(3,4);
// Enter compound names
C(1,1).entries = 'LaCrO3';
C(2,1).entries = 'LaMnO3';
C(3,1).entries = 'LaCoO3';
// Enter Magnetic moments from Hunds rule
C(1,2).entries = 3.0;
C(2,2).entries = 4.0;
C(3,2).entries = 5.0;
// Enter Magnetic moments from Band theory
C(1,3).entries = 2.82;
C(2,3).entries = 3.74;
C(3,3).entries = 4.16;
// Enter Magnetic moments from the Experiment
C(1,4).entries = 2.80;
C(2,4).entries = 3.90;
C(3,4).entries = 4.60;
printf("\n--------------------------------------------------");
printf("\nCompound  Magnetic moment per formula unit (in BM)  ");
printf("\n-----------------------------------------");
printf("\n          Hunds Rule      Band Theory     Experiment");
printf("\n--------------------------------------------------");
for i = 1:1:3
    printf("\n%s      %3.2f                %4.2f               %4.2f", C(i,1).entries, C(i,2).entries, C(i,3).entries, C(i,4).entries);
end
printf("\n--------------------------------------------------
");

// Result
// -------------------------------------------------------
// Compound  Magnetic moment per formula unit (in BM)
// -------------------------------------------
//            Hunds Rule       Band Theory       Experiment
// -------------------------------------------------------
// LaCrO3      3.00              2.82              2.80
// LaMnO3      4.00              3.74              3.90
// LaCoO3      5.00              4.16              4.60
// -------------------------------------------------------
```

Scilab code Exa 8.14 Magnetic structure of the solids from total energy

```
// Scilab code Ex8.14 Page:268 (2006)
clc; clear;
C = cell(4,4);
// Enter compound names
C(1,1).entries = 'LaTiO3';
C(2,1).entries = 'LaCrO3';
C(3,1).entries = 'LaFeO3';
C(4,1).entries = 'LaCoO3';
// Enter total energy difference w.r.t. ground state for Paramagnetics, mRyd
C(1,2).entries = 0.014;
```

```
C(2,2).entries = 158.3;
C(3,2).entries = 20.69;
C(4,2).entries = 0.000;
// Enter total energy difference w.r.t. ground state for Ferromagnetics, mRyd
C(1,3).entries = 0.034;
C(2,3).entries = 13.99;
C(3,3).entries = 0.006;
C(4,3).entries = 0.010;
// Enter total energy difference w.r.t. ground state for Antiferromagnetics, mRyd
C(1,4).entries = 0.000;
C(2,4).entries = 0.000;
C(3,4).entries = 0.000;
C(4,4).entries = 0.003;
printf("\n--------------------------------------------------");
printf("\nSolid      Total energy difference (mRyd) (w.r.t. ground state)");
printf("\n ------------------------------------------");
printf("\n             Paramagnetic     Ferromagnetic    Antiferromagnetic ");
printf("\n--------------------------------------------------");
for i = 1:1:4
    printf("\n%s      %10.3f       %10.3f       %10.3f     ", C(i,1).entries, C(i,2).entries, C(i,3).entries, C(i,4).entries);
end
printf("\n--------------------------------------------------");
printf("\nAll the solids given above crystallize in the antiferromagnetic state except that of LaCoO3
.");

// Result
// ---------------------------------------------------------
// Solid      Total energy difference (mRyd) (w.r.t. ground state)
// ------------------------------------------------
//               Paramagnetic    Ferromagnetic    Antiferromagnetic
// ---------------------------------------------------------
// LaTiO3           0.014           0.034            0.000
// LaCrO3         158.300          13.990            0.000
// LaFeO3          20.690           0.006            0.000
// LaCoO3           0.000           0.010            0.003
// ---------------------------------------------------------
// All the solids given above crystallize in the antiferromagnetic state except that of LaCoO3.
```

Chapter 9

Superconductivity

Scilab code Exa 9.1 Critical field required to destroy superconductivity

