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

Created by
Mohit Garg
B.Tech
Electronics Engineering
ABES Engineering College, Ghaziabad
College Teacher
None
Cross-Checked by
Lavitha Pereira

September 1, 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: Applied Physics

Author: P. K. Mittal

Publisher: I. K. International, New Delhi

Edition: 2

Year: 2006

ISBN: 978-81-89866-72-3

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 1

Bonding in Solids

Scilab code Exa 1.3 calculate potential energy

```
1 //chapter 1
2 //example 1.3
3 //calculate potential energy
4 //page 15
5 clear;
6 clc;
7 //given
8 r=2; //in angstrom(distance)
9 e=1.6E-19; // in C (charge of electron)
10 E_o= 8.85E-12;// absolute premittivity
11 //calculate
12 r=2*1E-10; // since r is in angstrom
13 V=-e^2/(4*%pi*E_o*r); // calculate potential
14 printf('\nThe potential energy is \tV=%3.3E J',V);
15 V=V/e; // changing to eV
16 printf('\nIn electron-Volt,\tV=%.2f eV',V);
17 // Note: the answer in the book is wrong due to
   calculation mistake
```

Scilab code Exa 1.4 calculate bond energy for NaCl

```
1 //chapter 1
2 //example 1.4
3 //calculate bond energy for NaCl
4 //page 15-16
5 clear;
6 clc;
7 //given
8 r0=0.236; //in nanometer(interionic distance)
9 e=1.6E-19; // in C (charge of electron)
10 E_o= 8.85E-12;// absolute premittivity
11 N=8; // Born constant
12 IE=5.14;// in eV (ionisation energy of sodium)
13 EA=3.65;// in eV (electron affinity of Chlorine)
14 pi=3.14; // value of pi used in the solution
15 //calculate
16 r0=r0*1E-9; // since r is in nanometer
17 PE=(e^2/(4*pi*E_o*r0))*(1-1/N); // calculate
   potential energy
18 PE=PE/e; //changing unit from J to eV
19 printf('\nThe potential energy is\tPE=%.2f eV',PE);
20 NE=IE-EA;// calculation of Net energy
21 printf('\nThe net energy is\tNE=%.2f eV',NE);
22 BE=PE-NE;// calculation of Bond Energy
23 printf('\nThe bond energy is\tBE=%.2f eV',BE);
24 // Note: (1)-In order to make the answer prcatically
    feasible and avoid the unusual answer, I have
   used r_0=0.236 nm instead of 236 nm. because
   using this value will give very much irrelevant
   answer.
25 //       (2) There is slight variation in the answer
    due to round off.
```

Scilab code Exa 1.5 calculate compressibility

```
1 //chapter 1
2 //example 1.5
3 //calculate compressibility
4 //page 16
5 clear;
6 clc;
7 //given
8 r_0=.41; //in mm(lattice constant)
9 e=1.6E-19; // in C (charge of electron)
10 E_o= 8.85E-12;// absolute premittivity
11 n=0.5; // repulsive exponent value
12 alpha=1.76; // Madelung constant
13 pi=3.14; // value of pi used in the solution
14 //calculate
15 r=.41*1E-3; // since r is in mm
16 Beta=72*pi*E_o*r^4/(alpha*e^2*(n-1)); // calculation
    compressibility
17 printf('\nThe compressibility  is\tBeta=%1.2E ',Beta
   );
18 // Note: the answer in the book is wrong due to
   calculation mistake
```

Scilab code Exa 1.6 calculate ionic cohesive energy and atomic cohesive energy

```
1 //chapter 1
2 //example 1.6
3 //calculate ionic cohesive energy and atomic
   cohesive energy
4 //page 16
5 clear;
6 clc;
7 //given
8 r_0=3.56; // in Angstrom
9 e=1.6E-19; // in C (charge of electron)
```

```
10 IE=3.89; //in eV (ionisation energy of Cs)
11 EA=-3.61; // in eV (electron affinity of Cl)
12 n=10.5; // Born constant
13 E_o= 8.85E-12;// absolute premittivity
14 alpha=1.763; // Madelung constant
15 pi=3.14; // value of pi used in the solution
16 //calculate
17 r_0=r_0*1E-10; // since r is in nanometer
18 U=-alpha*(e^2/(4*pi*E_o*r_0))*(1-1/n); // calculate
      potential energy
19 U=U/e; //changing unit from J to eV
20 printf('\nThe ionic cohesive energy is\t%.2f eV',U);
21 ACE=U+EA+IE; // calculation of atomic cohesive
      energy
22 printf('\nThe atomic cohesive energy is\t%.2f eV',
      ACE);
```

Scilab code Exa 1.7 calculate contribution per ions to the cohesive energy

```
1 //chapter 1
2 //example 1.7
3 //calculate contribution per ions to the cohesive
      energy
4 //page 17
5 clear;
6 clc;
7 //given
8 r_0=2.81; // in Angstrom
9 e=1.6E-19; // in C (charge of electron)
10 n=9; // Born constant
11 E_o= 8.85E-12;// absolute premittivity
12 alpha=1.748; // Madelung constant
13 pi=3.14; // value of pi used in the solution
14 //calculate
15 r_0=r_0*1E-10; // since r is in nanometer
```

```
16 V=-alpha*(e^2/(4*pi*E_o*r_0))*(1-1/n); // calculate
      potential energy
17 V=V/e; //changing unit from J to eV
18 printf('\nThe potential energy is\tV=%.2f eV',V);
19 V_1=V/2; // Since only half of the energy contribute
       per ion to the cohecive energy therfore
20 printf('\nThe energy contributing per ions to the
      cohesive energy is \t%.2f eV',V_1);
21  // Note: Answer in the book is wroong due to
       calculation mistake
```

Chapter 2

Crystal Structure

Scilab code Exa 2.1 calculate lattice constant

```
1 //chapter 2
2 //example 2.1
3 //calculate lattice constant
4 //page 40-41
5 clear;
6 clc;
7 //given
8 N=6.02E26; // in /Kg-molecule (Avogadro's number)
9 n=4; // number of molecules per unit cell ofr NaCl
10 M=58.5; // in Kg/Kg-molecule (molecular weight of
     NaCl)
11 p=2189; // in Kg/m^3 (density)
12 //calculate
13 a=nthroot((n*M/(N*p)),3);
14 printf('\nThe lattice constant is\ta=%1.2E m',a);
15 a=a*1E10; // changing unit to Angstrom
16 printf('\n\t\t\ta=%.2f Angstrom',a);
```

Scilab code Exa 2.2 calculate distance between two nearest Cu atoms

```
1 //chapter 2
2 //example 2.2
3 //calculate distance between two nearest Cu atoms
4 //page 41
5 clear;
6 clc;
7 //given
8 N=6.02E23; // in /gram-atom (Avogadro's number)
9 n=4; // number of atom per unit cell for fcc
     structure
10 M=63.5; //in gram/gram-atom (atomic weight of Cu)
11 p=8.96; // in g/cm^3 (density)
12 //calculate
13 a=nthroot((n*M/(N*p)),3);
14 printf('\nThe lattice constant is\ta=%1.2E cm',a);
15 a=a*1E8; // changing unit from cm to Angstrom
16 printf('\n\t\t\ta=%.2f Angstrom',a);
17 d=a/sqrt(2); // distance infcc lattice
18 printf('\nThe distance between two nearest Cu atoms
     is \td=%.2f Angstrom',d);
```

Scilab code Exa 2.3 calculate lattice constant

```
1 //chapter 2
2 //example 2.3
3 //calculate lattice constant
4 //page 41-42
5 clear;
6 clc;
7 //given
8 N=6.02E26; // in /Kg-atom (Avogadro's number)
9 n=2; // number of molecules per unit cell for bcc
     lattice
10 M=55.85; // in Kg/Kg-atom (atomic weight of Iron)
11 p=7860; // in Kg/m^3 (density)
```

```
12 //calculate
13 a=nthroot((n*M/(N*p)),3);
14 printf('\nThe lattice constant is\ta=%1.3E m',a);
15 a=a*1E10; // changing unit to Angstrom
16 printf('\n\t\t\ta=%.3f Angstrom',a);
```

Scilab code Exa 2.4 calculate lattice constant

```
1 //chapter 2
2 //example 2.4
3 //calculate lattice constant
4 //page 42
5 clear;
6 clc;
7 //given
8 N=6.02E26; // in /Kg-atom (Avogadro's number)
9 n=2; // number of molecules per unit cell for bcc
     lattice
10 M=6.94; // in Kg/Kg-atom (atomic weight of Iron)
11 p=530; // in Kg/m^3 (density)
12 //calculate
13 a=nthroot((n*M/(N*p)),3);
14 printf('\nThe lattice constant is\ta=%1.3E m',a);
15 a=a*1E10; // changing unit to Angstrom
16 printf('\n\t\t\ta=%.3f Angstrom',a);
```

Scilab code Exa 2.5 calculate distance between adjacent atoms in NaCl

```
1 //chapter 2
2 //example 2.5
3 //calculate distance between adjacent atoms in NaCl
4 //page 42-43
5 clear;
```

```
clc;
//given
N=6.02E23; // in /gram-molecule (Avogadro's number)
M=58.5; //in gram/gram-molecule (atomic weight of
   NaCl)
p=2.17; // in g/cm^3 (density)
//calculate
// since V=M/p
// (1/d)^-3=2N/V=2Np/M
// therefore d= (M/2Np)^-3
d=nthroot((M/(2*N*p)),3);
printf('\nThe distance between two adjacent atoms of
    NaCl is \td=%1.2E cm',d);
d=d*1E8; // changing unit from cm to Angstrom
printf('\n\t\t\t\t\t\td=%.2f Angstrom',d);
```

Scilab code Exa 2.6 calculate packing fraction and density

```
//chapter 2
//example 2.6
//calculate packing fraction and density
//page 43
clear;
clc;
//given
r_Na=0.98; // in Angstrom (radius of sodium ion)
r_Cl=1.81; // in Angstrom (radius of chloride ion)
M_Na=22.99; // in amu (atomic mass of sodium)
M_Cl=35.45; // in amu (atomic mass of chlorine)
//calculate
a=2*(r_Na+r_Cl); // lattice parameter
printf('\nLattice constant  is \ta=%.2f Angstrom',a)
   ;
//PF=volume of ions present in the unit cell/volume
   of unit cell
```

```
16 PF=((4*(4/3)*%pi)*r_Na^3+(4*(4/3)*%pi)*r_Cl^3)/a^3;
17 printf('\nPacking fraction is %.3f',PF);
18 //Density=mass of unit cell/volume of unit cell
19 p=4*(M_Na+M_Cl)*1.66E-27/(a*1E-10)^3;
20 printf('\nDensity is \tp=%.f Kg/m^3',p);
21 p=p*1E-3; //changing unit to gm/cm^-3
22 printf('\nDensity is \tp=%.2f g/cm^3',p);
```

Chapter 3

Planes in Crystals

Scilab code Exa 3.11 calculate bond energy for NaCl

```
1 //chapter 3
2 //example 3.11
3 //calculate interplanar spacing
4 //page 61
5 clear;
6 clc;
7 //given
8 h=3,k=2,l=1; // miller indices
9 a=4.2E-8; // in cm (lattice constant)
10 //calculate
11 d=a/sqrt(h^2+k^2+l^2); // calculation for
      interplanar spacing
12 printf('\nThe interplanar spacing is\td=%1.2E cm',d)
      ;
13 d=d*1E8; //changing unit from cm to Angstrom
14 printf('\n\t\t\td=%.2f Angstrom',d);
```

Scilab code Exa 3.12 calculate lattice spacing

```
1 //chapter 3
2 //example 3.12
3 //calculate lattice spacing
4 //page 61
5 clear;
6 clc;
7 //given
8 h=1,k=1,l=1; // miller indices
9 a=2.5,b=2.5,c=1.8; // in Angstrom (lattice constants
      for tetragonal lattice )
10 //calculate
11 d=1/sqrt((h/a)^2+(k/b)^2+(l/c)^2); // calculation
     for interplanar spacing
12 printf('\nThe lattice spacing is\td=%.2f Angstrom',d
     );
```

Scilab code Exa 3.15 calculate density

```
1 //chapter 3
2 //example 3.15
3 //calculate density
4 //page 63
5 clear;
6 clc;
7 //given
8 h=1,k=0,l=0; // miller indices
9 a=2.5; // in Angstrom (lattice constant)
10 //calculate
11 a=a*1E-10; //hence a is in Angstrom
12 d=a/sqrt(h^2+k^2+l^2); // calculation for
     interplanar spacing
13 p=d/a^3;
14 printf('\nThe density is\tp=%1.1E lattice points/m^2
     ',p);
```

Chapter 4

Crystal Diffraction

Scilab code Exa 4.1 Find spacing constant

```
1 //chapter 4
2 //example 4.1
3 //Find spacing constant
4 //page 75
5 clear;
6 clc;
7 //given
8 lambda=2.6; // in Angstrom (wavelength)
9 theta=20; // in Degree (angle)
10 n=2;
11 //calculate
12 lambda=lambda*1E-10; // since lambda is in Angstrom
13 // Since 2dsin(theta)=n(lambda)
14 // therefore d=n(lambda)/2sin(theta)
15 d=n*lambda/(2*sind(theta));
16 printf('\nThe spacing constant is \td=%1.1E m',d);
17 d=d*1E10; // changing unit from m to Angstrom
18 printf('\n\t\t\td=%.1f Angstrom',d);
```