```
// Scilab code Ex9.1 Page:278 (2006)
clc; clear;
H_c0 = 0.0803;    // Critical field at absolute zero, Tesla
T_c = 7.19;    // Transition temperature of specimen lead, Kelvin
T = 5;    // Temperature at which destruction of superconductivity is to be found, Kelvin
H_c = H_c0*[1-(T/T_c)^2];    // Critical field required to destroy superconductivity, Tesla
printf("\nCritical field required to destroy superconductivity = %6.4f T", H_c);

// Result
// Critical field required to destroy superconductivity = 0.0415 T
```

Scilab code Exa 9.2 Limiting magnetic field of Nb to serve as superconductor

```
// Scilab Code Ex9.2 Page:278 (2006)
clc; clear;
H0 = 1970;    // Critical field at absolute zero, Oe
T_c = 9.25;    // Transition temperature of specimen Nb, Kelvin
T = 4;    // Temperature at which destruction of superconductivity is to be found, Kelvin
H_c = H0*[1-(T/T_c)^2];    // Limiting magnetic field , Oe
printf("\nLimiting magnetic field of Nb to serve as superconductor = %4d Oe", round(H_c));

// Result
// Limiting magnetic field of Nb to serve as superconductor = 1602 Oe
```

Scilab code Exa 9.3 Transition temperature of a specimen

```
// Scilab Code Ex9.3 Page:278 (2006)
clc;clear;
T_1 = 14;    // Temperature, K
T_2 = 13;    // Temperature, K
H_c1 = 1.4e+05;    // Critical field at T_1, K
H_c2 = 4.2e+05;    // Critical field at T_2, K//As H_c1/H_c2 = (T_c^2-T_1^2)/(T_c^2-T_2^2), solving for T_c
T_c = sqrt((H_c2/H_c1*T_1^2 - T_2^2)/2);  // The superconducting transition temperature of a specimen, K
printf("\nTransition temperature of a specimen = %5.2f K", T_c);

// Result
// Transition temperature of a specimen = 14.47 K
```

Scilab code Exa 9.4 Coherence length of aluminium

```
// Scilab Code Ex9.4 Page:280 (2006)
clc;clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
E_g = 3.4e-04;     // Energy gap of aluminium, eV
v_F = 2.02e+08;     // Fermi velocity of aluminium, cm/sec
h_bar = 1.05e-034;     // Planck's constant
L = h_bar*v_F/(2*E_g*e);     // Coherence Length of aluminium, cm

printf("\nThe coherence length of aluminium = %4.2e cm", L);

// Result
// The coherence length of aluminium = 1.95e-04 cm
```

Scilab code Exa 9.6 Wavelength of photon required to break a Cooper pair

```
// Scilab Code Ex9.6 Page:284 (2006)
clc; clear;
h = 6.6e-034;      // Planck's constant, Js
e = 1.6e-019;         // Energy eqivalent of 1 eV, eV/J
k = 0.86e-004;  // Boltzmann constant, eV/K
T_c = 0.56;      // Critical temperature for superconducting Zr, K
E_g = 3.52*k*T_c;     // Energy gap of aluminium, J
c = 3e+08;       // Speed of light, m/s
```

```
lambda = h*c/(E_g*e);        // Wavelength of photon required to break a Cooper pair, m

printf("\nThe wavelength of photon required to break a Cooper pair = %3.1e m", lambda);

// Result
// The wavelength of photon required to break a Cooper pair = 7.3e-03 m (Answer given in the textbook is wrong)
```

Scilab code Exa 9.7 London penetration depth in Pb

```
// Scilab Code Ex9.7 :Page:285 (2006)
clc; clear;
Lambda_0 = 390;     // Penetration depth at absolute zero, angstorm
T_c = 7;     // Transition temperature of Pb, K
T = 2;       // Givn temperature, K
Lambda = Lambda_0*[1-(T/T_c)^2]^(-1/2);     // London penetration depth in Pb at 2K, angstorm
printf("\nThe London penetration depth in Pb at 2K = %7.4f angstorm", Lambda);
printf("\nThe London penetration depth at T = T_c becomes %d", %inf);