Scilab code Exa 4.2 Find glancing angle

```
1 //chapter 4
2 //example 4.2
3 //Find glancing angle
4 //page 75
5 clear;
6 clc;
7 //given
8 h=1,k=1,l=0; //miller indices
9 a=0.26; // in nanometer (lattice constant)
10 lambda=0.065; // in nanometer (wavelength)
11 n=2; // order
12 //calculate
13 d=a/sqrt(h^2+k^2+l^2); // calculation of
     interlattice spacing
14 // Since 2dsin(theta)=n(lambda)
15 // therefore we have
16 theta=asind(n*lambda/(2*d));
17 printf('\nThe glancing angle is \t%.2f degree',theta
     );
18 //Note: there is slight variation in the answer due
     to round off
```

Scilab code Exa 4.3 Find glancing angle

```
1 //chapter 4
2 //example 4.3
3 //Find glancing angle
4 //page 75-76
5 clear;
6 clc;
7 //given
8 d=3.04E-10; // in mm (spacing constant)
9 lambda=0.79; // in Angstrom (wavelength)
```

```
10 n=3; // order
11 //calculate
12 // Since 2dsin(theta)=n(lambda)
13 // therefore we have
14 lambda=lambda*1E-10; //since lambda is in angstrom
15 theta=asind(n*lambda/(2*d));
16 printf('\nThe glancing angle is \t%.3f degree',theta
   );
17 //Note: In question the value of d=3.04E-9 cm but in
    solution is using d=3.04E-10 m.
18 // I have used d=3.04E-10 cm as used in the solution
```

Scilab code Exa 4.4 Find wavelength and maximum order possible

```
1 //chapter 4
2 //example 4.4
3 //Find wavelength and maximum order possible
4 //page 76
5 clear;
6 clc;
7 //given
8 d=0.282; // in nanometer (spacing constant)
9 n=1; // order
10 theta=8.35; // in degree (glancing angle)
11 //calculate
12 d=d*1E-9; // since d is in nanometer
13 // Since 2dsin(theta)=n(lambda)
14 // therefore we have
15 lambda=2*d*sind(theta)/n;
16 printf('\nThe wavelength is \t%1.2E m',lambda);
17 lambda_1=lambda*1E10; //changing unit from m to
   Angstrom
18 printf('\n\t\t\t=%.3f Angstrom',lambda_1);
19 theta_1=90; // in degree (for maximum order theta
   =90)
```

```
20 n_max=2*d*sind(theta_1)/lambda; // calculation of
      maximum order.
21 printf('\nThe maximum order possible is \tn=%.f',
      n_max);
22 //Note: In question value of theta=8 degree and 35
      minutes but solution uses theta=8.35 degree
23 // I am using theta=8.35 degree
```

Scilab code Exa 4.5 Find wavelength in XU

```
1 //chapter 4
2 //example 4.5
3 //Find wavelength in X.U.
4 //page 76-77
5 clear;
6 clc;
7 //given
8 theta=6; // in degree (glancing angle)
9 p=2170; // in Kg/m^3 (density)
10 M=58.46; // Molecular weight of NaCl
11 N=6.02E26; // in Kg-molecule (Avogadro's number)
12 n=1; // order
13 XU=1E-12; //since 1X.U.= 1E-12m
14 //calculate
15 d=(M/(2*N*p))^(1/3);//calclation of lattice constant
16 printf('\nThe spacing constant is \td=%1.3E m',d);
17 // Since 2dsin(theta)=n(lambda)
18 // therefore we have
19 lambda=2*d*sind(theta)/n; //calculation of
      wavelength
20 printf('\n\nThe wavelength is \t\t=%1.2E m',lambda);
21 lambda=lambda/XU;
22 printf('\n\t\t\t\t=%.1f X.U.',lambda);
23 // Note: The answer in the book is wrong due to
      calculation mistake
```

Scilab code Exa 4.6 find wavelength and energy

```
1 //chapter 4
2 //example 4.6
3 //find wavelength and energy
4 //page 77
5 clear;
6 clc;
7 //given
8 h=1,k=1,l=1; // miller indices
9 a=5.63; // in Angstrom (lattice constant)
10 theta=27.5; // in degree (Glancing angle)
11 n=1; //order
12 H=6.625E-34; // in J-s (Plank's constant)
13 c=3E8; // in m/s (velocity of light)
14 e=1.6E-19;// charge of electron
15 //calculate
16 d=a/sqrt(h^2+k^2+l^2); // calculation for
      interplanar spacing
17 printf('\nThe lattice spacing is\td=%.2f Angstrom',d
      );
18 // Since 2dsin(theta)=n(lambda)
19 // therefore we have
20 lambda=2*d*sind(theta)/n; // calculation for
      wavelength
21 printf('\nThe wavelength is\t=%.f Angstrom',lambda);
22 E=H*c/(lambda*1E-10); //calculation of Energy
23 printf('\nThe energy of X-rays is E=%1.3E J',E);
24 E=E/e; // changing unit from J to eV
25 printf('\n\t\tE=%1.3E eV',E);
26 // Note: c=3E8 m/s but in solution c=3E10 m/s has
      been used that's why answer is different
```

Scilab code Exa 4.7 calculate interpalanr spacing

```
1 //chapter 4
2 //example 4.7
3 //calculate interpalanr spacing
4 //page 77-78
5 clear;
6 clc;
7 //given
8 V=344; // in V (accelerating voltage)
9 theta=60; // in degree (glancing angle)
10 m=9.1E-31; // in Kg (mass of electron)
11 h=6.625e-34; // in J-s (Plank's constant)
12 n=1; //order
13 e=1.6E-19; // charge on electron
14 //calculate
15 //Since K=m*v^2/2=e*V
16 // therefore  v=sqrt(2*e*V/m)
17 // since lambda=h/(m*v)
18 //therefore we have lambda=h/sqrt(2*m*e*V)
19 lambda=h/sqrt(2*m*e*V); // calculation of lambda
20 printf('\nThe wavelength is \t\t =%1.2E m',lambda);
21 lambda=lambda*1E10; //changing unit from m to
     Angstrom
22 printf('\n\t\t\t\t =%.2f Angstrom',lambda);
23 // Since 2dsin(theta)=n(lambda)
24 // therefore we have
25 d=n*lambda/(2*sind(theta));
26 printf('\nThe interplanar spacing  is \t d=%.2f
     Angstrom',d);
```

Scilab code Exa 4.8 calculate angle of first order diffraction maximum

```
//chapter 4
//example 4.8
//calculate angle of first order diffraction maximum
//page 78-79
clear;
clc;
//given
K=0.02; // in eV (kinetic energy)
d=2.0; // in Angstrom (Bragg's spacing)
m=1.00898; // in amu (mass of neutron)
amu=1.66E-27; // in Kg (1amu=1.66E-27 Kg)
h=6.625e-34; // in J-s (Plank's constant)
n=1; //order
e=1.6E-19; // charge on electron
//calculate
//Since K=m*v^2/2
// therefore  v=sqrt(2*K/m)
// since lambda=h/(m*v)
//therefore we have lambda=h/sqrt(2*m*K)
m=m*amu; //changing unit from amu to Kg
K=K*e; //changing unit to J from eV
lambda=h/sqrt(2*m*K); // calculation of lambda
printf('\nThe wavelength is \t\t =%1.1E m',lambda);
lambda=lambda*1E10; //changing unit from m to
   Angstrom
printf('\n\t\t\t\t =%.1f Angstrom',lambda);
// Since 2dsin(theta)=n(lambda)
// therefore we have
theta=asind(n*lambda/(2*d)); // calculation of angle
    of first order diffraction maximum
printf('\nThe angle of first order diffraction
   maximum is %.f Degree',theta);
```

Scilab code Exa 4.9 Show that given angles are successive order of difraction and find spacing constant

```
1 //chapter 4
2 //example 4.9
3 //Show that given angles are successive order of
     difraction and find spacing constant
4 //page 79
5 clear;
6 clc;
7 //given
8 lambda=0.586; // in Angstrom (wavelength of X-rays)
9 n1=1, n2=2, n3=3; // orders of diffraction
10 theta1=5+(58/60); // in degree (Glancing angle for
     first order of diffraction)
11 theta2=12+(01/60); //in degree (Glancing angle for
     second order of diffraction)
12 theta3=18+(12/60); //in degree (Glancing angle for
     third order of diffraction)
13 //calculate
14 K1=sind(theta1);
15 K2=sind(theta2);
16 K3=sind(theta3);
17 printf('The value of sine of different angle of
     diffraction is\nK1=%.4f\nK2=%.4f\nK3=%.4f',K1,K2,
     K3);
18 // Taking the ratios of K1:K2:K3
19 // We get K1:K2:K3=1:2:3
20 //Therefore we have
21 printf('\n\nOr we have \tK1:K2:K3=1:2:3');
22 printf('\nHence these angles of incidence are for
     Ist, 2nd and 3rd order reflections respectively')
     ;
23 // Since 2dsin(theta)=n(lambda)
24 // therefore we have
25 d1=n1*lambda/(2*K1);
26 d2=n2*lambda/(2*K2);
27 d3=n3*lambda/(2*K3);
28 d1=d1*1E-10; //changing unit from Angstrom to m
29 d2=d2*1E-10; //changing unit from Angstrom to m
30 d3=d3*1E-10; //changing unit from Angstrom to m
```

```
31 printf('\n\nThe spacing constants are \nd1=%1.3E m\
      nd2=%1.3E m\nd3=%1.3E m',d1,d2,d3);
32 d=(d1+d2+d3)/3;
33 printf('\n\nThe mean value of crystal spacing is d=
      %1.3E m',d);
```

Chapter 5

Principles of Quantum Mechanics

Scilab code Exa 5.1 Find velocity and kinetic energy

```
1 //chapter 5
2 //example 5.1
3 //Find velocity and kinetic energy
4 //page 102-103
5 clear;
6 clc;
7 //given
8 lambda=1; //in Angstrom (wavelength)
9 m=1.67E-27; // in Kg (mass of neutron)
10 h=6.625E-34; // in J-s (Planck's constant)
11 e=1.6E-19; // in C (charge of electron)
12 //calculate
13 lambda=lambda*1E-10; //since lambda is in Angstrom
14 // Since lambda=h/(m*v)
15 // Therefore we have
16 v=h/(m*lambda); //calculation of velocity
17 printf('\nThe velocity is \t v=%1.2E m/s',v);
18 K=m*v^2/2; //calculation of kinetic energy
19 printf('\nThe kinetic energy is\tK=%1.2E J',K);
```

```
20 K=K/e; //changing unit fro J to eV
21 printf('\n\t\t\t=%.4f eV',K);
22 //Note: Due to round off, there is slight variation
   in the answer
```

Scilab code Exa 5.2 Calculate deBroglie wavelength

```
1 //chapter 5
2 //example 5.2
3 //Calculate de-Broglie wavelength
4 //page 103-104
5 clear;
6 clc;
7 //given
8 K=50; // in eV (Kinetic energy)
9 m0=9.1E-31; // in Kg (mass of electron)
10 h=6.625E-34; // in J-s (Planck's constant)
11 e=1.6E-19; // in C (charge of electron)
12 //calculate
13 K=K*e; //changing unit from eV to J
14 //Since K=m*v^2/2
15 // Therefore v=sqrt(2*K/m)
16 // Since lambda=h/(m*v)
17 // Therefore we have
18 lambda=h/sqrt(2*m0*K); //calculation of wavelength
19 printf('\nThe wavelength is\t=%1.3E m',lambda);
20 lambda=lambda*1E10; //changing unit from m to
   Angstrom
21 printf('\n\t\t\t=%.3f Angstrom',lambda);
```

Scilab code Exa 5.3 Calculate wavelength

```
1 //chapter 5
```

```
2 //example 5.3
3 //Calculate  wavelength
4 //page 104
5 clear;
6 clc;
7 //given
8 E=2000; // in eV (Kinetic energy)
9 m=9.1E-31; // in Kg (mass of electron)
10 h=6.625E-34; // in J-s (Planck's constant)
11 e=1.6E-19; // in C (charge of electron)
12 //calculate
13 E=E*e; //changing unit from eV to J
14 //Since E=m*v^2/2
15 // Therefore v=sqrt(2*E/m)
16 // Since lambda=h/(m*v)
17 // Therefore we have
18 lambda=h/sqrt(2*m*E); //calculation of wavelength
19 printf('\nThe wavelength is\t=%1.3E m',lambda);
20 lambda=lambda*1E9; //changing unit from m to
     nanometer
21 printf('\n\t\t\t=%.4f nm',lambda);
```

Scilab code Exa 5.4 Calculate deBroglie wavelength

```
1 //chapter 5
2 //example 5.3
3 //Calculate de-Broglie wavelength
4 //page 104
5 clear;
6 clc;
7 //given
8 m_e=9.1E-31; // in Kg (mass of electron)
9 m_n=1.676E-27; // in Kg (mass of neutron)
10 h=6.625E-34; // in J-s (Planck's constant)
11 c=3E8; // in m/s (velocity of light)
```

```
12 //calculate
13 E_e=m_e*c^2; // rest mass energy of electron
14 E_n=2*E_e; // given (kinetic energy of neutron)
15 //Since K=m*v^2/2
16 // Therefore v=sqrt(2*K/m)
17 // Since lambda=h/(m*v)
18 // Therefore we have
19 lambda=h/sqrt(2*m_n*E_n); //calculation of
      wavelength
20 printf('\nThe wavelength is\t=%1.1E m',lambda);
21 lambda=lambda*1E10; //changing unit from m to
      Angstrom
22 printf('\n\t\t\t=%1.1E Angstrom',lambda);
```