// Result
// The London penetration depth in Pb at 2K = 406.9644 angstorm
// The London penetration depth at T = T_c becomes Inf
```

Scilab code Exa 9.8 Isotopic exponent in Isotopic effect of Hg

```
// Scilab Code Ex9.8: Page:286 (2006)
clc; clear;
M = [199.5 200.7 202.0 203.3];  // Isotopic mass of Hg, amu
T_c = [4.185 4.173 4.159 4.146];    // Critical temperature of Hg, kelvin
alpha = 0.5;    // Trial value of Isotopic exponent
// Accroding to isotopic effect , T_c = K*M^(-alpha), solving for K
K = T_c(1)/M(1)^(-alpha);  // Isoptopic coefficent
Tc = zeros(3);
for i = 2:1:4
    Tc(i-1) = K*M(i)^(-alpha);
    printf("\nTc(%d) = %5.3f", i, Tc(i-1));
end
if T_c(2)-Tc(1)<0.001 & T_c(3)-Tc(2)<0.001 & T_c(4)-Tc(3)<0.001 then
    printf("\nThe isotopic exponent in Isotopic effect of Hg = %3.1f", alpha);
end

// Result
// Tc(2) = 4.172
// Tc(3) = 4.159
// Tc(4) = 4.146
// The isotopic exponent in Isotopic effect of Hg = 0.5
```

Scilab code Exa 9.9 Transition temperature of isotope of Hg whose mass number is 199

```
// Scilab code Ex9.9 Page:286 (2006)
clc; clear;
M_1 = 202;    // Mass of first isotope of mercury, amu
```

```
M_2 = 199;    // Mass of second isotope of mercury, amu
T_c1 = 4.153;    // Transition temperature of first isotope of mercury, K
//As T_c1/T_c2 = (M_2/M_1)^1/2, solving for T_c2
T_c2 = sqrt(M_1/M_2)*T_c1;  //
printf("\nThe transition temperature of isotope of Hg whose mass number is %d = %5.3f K", M_2, T_c2);

// Result
// The transition temperature of isotope of Hg whose mass number is 199 = 4.184 K
```

Scilab code Exa 9.10 Constant of proportionality in Isotopc effect

```
// Scilab code Ex9.10 Page:287 (2006)
clc; clear;
alpha = 0.5;    // Isotopic exponent of Osmium
T_c = 0.655;    // Transition temperature of Osmium, K
M = 190.2;    // Mass of Osmium, amu
K = T_c*M^alpha;    // K is the constant of proportionality

printf("\nThe value of constant of proportionality = %4.2f ", K);

// Result
// The value of constant of proportionality = 9.03
```

Scilab code Exa 9.11 Transition temperature and energy gap of a material

```
// Scilab code Ex9.11 Page:298 (2006)
clc; clear;
k = 1.38e-023;    // Boltzmann constant, J/mol/K
e = 1.6e-019;        // Energy equivalent of 1 eV, eV/J
Theta_D = 96;     // Debye temperature, kelvin
N0 = 0.3678;     // Density of states at Fermi energy
V = 1;            // Volume of the material, metre cube
T_c = 1.14*Theta_D*exp(-1/(N0*V));     // Critical temperature of the material, K
Delta_0 = k*Theta_D/sinh(1/(N0*V)); // Energy gap at absolute zero, J
printf("\nThe transition temperature of a material = %4.2f K", T_c);
printf("\nThe energy gap of a material = %5.3e eV", Delta_0/e);

// Result
// The transition temperature of a material = 7.22 K
// The energy gap of a material = 1.097e-03 eV
```

Scilab code Exa 9.12 Transition temperature of a superconductor using McMillan formula

```
// Scilab code Ex9.12 Page:298 (2006)
clc; clear;
Theta_D = 350;     // Debye temperature, kelvin
Lambda = 0.828;     // Electron-phonon coupling constant
mu_prime = 0.1373;     // Reduced mass of a superconductor, amu
T_c = Theta_D/1.45*exp(-1.04*(1+Lambda)/(Lambda-mu_prime*(1+0.62*Lambda)));      // Transition temperature of a superconductor using McMillan
```