Scilab code Exa 5.5 Calculate wavelength

```
1 //chapter 5
2 //example 5.4
3 //Calculate wavelength
4 //page 104
5 clear;
6 clc;
7 //given
8 V=1600; // in V (Potential)
9 //calculate
10 lambda=12.27/sqrt(V); // calculation of wavelength
      in Angstrom
11 printf('\nThe wavelength is\t=%.3f Angstrom',lambda)
      ;
12 // Note: The answer in the book is wrong due to
      calculation mistake
```

Scilab code Exa 5.6 Calculate wavelength for photon and electron

```
1  //chapter 5
2  //example 5.6
3  //Calculate wavelength for photon and electron
4  //page 105
5  clear;
6  clc;
7  //given
8  E=1000; // in eV (Kinetic energy of photon)
9  K=1000; // in eV (Kinetic energy of electron)
10 m0=9.1E-31; // in Kg (mass of electron)
11 h=6.6E-34; // in J-s (Planck's constant)
12 c=3E8; // in m/s (velocity of light)
13 e=1.6E-19; // in C (charge on electron)
14 //calculate
15 E=E*e; // changing unit from eV to J
16 lambda_p=h*c/E; // For photon E=hc/lambda
17 printf('\nFor photon,the wavelength is\t=%1.2E m',
      lambda_p);
18 lambda_p=lambda_p*1E10; //changing unit from m to
      Angstrom
19 printf('\n\t\t\t=%.1f Angstrom',lambda_p)
20 //Since K=m*v^2/2
21 // Therefore v=sqrt(2*K/m)
22 // Since lambda=h/(m*v)
23 // Therefore we have
24 K=K*e; // changing unit from eV to J
25 lambda_e=h/sqrt(2*m0*K); //calculation of wavelength
26 printf('\nFor electron,the wavelength is\t=%1.1E m',
      lambda_e);
27 lambda_e=lambda_e*1E10; //changing unit from m to
      Angstrom
28 printf('\n\t\t\t=%.2f Angstrom',lambda_e);
29 // Note: The answer in the book is wrong because K
      =1.6E-16 J but the solution is using K=2.4*E-15 J
```

Scilab code Exa 5.7 Calculate velocity and kinetic energy

```
1 //chapter 5
2 //example 5.7
3 //Calculate velocity and kinetic energy
4 //page 105
5 clear;
6 clc;
7 //given
8 lambda=1.66E-10; // in m (wavelength)
9 m=9.1E-31; // in Kg (mass of electron)
10 h=6.626E-34; // in J-s (Planck's constant)
11 e=1.6E-19; // in C (charge on electron)
12 //calculate
13 // Since lambda=h/(m*v)
14 // Therefore we have
15 v=h/(m*lambda); // calculation of velocity
16 printf('\nThe velocity of electron is \tv=%1.3E m/s'
   ,v);
17 K=m*v^2/2;//calculation of kinetic energy
18 printf('\nThe kinetic energy is \tK=%1.4E J',K);
19 K=K/e; // changing unit from J to eV
20 printf('\n\t\t\t=%.3f eV',K);
21 // Note: The answer in the book for kinetic energy
   is wrong due to calculation mistake
```

Scilab code Exa 5.8 Calculate deBroglie wavelength

```
1 //chapter 5
2 //example 5.8
3 //Calculate de-Broglie wavelength
4 //page 106
5 clear;
6 clc;
7 //given
```

```
T=400; // in K (temperature)
m=6.7E-27; // in Kg (mass of He-atom)
h=6.6E-34; // in J-s (Planck's constant)
k=1.376E-23; // in J/degree (Boltzmann constant)
//calculate
// Since lambda=h/(m*v)
// E=mv^2/2;
// Therefore lambda=h/sqrt(2*m*E)
//E=kT
//Therefore lambda=h/sqrt(2*m*k*T)
lambda=h/sqrt(2*m*k*T)
printf('\nThe de-Broglie wavelength of He-atom is \t
   =%1.4E m',lambda);
lambda=lambda*1E10; //changing unit from m to
   Angstrom
printf('\n\t\t\t\t\t=%.4f Angstrom',lambda);
```

Scilab code Exa 5.9 Calculate deBroglie wavelength of proton

```
//chapter 5
//example 5.9
//Calculate de-Broglie wavelength of proton
//page 106
clear;
clc;
//given
m_e=9.1E-31; // in Kg (mass of electron)
m_p=1.6E-27; // in Kg (mass of proton)
h=6.626E-34; // in J-s (Planck's constant)
c=3E8; // in m/s (velocity of light)
//calculate
E=m_e*c^2; // in J (rest energy of electron)
// Since lambda=h/(m*v)
// E=mv^2/2;
// Therefore lambda=h/sqrt(2*m*E)
```

```
17 // Also E=m_e*c^2;
18 // therefore lambda=h/sqrt(2*m_p*m_e*c^2)
19 lambda=h/sqrt(2*m_p*m_e*c^2); // calculation of
     wavelength
20 printf('\nThe de-Broglie wavelength of proton is \t=
     %1.4E m',lambda);
21 lambda=lambda*1E10; //changing unit from m to
     Angstrom
22 printf('\n\t\t\t\t=%1.4E Angstrom',lambda);
```

Scilab code Exa 5.10 Calculate wavelength

```
1 //chapter 5
2 //example 5.10
3 //Calculate wavelength
4 //page 106
5 clear;
6 clc;
7 //given
8 V=10000; // in V (Potential)
9 //calculate
10 lambda=12.27/sqrt(V); // calculation of wavelength
     in Angstrom
11 printf('\nThe wavelength is\t=%.3f Angstrom',lambda)
     ;
```

Scilab code Exa 5.11 Calculate glancing angle

```
1 //chapter 5
2 //example 5.11
3 //Calculate glancing angle
4 //page 107
5 clear;
```

```
clc;
//given
V=100; // in V (Potential)
n=1; // order of diffraction
d=2.15; // in Angstrom (lattice spacing)
//calculate
lambda=12.27/sqrt(V); // calculation of wavelength
   in Angstrom
printf('\nThe wavelength is\t=%.3f Angstrom',lambda)
   ;
// Since 2*d*sind(theta)=n*lambda
//therefore we have
theta=asind(n*lambda/(2*d)); // calculation of
   glancing angle
printf('\nThe glancing angle is\t=%.1f degree',theta
   );
// Note: In question V=100 eV but the solution is
   using V=100V in the book and I have also used V
   =100V
```

Scilab code Exa 5.12 Calculate spacing of crystal

```
//chapter 5
//example 5.12
//Calculate spacing of crystal
//page 107
clear;
clc;
//given
V=344; // in V (Potential)
n=1; // order of diffraction
theta=60; // in degree (glancing angle)
//calculate
lambda=12.27/sqrt(V); // calculation of wavelength
   in Angstrom
```

```
13 printf('\nThe wavelength is\t\t=%.3f Angstrom',
      lambda);
14 // Since 2*d*sind(theta)=n*lambda
15 //therefore we have
16 d=n*lambda/(2*sind(theta)); // calculation of
      spacing constant
17 printf('\nThe spacing of the crystal is\td=%.4f
      Angstrom',d);
```

Scilab code Exa 5.13 Calculate velocity of electron

```
1 //chapter 5
2 //example 5.13
3 //Calculate velocity of electron
4 //page 107-108
5 clear;
6 clc;
7 //given
8 r=0.53E-10; // in m (radius of first Bohr orbit)
9 h=6.6E-34; // in J-s (Planck's constant)
10 m=9.1E-31; // in Kg (mass of electron)
11 n=1; // First Bohr orbit
12 pi=3.14; // value of pi used in the solution
13 //calculate
14 // Since 2*pi*r=n*lambda and lambda=h/(m*v)
15 //Threfore we have v=h*n/(2*pi*r*m)
16 v=h*n/(2*pi*r*m); //calculation of velocity
17 printf('\nThe velocity of electron is\tv=%1.1E m/s',
      v);
```

Scilab code Exa 5.14 Calculate uncertainty in the momentum and uncertainty in the velocity

```
1 //chapter 5
2 //example 5.14
3 //Calculate uncertainty in the momentum and
     uncertainty in the velocity
4 //page 108
5 clear;
6 clc;
7 //given
8 dx=0.2; // in Angstrom (uncertainty in the position)
9 h=6.6E-34; // in J-s (Planck's constant)
10 m0=9.1E-31; // in Kg (mass of electron)
11 pi=3.14; // value of pi used in the solution
12 //calculate
13 dx=dx*1E-10; //since dx is in Angstrom
14 // Since dx*dp=h/4*pi  (uncertainty relation)
15 dp=h/(4*pi*dx); // calculation of uncertainty in the
     momentum
16 printf('\nThe uncertainty in the momentum is\tdp=%1
     .2E Kg-m/s',dp);
17 //since dp=m*dv
18 dv=dp/m0; // calculation of uncertainty in the
     velocity
19 printf('\nThe uncertainty in the velocity is\tdv=%1
     .2E m/s',dv);
```

Scilab code Exa 5.15 Compare uncertainty in the velocity of electron and proton

```
1 //chapter 5
2 //example 5.15
3 //Compare uncertainty in the velocity of electron
     and proton
4 //page 108
5 clear;
6 clc;
```

```
7 //given
8 m_e=9.1E-31; // in Kg (mass of electron)
9 m_p=1.67E-27; // in Kg (mass of proton)
10 dx_p=1; // in nanometer (uncertainty in position of
   electron)
11 dx_n=1; // in nanometer (uncertainty in position of
   proton)
12 //calculate
13 // since dp=h/(4*pi*dx)
14 // since h/(4*pi) is constant and dx is same for
   electron and proton
15 // therefor both electron and proton have same
   uncertainty in the momentum
16 // since dv=dp/m and dp is same for both
17 // therefore dv_e/dv_p=m_p/m_e
18 // therefore
19 K=m_p/m_e; // ratio of uncertainty in the velocity
   of electron and proton
20 printf('\nThe ratio of uncertainty in the velocity
   of electron to that of proton is\t=%.f',K);
```

Scilab code Exa 5.16 Calculate minimum uncertainty in the momentum and minimum kinetic energy of proton

```
1 //chapter 5
2 //example 5.16
3 //Calculate minimum uncertainty in the momentum and
   minimum kinetic energy of proton
4 //page 108-109
5 clear;
6 clc;
7 //given
8 dx=5E-15; // in m (radius of nucleus or uncertainty
   in the position)
9 h=6.6E-34; // in J-s (Planck's constant)
```

```
10 m=1.67E-27; // in Kg (mass of proton)
11 pi=3.14; // value of pi used in the solution
12 e=1.6E-19; // in C (charge of electron)
13 //calculate
14 // Since dx*dp=h/4*pi (uncertainty relation)
15 dp=h/(4*pi*dx); // calculation of uncertainty in the
       momentum
16 printf('\nThe minimum uncertainty in the momentum of
       proton is\tdp=%1.2E Kg-m/s',dp);
17 p=dp; // minimum value of momentum to calculate
     mimimum kinetic energy
18 K=p^2/(2*m); // calculation of minimum kinetic
     energy of proton
19 printf('\nThe minimum kinetic energy of proton is\tK
     =%1.1E J',K);
20 K=K/e; //changing unit from J to eV
21 printf('\n\t\t\t\t\t=%1.1E eV',K);
22 K=K/1E6; // changing unit from eV to MeV
23 printf('\n\t\t\t\t\t=%.1f MeV',K);
```

Scilab code Exa 5.17 Calculate percentage of uncertainty in the momentum of electron

```
1 //chapter 5
2 //example 5.17
3 //Calculate percentage of uncertainty in the
     momentum of electron
4 //page 109
5 clear;
6 clc;
7 //given
8 K=1; // in KeV (kinetic energy of electron)
9 dx=1; // in Angstrom (uncertainty in the position)
10 h=6.63E-34; // in J-s (Planck's constant)
11 m=9.1E-31; // in Kg (mass of electron)
```

```
12 pi=3.14; // value of pi used in the solution
13 e=1.6E-19; // in C (charge of electron)
14 //calculate
15 dx=dx*1E-10; // since dx is in Angstrom
16 // Since dx*dp=h/4*pi  (uncertainty relation)
17 dp=h/(4*pi*dx); // calculation of uncertainty in the
       momentum
18 printf('\nThe uncertainty in the momentum of
     electron is\tdp=%1.2E Kg-m/s',dp);
19 K=K*1E3*1.6E-19; // changing unit from KeV to J
20 p=sqrt(2*m*K); // calculation of momentum
21 printf('\nThe momentum of electron is\t\t\t p=%1.2E
     Kg-m/s',p);
22 poc=(dp/p)*100; // calculation of percentage of
     uncertainty
23 printf('\nThe percentage of uncertainty in the
     momentum  is =%.1f',poc);
```

Scilab code Exa 5.18 Calculate uncertainty in the position of electron

```
1 //chapter 5
2 //example 5.18
3 //Calculate uncertainty in the position of electron
4 //page 109-110
5 clear;
6 clc;
7 //given
8 v=6.6E4; // m/s (speed of electron)
9 poc=0.01; // percentage of uncertainty
10 h=6.63E-34; // in J-s (Planck's constant)
11 m=9E-31; // in Kg (mass of electron)
12 pi=3.14; // value of pi used in the solution
13 //calculate
14 p=m*v; // calculation of momentum
15 printf('\nThe momentum of electron is \t\t\tp=%1.2E
```

```
    Kg-m/s',p);
16 dp=(poc/100)*p; // calculation of uncertainty in the
       momentum
17 printf('\nThe uncertainty in the momentum of
    electron is\tdp=%1.2E Kg-m/s',dp);
18 // Since dx*dp=h/4*pi  (uncertainty relation)
19 dx=h/(4*pi*dp); // calculation of uncertainty in the
       position
20 printf('\nThe uncertainty in the position of
    electron is\tdx=%1.2E Kg-m/s',dx);
21 // Note; solution is incomplete in the book
```

Scilab code Exa 5.19 Calculate uncertainty in the position of X ray photon

```
1  //chapter 5
2  //example 5.19
3  //Calculate uncertainty in the position of X-ray
       photon
4  //page 111-112
5  clear;
6  clc;
7  //given
8  lambda=1; // in Angstrom (wavelength)
9  pi=3.14; // value of pi used in the solution
10 dlambda=1E-6; // uncertainty in wavelength
11 //calculate
12 lambda=lambda*1E-10; // sinc lambda is in Angstrom
13 // By uncertainty principle, dx*dp>=h/(4*pi)  --(1)
14 // since p=h/lambda  -----(2)
15 // Or p*lambda=h
16 // diffrentiting this equation
17 // p*dlambda+lambda*dp=0
18 // dp=-p*dlambda/lambda      ----(3)
19 //from (2) and (3)  dp=-h*dlambda/lambda^2    ---(4)
```

```
20 // from (1) and(4)  dx*dlambda>=lambda^2/4*pi
21 // Or dx=lambda^2/(4*pi*dlambda)
22 dx=lambda^2/(4*pi*dlambda); //calculation of
      uncertainty in the position
23 printf('\nThe uncertainty in the position of X-ray
      photon is \tdx=%1.0E m',dx);
24 // Note: 1. In the question, wavelength accuracy is
      given as 1 in 1E8 but in book solution has used 1
       in 1E6 and same has been used by me.
25 //       2. ANSWEER IS WRONG DUE TO CALCULATION
      MISTAKE
```

Scilab code Exa 5.20 Compare minimum uncertainty in the frequency of the photon

```
1 //chapter 5
2 //example 5.20
3 //Compare minimum uncertainty in the frequency of
      the photon
4 //page 111
5 clear;
6 clc;
7 //given
8 dt=1E-8; // in sec (average life time)
9 pi=3.14; // value of pi used in the solution
10 //calculate
11 // Since dE*dt>=h/(4*pi)  (uncertainty relation for
      energy)
12 // and E=h*v      v is the frequency
13 // therefore we have dv>=1/(4*pi*dt)
14 dv=1/(4*pi*dt); // calculation of minimum
      uncertainty in the frequency
15 printf('\nThe minimum uncertainty in the frequency
      of the photon is \tdv=%1.1E sec^-1',dv);
```

Scilab code Exa 5.21 Calculate uncertainty in the energy of the photon

```
1 //chapter 5
2 //example 5.21
3 //Calculate uncertainty in the energy of the photon
4 //page 111
5 clear;
6 clc;
7 //given
8 dt=1E-12; // in sec (average life time)
9 h=6.63E-34; // in J-s (Planck'c constant)
10 pi=3.14; // value of pi used in the solution
11 e=1.6*1E-19; // in C (charge of electron)
12 //calculate
13 // Since dE*dt>=h/(4*pi) (uncertainty relation for
   energy)
14 dE=h/(4*pi*dt); // calculation of minimum
   uncertainty in the energy
15 printf('\nThe uncertainty in the energy of the
   photon is \tdE=%1.2E J',dE);
16 dE=dE/e; //changing unit from J to eV
17 printf('\n\t\t\t\t\t  =%1.1E eV',dE);
```

Scilab code Exa 5.22 Calculate minimum error in the energy

```
1 //chapter 5
2 //example 5.22
3 //Calculate minimum error in the energy
4 //page 111-112
5 clear;
6 clc;
7 //given
```

```
dT=2.5E-14; // in sec (average life time)
h=6.63E-34; // in J-s (Planck'c constant)
pi=3.14; // value of pi used in the solution
e=1.6*1E-19; // in C (charge of electron)
//calculate
// Since dE*dt>=h/(4*pi) (uncertainty relation for
   energy)
dE=h/(4*pi*dT); // calculation of minimum
   uncertainty in the energy
printf('\nThe uncertainty in the energy of the
   photon is \tdE=%1.1E J',dE);
dE=dE/e; //changing unit from J to eV
printf('\n\t\t\t\t\t  =%1.1E eV',dE);
```

Scilab code Exa 5.23 Calculate energy corresponding to the 2nd and 4th quantum states

```
//chapter 5
//example 5.23
//Calculate energy corresponding to the 2nd and 4th
   quantum states
//page 112
clear;
clc;
//given
a=2; // in Angstrom (length of the box)
m=9.1E-31; // in Kg (mass of electron)
h=6.626E-34; // in J-s (Planck'c constant)
n2=2, n4=4; //  two quantum states
e=1.6*1E-19; // in C (charge of electron)
//calculate
a=a*1E-10; // since a is in Angstrom
// Since  E_n=n^2*h^2/(8*m*a^2)   (Energy
   corresponding to nth quantum state)
E2=n2^2*h^2/(8*m*a^2); // calculation of energy
```

```
    corresponding to the 2nd quantum state
17 printf('\nThe energy corresponding to the 2nd
    quantum state is \tE2=%1.3E J',E2);
18 E2=E2/e; //changing unit from J to eV
19 printf('\n\t\t\t\t\t\t  =%1.4E eV',E2);
20 E4=n4^2*h^2/(8*m*a^2); // calculation of energy
    corresponding to the 4nd quantum state
21 printf('\nThe energy corresponding to the 4nd
    quantum state is \tE4=%1.3E J',E4);
22 E4=E4/e; //changing unit from J to eV
23 printf('\n\t\t\t\t\t\t  =%1.4E eV',E4);
24 // Note: The answer in the book is wrong due to
    calculation mistake
```

Scilab code Exa 5.24 Calculate energy corresponding to the ground and first two excited states

```
1  //chapter 5
2  //example 5.24
3  //Calculate energy corresponding to the ground and
    first two excited states
4  //page 113
5  clear;
6  clc;
7  //given
8  a=1E-10; // in m (width of the well)
9  m=9.1E-31; // in Kg (mass of electron)
10 h=6.626E-34; // in J-s (Planck'c constant)
11 n1=1, n2=2, n3=3; //  ground and first two excited
    states
12 e=1.6*1E-19; // in C (charge of electron)
13 //calculate
14 // Since  E_n=n^2*h^2/(8*m*a^2)   (Energy
    corresponding to nth quantum state)
15 E1=n1^2*h^2/(8*m*a^2); // calculation of energy
```

```
    corresponding to the Ground state
16 printf('\nThe energy corresponding to the ground
    state is \tE1=%1.3E J',E1);
17 E1=E1/e; //changing unit from J to eV
18 printf('\n\t\t\t\t\t\t\t  =%.2f eV',E1);
19 E2=n2^2*h^2/(8*m*a^2); // calculation of energy
    corresponding to the 1st excited state
20 printf('\nThe energy corresponding to the 1st
    excited state is \tE2=%1.3E J',E2);
21 E2=E2/e; //changing unit from J to eV
22 printf('\n\t\t\t\t\t\t\t  =%.2f eV',E2);
23 E3=n3^2*h^2/(8*m*a^2); // calculation of energy
    corresponding to the 2nd excited state
24 printf('\nThe energy corresponding to the 2nd
    excited state is \tE3=%1.3E J',E3);
25 E3=E3/e; //changing unit from J to eV
26 printf('\n\t\t\t\t\t\t\t  =%.2f eV',E3);
27 // Note: There is slight variation in the answer due
    to round off
```

Scilab code Exa 5.25 Calculate minimum uncertainty in the velocity of electron

```
1 //chapter 5
2 //example 5.25
3 //Calculate minimum uncertainty in the velocity of
    electron
4 //page 113
5 clear;
6 clc;
7 //given
8 dx=1E-8; // in m (length of box or uncertainty in
    the position)
9 h=6.626E-34; // in J-s (Planck'c constant)
10 m=9.1E-31; // in Kg (mass of electron)
```

```
11 //calculate
12 // From uncertainty principle dx*dp=h and dp=m*dv
13 // therefore we have
14 dv=h/(m*dx); // calculation of minimum uncertainty
      in the velocity
15 printf('\nThe minimum uncertainty in the velocity of
      electron is \t dv=%1.2E m/s',dv);
16 dv=dv*1E-3; // changing unit from m/s to Km/s
17 printf('\n\t\t\t\t\t\t\t    =%.1f Km/s',dv);
18 // Note: There is slight variation in the answer due
      to round off
```

Scilab code Exa 5.26 Calculate minimum energy of electron

```
1 //chapter 5
2 //example 5.26
3 //Calculate minimum energy of electron
4 //page 113-114
5 clear;
6 clc;
7 //given
8 a=4E-10; // in m (length of the box)
9 m=9.1E-31; // in Kg (mass of electron)
10 h=6.626E-34; // in J-s (Planck'c constant)
11 n1=1; //  ground state
12 e=1.6*1E-19; // in C (charge of electron)
13 //calculate
14 // Since  E_n=n^2*h^2/(8*m*a^2)   (Energy
      corresponding to nth quantum state)
15 E1=n1^2*h^2/(8*m*a^2); // calculation of energy
      corresponding to the ground state
16 printf('\nThe minimum energy of electron is \tE1=%1
      .3E J',E1);
17 E1=E1/e; //changing unit from J to eV
18 printf('\n\t\t\t\t\t  =%.3f eV',E1);
```

```
19 // Note: The answer in the book corresponding to J
      is wrong due to printing error.
```

Chapter 6

Electron Theory and Band Theory of Metals

Scilab code Exa 6.1 Calculate mean free path of electron

```
1 //chapter 6
2 //example 6.1
3 //Calculate mean free path of electron
4 //page 146
5 clear;
6 clc;
7 //given
8 n=8.5E28; // in 1/m^3 (density of electron)
9 m_e=9.11E-31; // in Kg (mass of electron)
10 k=1.38E-23; // in J/K (Boltzmann's constant)
11 e=1.6E-19; // in C (charge of electron)
12 T=300; // in K (temperature)
13 p=1.69E-8; // in ohm-m (resistivity)
14 //calculate
15 lambda=sqrt(3*k*m_e*T)/(n*e^2*p); // calculation of
    mean free path
16 printf('\nThe mean free path of electron is \t=%1.2E
    m',lambda);
17 lambda=lambda*1E9; // changing unit from m to
```

```
    nanometer
18 printf('\n\t\t\t\t=%.2f nm',lambda);
19 // Note: answer in the book is wrong due to printing
       mistake
```

Scilab code Exa 6.2 Calculate the temperature

```
1  //chapter 6
2  //example 6.2
3  //Calculate the temperature
4  //page 146
5  clear;
6  clc;
7  //given
8
9  k=1.38E-23; // in J/K (Boltzmann's constant)
10 e=1.6E-19; // in C (charge of electron)
11 P_E=1; // in percentage (probability that a state
       with an energy 0.5 eV above Fermi energy will be
       occupied)
12 E=0.5; // in eV (energy above Fermi level)
13 //calculate
14 P_E=1/100; // changing percentage into ratio
15 E=E*e; // changing unit from eV to J
16 // P_E=1/(1+exp((E-E_F)/k*T))
17 // Rearranging this equation, we get
18 // T=(E-E_F)/k*log((1/P_E)-1)
19 // Since E-E_F has been denoted by E  therefore
20 T=E/(k*log((1/P_E)-1));
21 printf('\nThe temperature is \tT=%.f K',T);
22 // Note: There is slight variation in the answer due
       to logarithm function
```

Scilab code Exa 6.3 Calculate relaxation time of conduction electrons

```
1 //chapter 6
2 //example 6.3
3 //Calculate relaxation time of conduction electrons
4 //page 147
5 clear;
6 clc;
7 //given
8 n=5.8E28; // in 1/m^3 (density of electron)
9 m=9.1E-31; // in Kg (mass of electron)
10 e=1.6E-19; // in C (charge of electron)
11 p=1.54E-8; // in ohm-m (resistivity)
12 //calculate
13 t=m/(n*e^2*p); // calculation of relaxation time
14 printf('\nThe relaxation time of conduction
   electrons is %1.2E sec',t);
```

Scilab code Exa 6.4 Calculate mean free path traveeled by the electrons

```
1 //chapter 6
2 //example 6.4
3 //Calculate mean free path traveeled by the
   electrons
4 //page 147
5 clear;
6 clc;
7 //given
8 n=8.5E28; // in 1/m^3 (density of electron)
9 m=9.1E-31; // in Kg (mass of electron)
10 e=1.6E-19; // in C (charge of electron)
11 sigma=6E7; // in 1/ohm-m (conductivity)
12 E_F=7; // in E=eV (Fermi energy of Copper)
13 //calculate
14 E_F=E_F*e; // changing unit from eV to J
```

```
15 v_F=sqrt(2*E_F/m); // calculation of velocity of
     electrons
16 printf('\nThe velocity of the electrons is \t\t\tv_F
     =%1.1E m/s',v_F);
17 // Since sigma=n*e^2*lambda/(2*m*v_F)
18 // Therefore we have
19 lambda=2*m*v_F*sigma/(n*e^2); // calculation of mean
      free path
20 lambda=lambda*1E10; // changing unit from m to
     Angstrom
21 printf('\n\nThe mean free path traveled by the
     electrons is \t%.f Angstrom',lambda);
22 // Note: Answer in the book is wrong due to the use
     of round-off value of v_F as calculated in the
     first part.
```