```
formula, K

printf("\nThe transition temperature of the superconductor using McMillan formula = %5.2f K", T_c);

// Result
// The transition temperature of the superconductor using McMillan formula = 11.26 K
```

Scilab code Exa 9.13 Superconducting transition temperature of a superconductor using mcMillan formula

```
// Scilab code Ex9.13 : Page:298 (2006)
clc; clear;
Theta_D = 350;    // Debye temperature, kelvin
Lambda = 0.641;    // Electron-phonon coupling constant
mu_prime = 0.143;    // Reduced mass of a superconductor, amu
T_c = Theta_D/1.45*exp(-1.04*(1+Lambda)/(Lambda-mu_prime*(1+0.62*Lambda)));    // Superconducting transition temperature of a superconductor using mcMillan's formula, K

printf("\nThe superconducting transition temperature of a superconductor using McMillan formula = %5.3f K", T_c);

// Result
// The superconducting transition temperature of a superconductor using McMillan formula = 5.043 K
```

Scilab code Exa 9.15 Superconducting transition temperature of a borocarbide superconductor

```
// Scilab code Ex9.15 Page:314 (2006)
clc; clear;
Theta_D = 490;    // Debye temperature, Kelvin
Lambda = 0.8;    // wavelength of a superconductor, angstorm
mu_prime = 0.13;    // Reduced mass of a superconductor, amu
T_c = Theta_D/1.45*exp(-1.04*(1+Lambda)/(Lambda-mu_prime*(1+0.62*Lambda)));
printf("\nThe superconducting transition temperature of a borocarbide superconductor = %4.1f K", T_c);

// Result
// The superconducting transition temperature of a borocarbide superconductor = 15.4 K
```

Scilab code Exa 9.16 Electron phonon coupling constant for a superconductor

```
// Scilab code Ex9.16 Page:314 (2006)
clc; clear;
T_c = 16.5;    // Transition temperature of a superconductor, K
Lambda = [0.7 0.8 0.9 1.0];    // Electron-phonon coupling constants at different Tc values
Theta_D = 503;    // Debye temperature, kelvin
mu_prime = 0.13;    // Reduced mass of a superconductor, amu
Tc = zeros(4);
printf("\n_____________________");
printf("\nLambda          Tc");
printf("\n______________________");
for i = 1:1:4
   Tc(i) = Theta_D/1.45*exp(-1.04*(1+Lambda(i))/(Lambda(i)-mu_prime*(1+0.62*Lambda(i))));
   if abs(Tc(i) - 16.5) < 1.0 then
       best_Lvalue = Lambda(i);
   end
   printf("\n%3.1f        %8.1f K", Lambda(i), Tc(i));
end
printf("\n______________________");

printf("\nThe best electron-phonon coupling constant should be slightly above %3.1f ", best_Lvalue);

// Result
// ----------------------
// Lambda           Tc
// ----------------------
// 0.7             11.1 K
// 0.8             15.8 K
// 0.9             20.4 K
// 1.0             24.9 K
// ----------------------
// The best electron-phonon coupling constant should be slightly above 0.8
```

Scilab code Exa 9.17 Debye temperature of a BCS superconductor

```
// Scilab code Ex9.17 Page:317 (2006)
clc; clear;
T_c = 39.4;     // Transition temperature of a superconductor, K
Lambda = 1;     // Electron-phonon coupling constant for a superconductor
```

```
mu_prime= 0.15;    // Reduced mass of a superconductor, amu
// As T_c = Theta_D/1.45*exp(-1.04*(1+Lambda)/(Lambda-mu_prime*(1+0.62*Lambda))), solving for Theta_D
Theta_D = T_c*1.45*exp(1.04*(1+Lambda)/(Lambda-mu_prime*(1+0.62*Lambda)));

printf("\nThe Debye temperature of a BCS superconductor = %3d K", Theta_D);

// Result
// The Debye temperature of a BCS superconductor = 891 K
```