Scilab code Exa 6.5 Calculate relaxation time of conduction electrons

```
1 //chapter 6
2 //example 6.5
3 //Calculate relaxation time of conduction electrons
4 //page 147-148
5 clear;
6 clc;
7 //given
8 n=6.5E28; // in 1/m^3 (density of electron)
9 m=9.1E-31; // in Kg (mass of electron)
10 e=1.6E-19; // in C (charge of electron)
11 p=1.43E-8; // in ohm-m (resistivity)
12 //calculate
13 t=m/(n*e^2*p); // calculation of relaxation time
14 printf('\nThe relaxation time of conduction
     electrons is %1.2E sec',t);
```

Scilab code Exa 6.6 Calculate average kinetic energy and velocity of molecules

```
1 //chapter 6
2 //example 6.6
3 //Calculate average kinetic energy and velocity of
     molecules
4 //page 148
5 clear;
6 clc;
7 //given
8 T=30; // in Celcius (temperature)
9 k=1.38E-23; // in J/K (Boltzmann's constant)
10 m_p=1.67E-27; // in Kg (mass of proton)
11 e=1.6E-19; // in C (charge of electron)
12 //calculate
13 T=T+273; // changing temperature from Celcius to
     Kelvin
14 KE=(3/2)*k*T; // calculation of average kinetic
     energy
15 printf('\nThe average kinetic energy of gas ,
     molecules is \tKE=%3.2E J',KE);
16 KE=KE/e; // changing unit from J to eV
17 printf('\n\t\t\t\t\t\t  =%f eV',KE);
18 m=1.008*2*m_p; // calculating mass of hydrogen gas
     molecule
19 c=sqrt(3*k*T/m); // calculation of velocity
20 printf('\n\nThe velocity of molecules is \tc=%.2f m/
     s',c);
21 // Note: There is calculation mistake in the answer
     of energy given in eV and that of velocity
```

Scilab code Exa 6.7 Calculate velocity of electron and proton

```
1 //chapter 6
2 //example 6.7
3 //Calculate velocity of electron and proton
4 //page 148-149
5 clear;
6 clc;
7 //given
8 E=10; // in eV (kinetic energy for each electron and
      proton)
9 m_e=9.1E-31; // in Kg (mass of electron)
10 m_p=1.67E-27; // in Kg (mass of proton)
11 e=1.6E-19; // in C (charge of electron)
12 //calculate
13 E=E*e; // changing unit from eV to J
14 // since E=m*v^2/2
15 // therefore  v=sqrt(2E/m)
16 v_e=sqrt(2*E/m_e); // calculation of kinetic energy
    of electron
17 printf('\nThe kinetic energy of electron is \tv_e=%1
    .3E m/s',v_e);
18 v_p=sqrt(2*E/m_p); // calculation of kinetic energy
    of proton
19 printf('\nThe kinetic energy of proton is \tv_p=%1.3
    E m/s',v_p);
20 // Note: The answer in the book for both kinetic
    energy of electron and that of proton is wrong
    due to calculation mistake
```

Scilab code Exa 6.8 Calculate drift velocity of free electrons

```
1 //chapter 6
2 //example 6.8
3 //Calculate drift velocity of free electrons
4 //page 149
5 clear;
```

```
clc;
//given
I=100; // in A (current in the wire)
e=1.6E-19; // in C (charge of electron)
A=10; // in mm^2 (cross-sectional area)
n=8.5E28; // in 1/m^3 (density of electron)
//calculate
A=A*1E-6; // changing unit from mm^2 to m^2
v_d=I/(n*A*e);
printf('\nThe drift velocity of free electrons is \
   tv_d=%1.3E m/s',v_d);
```

Scilab code Exa 6.9 Calculate average drift velocity of electrons

```
//chapter 6
//example 6.9
//Calculate average drift velocity of electrons
//page 149
clear;
clc;
//given
I=4; // in A (current in the conductor)
e=1.6E-19; // in C (charge of electron)
A=1E-6; // in m^2 (cross-sectional area)
N_A=6.02E23; // in atoms/gram-atom (Avogadro's
   number)
p=8.9; // in g/cm^3 (density)
M=63.6; // atomic mass of copper
//calculate
n=N_A*p/M; // Calculation of density of electrons in
    g/cm^3
printf('\nThe density of copper atoms is \tn=%1.2E
   atoms/m^3',n);
n=n*1E6; // changing unit from g/cm^3 to g/m^3
printf('\n\t\t\t\t =%1.2E atoms/m^3',n);
```

```
19 v_d=I/(n*A*e);
20 printf('\n\nThe average drift velocity of free
      electrons is \tv_d=%1.1E m/s',v_d);
```

Scilab code Exa 6.10 Calculate mobility of electrons

```
1 //chapter 6
2 //example 6.10
3 //Calculate mobility of electrons
4 //page 149-150
5 clear;
6 clc;
7 //given
8 n=9E28; // in 1/m^3 (density of valence electrons)
9 sigma=6E7; // in mho/m (conductivity of copper)
10 e=1.6E-19; // in C (charge of electron)
11 //calculate
12 // Since sigma=n*e*mu  therefore
13 mu=sigma/(n*e); // calculation of mobility of
      electron
14 printf('\n\nThe mobility of electrons is \t%1.2E  m
      ^2/V-s',mu);
```

Scilab code Exa 6.11 Calculate average energy of free electron at 0K and corresponding temperature for a classical particle or an ideal gas

```
1 //chapter 6
2 //example 6.11
3 //Calculate average energy of free electron at 0K
      and corresponding temperature for a classical
      particle (an ideal gas)
4 //page 150
5 clear;
```

```
6  clc;
7  //given
8  E_F=5.51; // in eV (Fermi energy in Silver)
9  k=1.38E-23; // in J/K (Boltzmann's constant)
10 e=1.6E-19; // in C (charge of electron)
11 //calculate
12 // part-(a)
13 Eo=(3/5)*E_F; // calculation of average energy of
      free electron at 0K
14 printf('\n\nThe average energy of free electron at 0
      K is \tEo=%.3f eV',Eo);
15 // part-(b)
16 Eo=Eo*e; // changing unit from eV to J
17 // Since for a classical particle E=(3/2)*k*T
18 // therefroe we have
19 T=(2/3)*Eo/k; // calculation of temperature for a
      classical particle (an ideal gas)
20 printf('\n\nThe temperature at which a classical
      particle have this much energy is \t T=%1.3E K',T
      );
```

Scilab code Exa 6.12 Calculate electron density for a metal

```
1  //chapter 6
2  //example 6.12
3  //Calculate electron density for a metal
4  //page 150
5  clear;
6  clc;
7  //given
8  E_F_L=4.7; // in eV (Fermi energy in Lithium)
9  E_F_M=2.35; // in eV (Fermi energy in a metal)
10 n_L=4.6E28; // in 1/m^3 (density of electron in
      Lithium)
11 //calculate
```

```
12 // Since n=((2*m/h)^3/2)*E_F^(3/2)*(8*pi/3) and all
   things except E_F are constant
13 // Therefore we have n=C*E_F^(3/2)   where C is
   proportionality constant
14 // n1/n2=(E_F_1/E_F_2)^(3/2)
15 // Therefore we have
16 n_M=n_L*(E_F_M/E_F_L); // calculation of electron
   density for a metal
17 printf('\nThe lectron density for a metal is \t=%1.1
   E 1/m^3', n_M);
18 //Note: Answer in the book is wrong due to priting
   error
```

Chapter 7

Dielectric Properties

Scilab code Exa 7.1 Calculate the capacitance of capacitor and charge on the plates

```
1 //chapter 7
2 //example 7.1
3 //Calculate the capacitance of capacitor and charge
     on the plates
4 //page 187
5 clear;
6 clc;
7 //given
8 A=100; // in cm^2 (cross-sectional area)
9 d=1; // in cm (seperation between plates)
10 Eo=8.85E-12; // in F/m (absolute permittivity)
11 V=100; // in V (potential difference)
12 //calculate
13 A=A*1E-4; // changing unit from cm^2 to m^2
14 d=d*1E-2; // changing unit from cm to m
15 C=Eo*A/d;// calculation of capacitance
16 Q=C*V; // calculation of charge
17 printf('\nThe capacitance of capacitor is \t C=%1.2E
     C',C);
18 C=C*1E12; // changing unit of capacitance from F to
```

```
    pF
19 printf('\n\t\t\t\t\t  =%.2f pF',C);
20 printf('\n\nThe charge on the plates is \t\t Q=%1.2E
    C',Q);
```

Scilab code Exa 7.2 Calculate the resultant voltage across the capacitor

```
1  //chapter 7
2  //example 7.2
3  //Calculate the resultant voltage across the
    capacitor
4  //page 187
5  clear;
6  clc;
7  //given
8  A=650; // in mm^2 (cross-sectional area)
9  d=4; // in mm (seperation between plates)
10 Eo=8.85E-12; // in F/m (absolute permittivity)
11 Er=3.5; // di-electric constant of the material
12 Q=2E-10; // in C (charge on plates)
13 //calculate
14 A=A*1E-6; // changing unit from mm^2 to m^2
15 d=d*1E-3; // changing unit from mm to m
16 C=Er*Eo*A/d;// calculation of capacitance
17 V=Q/C; // calculation of charge
18 printf('\nThe capacitance of capacitor is \t C=%1.2E
    C',C);
19 C=C*1E12; // changing unit of capacitance from F to
    pF
20 printf('\n\t\t\t\t\t  =%.2f pF',C);
21 printf('\n\nThe resultant voltage across the
    capacitor is \t V=%.2f V',V);
22 // NOTE: The answer is wrong due to calculation
    mistake. The mistake is that in the book Value of
     cross-sectional area and seperation
```

```
23 //        between plates is considered in cm and di-
      electric constant has not been considered.
```

Scilab code Exa 7.3 Calculate the radius of electron cloud and dispalcement

```
1  //chapter 7
2  //example 7.3
3  //Calculate the radius of electron cloud and
      dispalcement
4  //page 188
5  clear;
6  clc;
7  //given
8  N=2.7E25; // in 1/m^3 (density of atoms)
9  E=1E6; // in V/m (electric field)
10 Z=2; // atomic number of Helium
11 Eo=8.85E-12; // in F/m (absolute permittivity)
12 Er=1.0000684; // (dielectric constant of the
      material)
13 e=1.6E-19; // in C (charge of electron)
14 pi=3.14; // value of pi used in the solution
15 //calculate
16 // since  alpha=Eo*(Er-1)/N=4*pi*Eo*r_0^3
17 // Therefore we have r_0^3=(Er-1)/(4*pi*N)
18 r_0=((Er-1)/(4*pi*N))^(1/3);// calculation of radius
      of electron cloud
19 printf('\nThe radius of electron cloud is \t r_0=%1
      .2E m',r_0);
20 x=4*pi*Eo*E*r_0/(Z*e); // calculation of
      dispalcement
21 printf('\n\nThe displacement is x=%1.2E m',x);
22 // NOTE: The answer is wrong due to calculation
      mistake.
```

Scilab code Exa 7.4 Calculate the dipole moment induced in each atom and atomic polarisability

```
1 //chapter 7
2 //example 7.4
3 //Calculate the dipole moment induced in each atom
     and atomic polarisability
4 //page 188-189
5 clear;
6 clc;
7 //given
8 K=1.000134; // di-elecrtic constant of the neon gas
     at NTP
9 E=90000; // in V/m (electric field)
10 Eo=8.85E-12; // in C/N-m^2 (absolute premittivity)
11 N_A=6.023E26; // in atoms/Kg-mole (Avogadro's number
     )
12 V=22.4; // in m^3 (volume of gas at NTP
13 //calculate
14 n=N_A/V; // calculaton of density of atoms
15 // Since P=n*p=(k-1)*Eo*E
16 // therefore we have
17 p=(K-1)*Eo*E/n; // calculation of dipole moment
     induced
18 printf('\nThe dipole moment induced in each atom is
     \tp=%1.2E C-m',p);
19 alpha=p/E; // calculation of atomic polarisability
20 printf('\n\nThe atomic polarisability of neon is \t=
     %1.2E c-m^2/V',alpha);
21 // NOTE: The answer of atomic polarisability is
     wrong due to printing error
```

Scilab code Exa 7.5 Calculate the electronic polarisability of sulphur

```
1 //chapter 7
2 //example 7.5
3 //Calculate the electronic polarisability of sulphur
4 //page 189
5 clear;
6 clc;
7 //given
8 Er=3.75; // di-elecrtic constant of sulphur at 27
     degree Celcius
9 gama=1/3; // internal field constant
10 p=2050; // in Kg/m^3 (density)
11 M_A=32; // in amu (atomic weight of sulphur)
12 Eo=8.85E-12; // in F/m (absolute permittivity)
13 N=6.022E23; // Avogadro's number
14 //calculate
15 // Since ((Er-1)/(Er+2))*(M_A/p)=(N/(3*Eo))*alpha_e
16 // therefore we have
17 alpha_e=((Er-1)/(Er+2))*(M_A/p)*(3*Eo/N); //
     calculation of electronic polarisability of
     sulphur
18 printf('\nThe electronic polarisability of sulphur
     is \t=%1.2E Fm^2',alpha_e);
19 // NOTE: There is slight variation in the answer due
      to round off
```

Scilab code Exa 7.6 Calculate the electronic polarisability of Helium atoms

```
1 //chapter 7
2 //example 7.6
3 //Calculate the electronic polarisability of Helium
     atoms
4 //page 189-190
5 clear;
```

```
clc;
//given
Er=1.0000684; // di-elecrtic constant of Helium gas at NTP
Eo=8.85E-12; // in F/m (absolute permittivity)
N=2.7E25; // number of atomsper unit volume
//calculate
// Since Er-1=(N/Eo)*alpha_e
// therefore we have
alpha_e=Eo*(Er-1)/N; // calculation of electronic polarisability of Helium
printf('\nThe electronic polarisability of Helium gas is \t=%1.2E Fm^2',alpha_e);
// NOTE: There is slight variation in the answer due to round off
```

Scilab code Exa 7.7 Calculate the dielectric constant of the material

```
//chapter 7
//example 7.7
//Calculate the dielectric constant of the material
//page 190
clear;
clc;
//given
N=3E28; // in atoms/m^3 (density of atoms)
alpha_e=1E-40; // in F-m^2 (electronic polarisability)
Eo=8.85E-12; // in F/m (absolute permittivity)
//calculate
// Since (Er-1)/(Er+2)=N*alpha_e/(3*Eo)
// therefore we have
Er=(2*(N*alpha_e/(3*Eo))+1)/(1-(N*alpha_e/(3*Eo)));
 // calculation of dielectric constant of the material
```

```
16 printf('\nThe dielectric constant of the material is
       \tEr=%.3f F/m',Er);
17 // NOTE: The answer in the book is wrong due to
   calculation mistake
```

Scilab code Exa 7.8 Calculate the atomic polarisability of sulphur

```
1  //chapter 7
2  //example 7.8
3  //Calculate the atomic polarisability of sulphur
4  //page 190
5  clear;
6  clc;
7  //given
8  Er=4; // relative permittivity of sulphur
9  Eo=8.85E-12; // in F/m (absolute permittivity)
10 NA=2.08E3; // in Kg/m^3 (density of atoms in sulphur
       )
11 //calculate
12 // Since ((Er-1)/(Er+2))*(M_A/p)=(N/(3*Eo))*alpha_e
13 // therefore we have
14 alpha_e=((Er-1)/(Er+2))*(3*Eo/NA); // calculation of
       electronic polarisability of sulphur
15 printf('\nThe electronic polarisability of sulphur
   is \t=%1.2E Fm^2',alpha_e);
16 // NOTE: The answer in the book is wrong due to
   calculation mistake. Also one point to be
   mentioned is that wrong formula has been used in
   the solution but i have used the formula as used
   in the solution.
```

Scilab code Exa 7.9 calculate polarisability due to permanent dipole moment and due to deformation of the molecules

```
// chapter 7
// example 7.9
// calculate polarisability due to permanent dipole moment and due to deformation of the molecules
// page 190-191
clear;
clc;
// given
alpha1=2.5E-39; // in C^2-m/N (dielectric constant at 300K)
alpha2=2.0E-39; // in C^2-m/N (dielectric constant at 400K)
T1=300; // in K(first temperature)
T2=400; // in K(second temperature)
//calculate
// since alpha=alpha_d+alpha0 and alpha0=Beta/T
// therefore alpha=alpha_d+(Beta/T)
// since alpha1=alpha_d+(Beta/T1)  and alpha2=alpha_d+(Beta/T2)
// therefore alpha1-apha2=Beta*((1/T1)-(1/T2))
// or Beta= (alpha1-apha2)/ ((1/T1)-(1/T2))
Beta= (alpha1-alpha2)/ ((1/T1)-(1/T2)); // calculation of Beta
alpha_d=alpha1-(Beta/T1); // calculation of polarisability due to defromation
alpha0_1=Beta/T1; // calculation of polarisability due to permanent dipole moment at 300K
alpha0_2=Beta/T2;  // calculation of polarisability due to permanent dipole moment at 400K
printf('\nThe polarisability due to permanent dipole moment at 300K is \t %1.2E C^2-m/N',alpha0_1);
printf('\nThe polarisability due to permanent dipole moment at 400K is \t %1.2E C^2-m/N',alpha0_2);
printf('\n\nThe polarisability due to deformation of the molecules is \t %1.2E C^2-m/N',alpha_d);
```

Scilab code Exa 7.10 determine the percentage of ionic polarisability in sodium crystal

```
1 // chapter 7
2 // example 7.10
3 // determine the percentage of ionic polarisability
     in sodium crystal
4 // page 191-192
5 clear;
6 clc;
7 // given
8 n=1.5; // refractive index
9 Er=5.6;// dielectric constant
10 //calculate
11 // since (Er-1)/(Er+2)=N*(alpha_e+alpha_i)/(3*E0)
     Clausius-Mossotti equation
12 // and  (n^2-1)/(n^2+2)=N*alpha_e/(3*E0)
13 // from above two equations, we get  ((n^2-1)/(n
     ^2+2))*((Er+2)/(Er-1))=alpha_e/(alpha_e+alpha_i)
14 // or alpha_i/ (alpha_e+alpha_i)= 1-((n^2-1)/(n^2+2)
     )*((Er+2)/(Er-1))= (say P)
15 // where P is fractional ionisational polarisability
16 P=1-((n^2-1)/(n^2+2))*((Er+2)/(Er-1)); //
     calculation of fractional ionisational
     polarisability
17 P=P*100; // calculation of percentage of
     ionisational polarisability
18 printf('\nThe percentage of  ionisational
     polarisability is \t%.1f percent',P);
```

Chapter 8

Magnetic Properties

Scilab code Exa 8.1 Calculate intensity of magnetism and magnetic flux density

```
1 //chapter 8
2 //example 8.1
3 //Calculate intensity of magnetism and magnetic flux
      density
4 //page 236
5 clear;
6 clc;
7 //given
8 X=-0.5E-5; // magnetic susceptibility of silicon
9 H=0.9E4; // in A/m (magnetic field intensity)
10 mu0=4*%pi*1E-7; // in H/m (absolute permeability)
11 //calculate
12 I=X*H; // calculation of intensity of magnetism
13 printf('\nThe intensity of magnetism is \tI=%.3f A/m
     ',I);
14 B=mu0*H*(1+X); // calculation of magnetic flux
     density
15 printf('\nThe magnetic flux density is \tB=%.3f Wb/m
     ^2',B);
16 // Note: The answer in the book is wrong. This is
```

```
because the value of H given in the question is H
=0.9E4 A/m but in the solution the value of H
that has been used is H=9.9E4 A/m.
```

Scilab code Exa 8.2 Calculate change in magnetic moment

```
1 //chapter 8
2 //example 8.2
3 //Calculate change in magnetic moment
4 //page 236
5 clear;
6 clc;
7 //given
8 r=0.052; // in nm (radius of orbit)
9 B=1; // in Wb/m^2 (magnetic field of induction)
10 e=1.6E-19; // in C (charge of electron)
11 m=9.1E-31; // in Kg (mass of electron)
12 //calculate
13 r=0.052*1E-9; // changing unit from nm to m
14 d_mu=(e^2*r^2*B)/(4*m); // calculation of change in
   magnetic moment
15 printf('\nThe change in magnetic moment is \t%1.4E
   Am^2',d_mu);
16 // Note: The answer in the book is wrong due to
   caluclation mistake
```

Scilab code Exa 8.3 Calculate relative permeability of a ferromagentic material

```
1 //chapter 8
2 //example 8.3
3 //Calculate relative permeability of a ferromagentic
    material
```

```
4 //page 236
5 clear;
6 clc;
7 //given
8 H=220; // in A/m (magnetic field intensity)
9 I=3300; // in A/m (intensity of magnetisation)
10 //calculate
11 mu_r=1+(I/H); // calculation of relative
      permeability
12 printf('\nThe relative permeability of a
      ferromagentic material is  %.f',mu_r);
```

Scilab code Exa 8.4 Calculate magnetic force and relative permeability

```
1 //chapter 8
2 //example 8.4
3 //Calculate magnetic force and relative permeability
4 //page 236-237
5 clear;
6 clc;
7 //given
8 I=3000; // in A/m (intensity of magnetisation)
9 B=0.005; // in Wb/m^2 (magnetic flus intensity)
10 pi=3.14;// value of pi used in the solution
11 mu0=4*pi*1E-7; // in H/m (absolute permeability)
12 //calculate
13 H=(B/mu0)-I; // calculation of magnetic force
14 printf('\nThe magnetic force is  \tH=%.3f',H);
15 mu_r=(I/H)+1; // calculation of relative
      permeability
16 printf('\nThe relative permeability is \t%.3f',mu_r)
      ;
```

Scilab code Exa 8.5 Calculate current through the solenoid

```
1 //chapter 8
2 //example 8.5
3 //Calculate current through the solenoid
4 //page 237
5 clear;
6 clc;
7 //given
8 H=4E3; // in A/m (magnetic field intensity)
9 N=60; // number of turns
10 l=12; // in cm (length of solenoid)
11 //calculate
12 n=N/(l*1E-2); // calculation of number of turns per
      unit metre
13 // Snice H=n*i;
14 i=H/n; // calculation of current through the
      solenoid
15 printf('\nThe current through the solenoid is \ti=%.
      f A',i);
```

Scilab code Exa 8.6 Calculate flux density magnetic intensity and relative permeability

```
1 //chapter 8
2 //example 8.6
3 //Calculate flux density, magnetic intensity and
      relative permeability
4 //page 237
5 clear;
6 clc;
7 //given
8 l=30; // in cm (length of solenoid)
9 A=1; // in cm^2 (cross-sectional area)
10 N=300; // number of turns
```

```
11 i=0.032; // in A (current through the winding)
12 phi_B=2E-6; // in Wb (magnetic flux)
13 pi=3.14;// value of pi used in the solution
14 mu0=4*pi*1E-7; // in H/m (absolute permeability)
15 //calculate
16 l=l*1E-2; // changing unit from cm to m
17 A=A*1E-4; // changing unit from cm^2 to m^2
18 B=phi_B/A; // calculation of flux density
19 printf('\nThe flux density is \tB=%1.0E Wb/m^2',B);
20 H=N*i/l; // calculation of magnetic intensity
21 printf('\nThe magnetic intensity is \tH=%.f A-turns/
     m',H);
22 mu=B/H; // calcluation of absolute permeability of
     iron
23 mu_r=mu/mu0; // calcluation of relative permeability
      of iron
24 printf('\nThe relative permeability of iron is \t%.f
     ',mu_r);
25 // Note: The value of relative permeability varies
     slightly due to the use of round off value mu as
     calculated
```

Scilab code Exa 8.7 Calculate Hystersis loss per cycle

```
1 //chapter 8
2 //example 8.7
3 //Calculate Hystersis loss per cycle
4 //page 238
5 clear;
6 clc;
7 //given
8 A=100; // in m^2 (area of Hysteresis loop)
9 B=0.01; // in Wb/m^2 (unit space along vertical axis
      or magnetic flux density)
10 H=40; // in A/m (unit space along horizontal axis or
```

```
    magnetic fild ntensity)
//calculate
H_L=A*B*H; // calculation of magnetic intensity
printf('\nThe Hystersis loss per cycle is  %.f J/m^2
   ',H_L);
```

Chapter 9

Semiconductors

Scilab code Exa 9.2 Find the temperature at which number of electrons becomes 10 times

```
1 // chapter 9
2 // example 9.2
3 // Find the temperature at which number of electrons
      becomes 10 times
4 // page 272
5 clear;
6 clc;
7 //given
8 Eg=0.67; // in eV (Energy band gap)
9 k=1.38E-23; // in J/K (Boltzmann s constant)
10 T1=298; // in K (room temperature)
11 e=1.6E-19; // in C (charge of electron)
12 K=10; // ratio of number of electrons at different
     temperature
13 // calculate
14 Eg=Eg*e; // changing unit from eV to Joule
15 // since ne=Ke*exp(-Eg/(2*k*T))
16 // and ne/ne1=exp(-Eg/(2*k*T))/exp(-Eg/(2*k*T1)) and
      ne/ne1=K=10
17 // therefore we have 10=exp(-Eg/(2*k*T))/exp(-Eg/(2*
```

```
    k*T1))
18 // re-arranging the equation for T, we get T2=1/((1/
    T1)-((2*k*log(10))/Eg))
19 T=1/((1/T1)-((2*k*log(10))/Eg)); // calculation of
    the temperature
20 printf('\nThe temperature at which number of
    electrons in the conduction band of a
    semiconductor increases by a factor of 10 is \tT=
    %.f K',T);
21 // Note: there s slight variation in the answer due
    to round off calculation
```

Scilab code Exa 9.3 find the resistance of intrinsic germanium

```
1  // chapter 9
2  // example 9.3
3  // find the resistance of intrinsic germanium
4  // page 272-273
5  // given
6  clear;
7  clc;
8  ni=2.5E13; // in /cm^3 (intrinsic carrier density)
9  ue=3900; // in cm^2/(V-s) (electron mobilities)
10 uh=1900; // in cm^2/(V-s) (hole mobilities)
11 e=1.6E-19; // in C (charge of electron)
12 l=1; // in cm (lenght of the box)
13 b=1,h=1; // in mm (dimensions of germanium rod )
14 // calculate
15 ni=ni*1E6; // changing unit from 1/cm^3 to 1/m^3
16 ue=ue*1E-4; // changing unit from cm^2 to m^2
17 uh=uh*1E-4; // changing unit from cm^2 to m^2
18 sigma=ni*e*(ue+uh); // calculation of conductivity
19 rho=1/sigma; // calculation of resistivity
20 l=l*1E-2; // changing unit from mm to m for length
21 A=(b*1E-3)*(h*1E-3); // changing unit from mm to m
```

```
    for width and height and calculation of cross-
    sectional area
22 R=rho*l/A; // calculation of resistance
23 printf('\nThe resistance of intrinsic germanium is \
    tR=%1.1E ohm',R);
```

Scilab code Exa 9.4 find the electrical conductivity and resistivity of germanium

```
1  // chapter 9
2  // example 9.4
3  // find the electrical conductivity and resistivity
       of germanium
4  // page 273
5  clear;
6  clc;
7  // given
8  ne=2.5E19; // in /m^3 (electron  density)
9  nh=2.5E19; // in /m^3 (hole density)
10 ue=0.36; // in m^2/(V-s) (electron mobilities)
11 uh=0.17; // in m^2/(V-s) (hole mobilities)
12 e=1.6E-19; // in C (charge of electron)
13 // calculate
14 // since ne=nh=ni,  therefore we have
15 ni=nh;
16 sigma=ni*e*(ue+uh); // calculation of conductivity
17 printf('\nThe conductivity of germanium is  %.2f /
    ohm-m',sigma);
18 rho=1/sigma; // calculation of resistivity
19 printf('\nThe resistivity of germanium is  %.2f ohm-
    m',rho);
20 // Note:  In the question, the value of ni has been
    misprinted as 2.5E-19 /m^3 rather it should be
    2.5E19 /m^3. I have used 2.5E19 /m^3
```

Scilab code Exa 9.5 find the equilibrium hole concentration and conductivity

```
// chapter 9
// example 9.5
// find the equilibrium hole concentration and
   conductivity
// page 273-274
clear;
clc;
// given
ni=1.5E16; // in /m^3 (intrinsic carrier density)
ue=0.135; // in m^2/(V-s) (electron mobilities)
uh=0.048; // in m^2/(V-s) (hole mobilities)
e=1.6E-19; // in C (charge of electron)
ND=1E23; // in atom/m^3 (doping concentration)
// calculate
sigma_i=ni*e*(ue+uh); // calculation of intrinsic
   conductivity
printf('\nThe intrinsic conductivity for silicon is
    %1.2E S',sigma_i);
sigma=ND*ue*e; // calculation of conductivity after
   doping
printf('\n\nThe conductivity after doping with
   phosphorus atoms is  %1.2E S',sigma);
rho=ni^2/ND; // calculation of equilibrium hole
   concentration
printf('\n\nThe equilibrium hole concentration is
   %1.2E /m^3',rho);
```

Scilab code Exa 9.6 find intrinsic concuctivity and doping conductivity

```
// chapter 9
// example 9.6
// find intrinsic concuctivity and doping
   conductivity
// page 274
clear;
clc;
// given
ni=1.5E16; // in /m^3 (intrinsic carrier density)
ue=0.13; // in m^2/(V-s) (electron mobilities)
uh=0.05; // in m^2/(V-s) (hole mobilities)
e=1.6E-19; // in C (charge of electron)
ne=5E20; // in /m^3 (concentration of donor type
   impurity)
nh=5E20; // in /m^3 (concentration of acceptor type
   impurity)
// calculate
// part-i
sigma=ni*e*(ue+uh); // calculation of intrinsic
   conductivity
printf('\nThe intrinsic conductivity for silicon is
    %1.2E (ohm-m)^-1',sigma);
// part-ii
// since 1 donor atom is in 1E8 Si atoms, hence
   holes concentration can be neglected
sigma=ne*e*ue; // calculation of conductivity after
   doping with donor type impurity
printf('\n\nThe conductivity after doping with donor
    type impurity is  %.1f (ohm-m)^-1',sigma);
// part-iii
// since 1 acceptor atom is in 1E8 Si atoms, hence
   electron concentration can be neglected
sigma=nh*e*uh; // calculation of conductivity after
   doping with acceptor type impurity
printf('\n\nThe conductivity after doping with
   acceptor type impurity is  %.f (ohm-m)^-1',sigma)
   ;
// Note: In question the value of  ne and nh has
```

```
been misprinted as 5E28 atoms/m^3 which is too
big but the solution has used the correct value 5
E20 atoms/m^3. I have also used this value.
```

Scilab code Exa 9.7 find density of hole carriers at room temperature

```
1 // chapter 9
2 // example 9.7
3 // find density of hole carriers at room temperature
4 // page 274-275
5 clear;
6 clc;
7 // given
8 ni=1E20; // in /m^3 (intrinsic carrier density)
9 ND=1E21; // in /m^3 (donor impurity concentration)
10 // calculate
11 nh=ni^2/ND; // calculation of density of hole
   carriers at room temperature
12 printf('\nThe density of hole carriers at room
   temperature is \tnh=%1.0E /m^3',nh);
13 // Note: answer in the book is wrong due to printing
    mistake
```

Scilab code Exa 9.8 find intrinsic carrier density and conductivity at 300K in germanium

```
1 // chapter 9
2 // example 9.8
3 // find intrinsic carrier density and conductivity
   at 300K in germanium
4 // page 275
5 clear;
6 clc;
```

```
M=72.6; // atomic mass of germanium
P=5400; // in Kg/m^3 (density)
ue=0.4; // in m^2/V-s (mobility of electrons)
uh=0.2; // in m^2/V-s (mobility of holes)
Eg=0.7; // in eV (Band gap)
m=9.1E-31; // in Kg (mass of electron)
k=1.38E-23; // in J/K (Boltzmann s constant)
T=300; // in K (temperature)
h=6.63E-34;// in J/s (Planck s constant)
pi=3.14; // value of pi used in the solution
e=1.6E-19; // in C(charge of electron)
// calculate
Eg=Eg*e; // changing unit from eV to J
ni=2*(2*pi*m*k*T/h^2)^(3/2)*exp(-Eg/(2*k*T));
printf('\nThe intrinsic carrier density for
   germanium at 300K is \tni=%1.1E /m^3',ni);
sigma=ni*e*(ue+uh);
printf('\nThe conductivity of germanium is \t%1.2f (
   ohm-m)^-1',sigma);
// Note: Answer in the book is wrong due to
   calculation mistake
```

Scilab code Exa 9.9 Find the energy band gap

```
// chapter 9
// example 9.9
// Find the energy band gap
// page 275
clear;
clc;
//given
rho1=4.5;// in ohm-m (resistivity at 20 degree
   Celcius)
rho2=2.0;// in ohm-m (resistivity at 32 degree
   Celcius)
```

```
10 k=1.38E-23; // in J/K (Boltzmann s constant)
11 T1=20, T2=32; // in degree Celcius (two temperatures
   )
12 e=1.6E-19; // in C (charge of electron)
13 // calculate
14 T1=T1+273;// changing unit from degree Celius to K
15 T2=T2+273;// changing unit from degree Celius to K
16 // since sigma=e*u*C*T^(3/2)*exp(-Eg/(2*k*T))
17 // therefore sigma1/sigma2=(T1/T2)^3/2*exp((-Eg/(2*k
   )*((1/T1)-(1/T2))
18 // and sigma=1/rho
19 // therefore we have rho2/rho1=(T1/T2)^3/2*exp((-Eg
   /(2*k)*((1/T1)-(1/T2))
20 // re-arranging above equation for Eg, we get Eg=(2*
   k/((1/T1)-(1/T2)))*((3/2)*log(T1/T2)-log(rho2/
   rho1))
21 Eg=(2*k/((1/T1)-(1/T2)))*((3/2)*log(T1/T2)-log(rho2/
   rho1));
22 printf('\nThe energy band gap is \tEg=%1.2E J',Eg);
23 Eg=Eg/e;// changing unit from J to eV
24 printf('\n\t\t\t  =%.2f eV',Eg);
```

Scilab code Exa 9.10 Find the electron and hole concentrations and the resistivity

```
1 // chapter 9
2 // example 9.10
3 // Find the electron and hole concentrations and the
    resistivity
4 // page 276
5 clear;
6 clc;
7 //given
8 rho=2300; // in ohm-m (resistivity of pure silicon)
9 ue=0.135; // in m^2/V-s (mobility of electron)
```

```
uh=0.048; // in m^2/V-s (mobility of electron)
Nd=1E19;// in /m^3 (doping concentration)
e=1.6E-19;// in C (charge of electron)
//calculate
// since sigma=ni*e*(ue+uh) and sigma=1/rho
// therefore ni=1/(rho*e*(ue+uh))
ni=1/(rho*e*(ue+uh)); // calculation of intrinsic
   concentration
ne=Nd; // calculation of electron concentration
printf('\nThe electron concentration is \tne=%1.1E /
   m^3',ne);
nh=ni^2/Nd; // calculation of hole concentration
printf('\nThe hole concentration is \tnh=%1.1E /m^3'
   ,nh);
sigma=ne*ue*e+nh*uh*e; // calculation of
   conductivity
rho=1/sigma; // calculation of resistivity
printf('\nThe resistivity of the specimen  is \t%.2f
    ohm-m',rho);
```

Scilab code Exa 9.11 Find the conductivity of p type Ge crystal

```
// chapter 9
// example 9.11
// Find the conductivity of p-type Ge crystal
// page 276-277
clear;
clc;
//given
uh=1900; // in cm^2/V-s (mobility of electron)
Na=2E17;// in /m^3 (acceptor doping concentration)
e=1.6E-19; // in C(charge of electron)
//calculate
uh=uh*1E-4; // changing unit from cm^2/V-s to m^2/V-
   s
```

```
13 Na=Na*1E6; // changing unit from 1/cm^3 to 1/m^3
14 nh=Na; // hole concentration
15 // since sigma=ne*ue*e+nh*uh*e and nh>>ne
16 // therefore sigma=nh*uh*e
17 sigma=nh*uh*e; // calculation of conductivity
18 printf('\nThe conductivity of p-type Ge crystal is \
     t%.f  /ohm-m',sigma);
19 // Note: there is slight variation in the answer due
     to round off calculation
```

Scilab code Exa 9.12 Find the diffusion coefficient of electron in silicon

```
1 // chapter 9
2 // example 9.12
3 // Find the diffusion co-efficient of electron in
     silicon
4 // page 277
5 clear;
6 clc;
7 //given
8 ue=0.19; // in m^2/V-s (mobility of electron)
9 T=300; // in K (temperature)
10 k=1.38E-23; // in J/K (Boltzmann s constant)
11 e=1.6E-19; // in C(charge of electron)
12 //calculate
13 Dn=ue*k*T/e; // calculation of diffusion co-
     efficient
14 printf('\nThe diffusion co-efficient of electron in
     silicon is \tDn=%1.1E m^2/s',Dn);
```

Scilab code Exa 9.13 Find the probability of occupation of lowest level in conduction band

```
// chapter 9
// example 9.13
// Find the probability of occupation of lowest
   level in conduction band
// page 277-278
clear;
clc;
//given
Eg=0.4; // in eV (Band gap of semiconductor)
k=1.38E-23; // in J/K (Boltzmann s constant)
T1=0; // in degree Celcius (first temperature)
T2=50; // in degree Celcius (second temperature)
T3=100; // in degree Celcius (third temperature)
e=1.602E-19; //in C (charge of electron)
// calculate
T1=T1+273; // changing temperature form Celcius to
   Kelvin
T2=T2+273; // changing temperature form Celcius to
   Kelvin
T3=T3+273; // changing temperature form Celcius to
   Kelvin
Eg=Eg*e; // changing unit from eV to Joule
//Using F_E=1/(1+exp(Eg/2*k*T))
F_E1=1/(1+exp(Eg/(2*k*T1))); // calculation of
   probability of occupation of lowest level at 0
   degree Celcius
F_E2=1/(1+exp(Eg/(2*k*T2))); // calculation of
   probability of occupation of lowest level at 50
   degree Celcius
F_E3=1/(1+exp(Eg/(2*k*T3))); // calculation of
   probability of occupation of lowest level at 100
   degree Celcius
printf('\nThe probability of occupation of lowest
   level in conduction band is\n\n');
printf('\t\t at 0 degree Celcius, F_E=%1.3E eV\n',
   F_E1);
printf('\t\t at 50 degree Celcius, F_E=%1.2E eV\n',
   F_E2);
```

```
26 printf('\t\t at 100 degree Celcius, F_E=%1.3E eV',
     F_E3);
```

Scilab code Exa 9.14 Find the ratio of conductivity at 600K and at 300K

```
1  // chapter 9
2  // example 9.14
3  // Find the ratio of conductivity at 600K and at 300
     K
4  // page 278
5  clear;
6  clc;
7  //given
8  Eg=1.2; // in eV (Energy band gap)
9  k=1.38E-23; // in J/K (Boltzmann s constant)
10 T1=600, T2=300; // in K (two temperatures)
11 e=1.6E-19; // in C (charge of electron)
12 // calculate
13 Eg=Eg*e; // changing unit from eV to Joule
14 // since sigma is proportional to exp(-Eg/(2*k*T))
15 // therefore ratio=sigma1/sigma2=exp(-Eg/(2*k*((1/T1
     )-(1/T2))));
16 ratio= exp((-Eg/(2*k))*((1/T1)-(1/T2))); //
     calculation of ratio of conductivity at 600K and
     at 300K
17 printf('\nThe ratio of conductivity at 600K and at
     300K is \t%1.2E',ratio);
```

Scilab code Exa 9.15 Find the electron and hole densities and conductivity and the resistance

```
1  // chapter 9
2  // example 9.15
```

```
// Find the electron and hole densities and
   conductivity and the resistance
// page 278-279
clear;
clc;
//given
ue=0.39; // in m^2/V-s (mobility of electron)
n=5E13;//  number of donor atoms
ni=2.4E19; // in atoms/m^3 (intrinsic carrier
   density)
l=10; // in mm (length of rod)
a=1; // in mm (side of square cross-section)
e=1.6E-19;// in C (charge of electron)
//calculate
l=l*1E-3; // changing unit from mm to m
a=a*1E-3; // changing unit from mm to m
A=a^2; // calculation of cross-section area
Nd=n/(l*A); // calculation of donor concentration
ne=Nd; // calculation of electron density
nh=ni^2/Nd; // calculation of hole density
printf('\nThe electron density is \tne=%1.0E /m^3',
   ne);
printf('\nThe hole density is \tnh=%1.2E /m^3',nh);
// since sigma=ne*e*ue+nh*e*ue and since ne>>nh
// therefore sigma=ne*e*ue
sigma=ne*e*ue; // calculation of conductivity
printf('\nThe conductivity is \t%.f /ohm-m',sigma);
rho=1/sigma; // calculation of resistivity
R=rho*l/A; // calculation of resistance
printf('\nThe resistance is \tR=%.f ohm',R);
```

Scilab code Exa 9.16 Find the mobility and density

```
// chapter 9
// example 9.16
```

```
// Find the mobility and density
// page 279
clear;
clc;
//given
RH=3.66E-4; // in m^3/C (Hall coefficient)
rho=8.93E-3; // in ohm-m (resistivity)
e=1.6E-19; // in C (charge of electron)
// calculate
u=RH/rho; // calculation of mobility
n=1/(RH*e); // calculation of density
printf('\nThe mobility is \tu=%.4f m^2/(V-s)',u);
printf('\nThe density is \tn=%1.1E /m^3',n);
```

Scilab code Exa 9.17 Find the mobility and density of charge carrier

```
// chapter 9
// example 9.17
// Find the mobility and density of charge carrier
// page 279-280
clear;
clc;
//given
RH=3.66E-4; // in m^3/C (Hall coefficient)
rho=8.93E-3; // in ohm-m (resistivity)
e=1.6E-19; // in C (charge of electron)
// calculate
nh=1/(RH*e); // calculation of density of charge
   carrier
uh=1/(rho*nh*e); // calculation of mobility of
   charge carrier
printf('\nThe density of charge carrier is \tnh=%1.4
   E /m^3',nh);
printf('\nThe mobility of charge carrier is \tuh=%.3
   f m^2/(V-s)',uh);
```

Chapter 10

Superconductivity

Scilab code Exa 10.1 Calculate magnitude of critical magnetic field

```
1 //chapter 10
2 //example 10.1
3 //Calculate magnitude of critical magnetic field
4 //page 313
5 clear;
6 clc;
7 //given
8 Tc=7.2; // in K (critical temperature)
9 T=5; // in K (given temperature)
10 H0=6.5E3; // in A/m (critical magnetic field at 0K)
11 //calculate
12 Hc=H0*(1-(T/Tc)^2); // calculation of magnitude of
     critical magnetic field
13 printf('\nThe magnitude of critical magnetic field
     is \tHc=%1.3E A/m',Hc);
```

Scilab code Exa 10.2 Calculate critical current value

```
1 //chapter 10
2 //example 10.2
3 //Calculate critical current value
4 //page 313
5 clear;
6 clc;
7 //given
8 r=0.02; // in m (radius of ring)
9 Hc=2E3; // in A/m (critical magnetic field at 5K)
10 pi=3.14; // value of pi used in the solutiion
11 //calculate
12 Ic=2*pi*r*Hc; // calculation of critical current
      value
13 printf('\nThe critical current value is \tIc=%.1f A'
      ,Ic);
```

Scilab code Exa 10.3 calculate isotropic mass

```
1 //  chapter 10
2 // example 10.3
3 // calculate isotropic mass at 5.1K
4 // page 313
5 clear;
6 clc;
7 // given
8 M1=199.5; // in amu (isotropic mass at 5K)
9 T1=5; // in K (first critical temperature)
10 T2=5.1; // in K (second critical temperature)
11 //calculate
12 // since Tc=C*(1/sqrt(M)
13 // therefore T1*sqrt(M1)=T2*sqrt(M2)
14 // therefore we have M2=(T1/T2)^2*M1
15 M2=(T1/T2)^2*M1; //calculation of isotropic mass at
      5.1K
16 printf('\nThe isotropic mass at 5.1K is \t M2=%.3f a
```

```
.m.u.',M2);
```

Scilab code Exa 10.4 calculate transition temperature

```
1 //  chapter 10
2 // example 10.4
3 // calculate transition temperature
4 // page 314
5 // given
6 clear;
7 clc;
8 T=6; // in K (given temperature)
9 Hc=5E3; // in A/m (critical magnetic field at 5K)
10 H0=2E4; // in A/m (critical magnetic field at 0K)
11 //calculate
12 // since Hc=H0*(1-(T/Tc)^2)
13 // therefor we have Tc=T/sqrt(1-(Hc/H)^2)
14 Tc=T/sqrt(1-(Hc/H0)); // calculation of transition
   temperature
15 printf('\nThe transition temperature is \tTc=%.3f K'
   ,Tc);
16 // Note: answer in the book is wrong due to
   calculation mistake
```

Scilab code Exa 10.5 calculate critical current at 5K

```
1 //  chapter 10
2 // example 10.5
3 // calculate  critical current at 5K
4 // page 314
5 // given
6 clear;
7 clc;
```

```
T=5; // in K (given temperature)
d=3; // in mm (diameter of the wire)
Tc=8; // in K (critical temperature for Pb)
H0=5E4; // in A/m (critical magnetic field at 0K)
pi=3.14; // value of pi used in the solution
//calculate
 Hc=H0*(1-(T/Tc)^2); // calculation of critical
    magnetic field at 5K
printf('\nThe critical magnetic field at 5K is \tHc=
   %1.3E A/m',Hc);
r=(d*1E-3)/2; // calculation of radius in m
Ic=2*pi*r*Hc; // calculation of critical current at
   5K
printf('\nThe critical current at 5K is \tIc=%.4f A'
   ,Ic);
//Note: there is slight variation in the answer due
   to round off
```

Scilab code Exa 10.6 calculate frequency of EM waves

```
//  chapter 10
// example 10.6
// calculate  frequency of EM waves
// page 314
clear;
clc;
// given
V=8.50; // in micro V (voltage across Josephson
   junction )
e=1.6E-19; // in C (charge of electron)
h=6.626E-34; // in J/s (Planck s constant)
//calculate
V=V*1E-6; // changing unit from V to microVolt
 v=2*e*V/h; // calculation of frequency of EM waves
printf('\nThe frequency of EM waves is \tv=%1.3E Hz'
```

```
    ,v);
15 // Note: the answer in the book is wrong due to
    calculation misatke
```

Scilab code Exa 10.7 calculate transition temperature of the isotopes

```
1 //  chapter 10
2 // example 10.7
3 // calculate transition temperature of the isotopes
4 // page 315
5 clear;
6 clc;// given
7 M1=200.59; // in amu (average atomic mass  at 4.153K
    )
8 Tc1=4.153; // in K (first critical temperature)
9 M2=204; // in amu (average atomic mass of isotopes)
10 //calculate
11 // since Tc=C*(1/sqrt(M)
12 // therefore T1*sqrt(M1)=T2*sqrt(M2)
13 // therefore we have Tc2=Tc1*sqrt(M1/M2)
14 Tc2=Tc1*sqrt(M1/M2); //calculation of transition
    temperature of the isotopes
15 printf('\nThe transition temperature of the isotopes
    is \t Tc2=%.3f K',Tc2);
```

Chapter 12

Fibre Optics

Scilab code Exa 12.1 calculate fractional index change for a given optical fibre

```
1 //  chapter 12
2 // example 12.1
3 // calculate fractional index change for a given
     optical fibre
4 // page 360
5 clear;
6 clc;
7 // given
8 u1=1.563; // refractive index of core
9 u2=1.498; // refractive index of cladding
10 //calculate
11 d=(u1-u2)/u1; // calculation of fractional index
     change
12 printf('\nThe fractional index change for a given
     optical fibre is %.4f',d);
```

Scilab code Exa 12.2 calculate numerical aperture and the acceptance angle of an optical fibre

```
1 //  chapter 12
2 // example 12.2
3 // calculate numerical aperture and the acceptance
     angle of an optical fibre
4 // page 360
5 clear;
6 clc;
7 // given
8 u1=1.55; // refractive index of core
9 u2=1.50; // refractive index of cladding
10 //calculate
11 d=(u1-u2)/u1; // calculation of fractional index
     change
12 NA=u1*sqrt(2*d); // calculation of numerical
     aperture
13 printf('\nThe numerical aperture of the fibre is \
     tNA=%.3f',NA);
14 theta=asind(NA); // calculation of acceptance angle
15 printf('\nThe acceptance angle of the optical fibre
     is \t%.1f degree',theta);
```

Scilab code Exa 12.3 calculate the acceptance angle of an optical fibre

```
1 //  chapter 12
2 // example 12.3
3 // calculate the acceptance angle of an optical
     fibre
4 // page 360
5 // given
6 clear;
7 clc;
8 u1=1.563; // refractive index of core
9 u2=1.498; // refractive index of cladding
10 //calculate
11 NA=sqrt(u1^2-u2^2); // calculation of numerical
```

```
    aperture
12 printf('\nThe numerical aperture of the fibre is \
    tNA=%.4f',NA);
13 theta=asind(NA); // calculation of acceptance angle
14 printf('\nThe acceptance angle of the optical fibre
    is \t%.2f degree',theta);
```

Scilab code Exa 12.4 calculate refractive index of material of the core

```
1 //  chapter 12
2 // example 12.4
3 // calculate refractive index of material of the
    core
4 // page 360-361
5 clear;
6 clc;
7 // given
8 NA=0.39; //numerical aperture of the optical fibre
9 d=0.05; // difference in the refractive index of the
    material of the core and cladding
10 //calculate
11 // since NA=u1*sqrt(2*d)
12 //we have u1=NA/sqrt(2*d)
13 u1= NA/sqrt(2*d); // calculation of refractive index
    of  material of the core
14 printf('\nThe refractive index of  material of the
    core is \tu1=%.3f',u1);
```

Scilab code Exa 12.5 calculate numerical aperture acceptance angle and the critical angle of the optical fibre

```
1 //  chapter 12
2 // example 12.5
```

```
// calculate numerical aperture,acceptance angle and
    the critical angle of the optical fibre
// page 361
clear;
clc;
// given
u1=1.50; // refractive index of core
u2=1.45; // refractive index of cladding
//calculate
d=(u1-u2)/u1; // calculation of fractional index
   change
NA=u1*sqrt(2*d); // calculation of numerical
   aperture
printf('\nThe numerical aperture of the fibre is \
   tNA=%.3f',NA);
theta_0=asind(NA); // calculation of acceptance
   angle
printf('\nThe acceptance angle of the optical fibre
   is \t%.2f degree',theta_0);
theta_c=asind(u2/u1); // calculation of critical
   angle
printf('\nThe critical angle of the optical fibre is
    \t%.1f degree',theta_c);
```

Scilab code Exa 12.6 calculate refractive index of the core and cladding material of a fibre

```
//  chapter 12
// example 12.6
// calculate refractive index of the core and
   cladding material of a fibre
// page 361
clear;
clc;
// given
```

```
NA=0.33; // numerical aperture
d=0.02; // difference in the refractive index of the
    core and cladding of the material
//calculate
// since NA=u1*sqrt(2*d)
// therefore we have
u1=NA/sqrt(2*d); // calculation of refractive index
   of the core
// since d=(u1-u2)/u2
// therefore we have
u2=(1-d)*u1; // calculation of refractive index of
   the cladding
printf('\nThe refractive index of the core is \tu1=%
   .1f',u1);
printf('\nThe refractive index of the cladding is \
   tu2=%.3f',u2);
// Note: In the question, it is given that NA=0.33
   but in the book NA=0.22 has been used in the
   solution. That's why answer in the book is
   different from that of generated from the code
```

Scilab code Exa 12.7 calculate numerical aperture and acceptance angle of the symmetrical fibre

```
//  chapter 12
// example 12.7
// calculate numerical aperture and acceptance angle
    of the symmetrical fibre
// page 361
clear;
clc;
// given
u1=3.5; // refractive index of core
u2=3.45; // refractive index of cladding
u0=1; // refractive index of the air
```

```
11 //calculate
12 NA=sqrt(u1^2-u2^2); // calculation of numerical
      aperture
13 NA=NA/u0;
14 printf('\nThe numerical aperture of the fibre is \
      tNA=%.2f',NA);
15 alpha=asind(NA); // calculation of acceptance angle
16 printf('\nThe acceptance angle of the optical fibre
      is \t%.2f degre',alpha);
```

Scilab code Exa 12.8 calculate numerical aperture and acceptance angle of the symmetrical fibre

```
1 //  chapter 12
2 // example 12.8
3 // calculate numerical aperture and acceptance angle
      of an optical fibre
4 // page 361-362
5 clear;
6 clc;
7 // given
8 u1=1.48; // refractive index of core
9 u2=1.45; // refractive index of cladding
10 //calculate
11 NA=sqrt(u1^2-u2^2); // calculation of numerical
      aperture
12 printf('\nThe numerical aperture of the fibre is \
      tNA=%.3f',NA);
13 theta=asind(NA); // calculation of acceptance angle
14 printf('\nThe acceptance angle of the optical fibre
      is \t%.2f degree',theta);
15 // Note: there is slight variation in the answer due
      to round off
```