

Scilab Textbook Companion for
Applied Physics
by M. Arumugam¹

Created by
Mathews George
B.ts
Others
M.g
College Teacher
None
Cross-Checked by
Reshma

March 29, 2016

¹Funded 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: M. Arumugam

Publisher: Anuradha Publications

Edition: 1

Year: 2015

ISBN: 9788189638726

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	4
1 Bonding in Solids and Crystal Structures	5
2 Crystal Planes and Xray Diffraction	12
3 Defects In Solids	21
4 Electron Theory of Metals	29
5 Dielectric Properties and Magnetic Properties	36
6 Semiconductors and Superconductivity	43
7 Lasers	57
8 Fiber Optics	61

List of Scilab Codes

Exa 1.1	Calculation of youngs modulus	5
Exa 1.2	Find the Effective charge	5
Exa 1.3	Find the Cohesive energy	6
Exa 1.4	Find the Coulomb energy	6
Exa 1.5	Find the Distance of separation	7
Exa 1.6	Find the Bond Energy	7
Exa 1.7	Find the density	8
Exa 1.8	Find the radius of sphere	9
Exa 1.9	increase of density or the decrease of volume	9
Exa 1.10	spacing between the nearest neighbouring ions	10
Exa 1.11	lattice constant	11
Exa 1.12	Density of iron	11
Exa 2.1	Number of atoms per unit area	12
Exa 2.2	Surface area of surfaces	12
Exa 2.3	Calculate ratio	13
Exa 2.4	Calcutate length	14
Exa 2.5	Calcutate two lengths	14
Exa 2.7	Calcutate lengths of two spheres	15
Exa 2.8	length by diffraction given by angles	16
Exa 2.9	Given wavelength calculate n	16
Exa 2.10	Calcutate length	17
Exa 2.11	Calculate the angle	17
Exa 2.12	Calculate h and k values	18
Exa 2.13	Calculate wavelength and energy	19
Exa 2.14	Calculate wavelength and angle	19
Exa 3.1	The number of vacancies per kilomole of copper	21
Exa 3.2	Fraction of vacancies at 1000 degree	22
Exa 3.3	The concentration of Schottky defects	22

Exa 3.4	amount of climb down by the dislocation	23
Exa 3.5	Velocity and wavelength	23
Exa 3.6	Momentum and de Brolie wavelength	24
Exa 3.7	wavelength	24
Exa 3.8	Wavelength	25
Exa 3.9	The permitted electron energies	25
Exa 3.10	Calculate delta	26
Exa 3.11	Calculate energy difference	26
Exa 3.12	Calculate energy	27
Exa 3.13	Integration	27
Exa 4.1	energy difference	29
Exa 4.2	Calculate temperature	30
Exa 4.3	Calculate fermi energy	30
Exa 4.4	Find energy	31
Exa 4.5	Number of states	31
Exa 4.6	mean free collision time	32
Exa 4.7	The mean free collision time	32
Exa 4.8	Relaxation time	33
Exa 4.9	Temperature coefficient of resistivity	33
Exa 4.10	Increase in resistivity in copper	34
Exa 5.1	insulation resistance	36
Exa 5.2	polarisability of He	37
Exa 5.3	total dipole moment	37
Exa 5.4	the complex polarizability	38
Exa 5.5	temperature rise	38
Exa 5.6	magnetic field at the centre	39
Exa 5.7	intensity of magnetisation	40
Exa 5.8	number of Bohr magnetons	40
Exa 5.9	average magnetic moment	41
Exa 5.10	hysteresis loss	41
Exa 6.1	number of electron hole pairs	43
Exa 6.2	intrinsic conductivity	44
Exa 6.3	charge carrier density	45
Exa 6.4	conductivity during donor impurity	45
Exa 6.5	conductivity	46
Exa 6.6	Concentration	47
Exa 6.7	Current density	47
Exa 6.8	Position of fermi level	48

Exa 6.9	Conductivity of intrinsic germanium at 300K	48
Exa 6.10	Conductivity	49
Exa 6.11	The electron concentration	50
Exa 6.12	resistance	50
Exa 6.13	Calculate Dn and Dp	50
Exa 6.14	Electrical Conductivity	51
Exa 6.15	Current density	52
Exa 6.16	Drift velocity	52
Exa 6.17	critical field	53
Exa 6.18	Frequency of generated microwaves	53
Exa 6.19	Penetration depth	54
Exa 6.20	Calculate wavelength	55
Exa 6.21	Critical current density	55
Exa 6.22	New critical temperature for mercury	56
Exa 7.1	Calcutate Divergence	57
Exa 7.2	Relative Population	57
Exa 7.3	Power density	58
Exa 7.4	Wavelength	59
Exa 7.5	Band gap	59
Exa 1.1	Critical angle	61
Exa 8.2	No of modes propogated inside the fiber	61
Exa 8.3	Numerical aperture	62
Exa 8.4	Numerical aperture	62
Exa 8.5	maximum no of modes propogating through fiber	63
Exa 8.6	Number of modes	63
Exa 8.7	Cutoff Wavellength	64
Exa 8.8	Maximum core radius	64
Exa 8.9	Acceptance angle	65
Exa 8.10	Numerical aperture and Acceptance angle	65
Exa 8.11	Core radius	66
Exa 8.12	Total dist travelled by light over one metre of fiber	66
Exa 8.13	No of modes	67
Exa 8.14	ignal attention per unit length	67
Exa 8.15	Bandwidth length product	68

Chapter 1

Bonding in Solids and Crystal Structures

Scilab code Exa 1.1 Calculation of youngs modulus

```
1 //Initialisation of variables
2 clc
3 //Variable declaration
4 a=7.68*10**-29;
5 r0=2.5*10**-10; //radius (m)
6
7 //Calculation
8 b=a*(r0**8)/9;
9 y=((-2*a*r0**8)+(90*b))/r0**11;
10 E=y/r0; //young's modulus (Pa)
11
12 //Result
13
14 printf('youngs modulus is %0.2f GPa',(E/10^9))
```

Scilab code Exa 1.2 Find the Effective charge


```

1 //Initialisation of variables
2 clc
3
4 d=((1.98)*10**-29)*1/3;           //dipole moment
5 b=(0.92);                         //bond length
6 EC=d/(b*10**-10);                 //Effective charge
7
8 //Result
9 printf('Effective charge =%0.2f *10**-29 coulomb',((
    EC*10**19)))

```

Scilab code Exa 1.3 Find the Cohesive energy

```

1 //Initialisation of variables
2 clc
3
4 A=1.748                             //Madelung Constant
5 N=6.02*10**26                       //Avagadro Number
6 e=1.6*10**-19
7 n=9.5
8 r=(0.324*10**-9)*10**3
9 E=8.85*10**-12
10 //Calculations
11 U=((N*A*(e)**2)/(4*%pi*E*r))*(1-1/n) //
    Cohesive energy
12
13 //Result
14 printf('Cohesive energy =%0.2f *10**3 kJ/kmol \n',(U
    /10**3))
15 printf('//Answer varies due to rounding of numbers')

```

Scilab code Exa 1.4 Find the Coulomb energy

```

1 //variable declaration
2 I=5; //Ionisation energy
3 A=4; //Electron Affinity
4 e=(1.6*10**-19)
5 E=8.85*10**-12 //epsilon constant
6 r=0.5*10**-19 //dist between A and B
7
8 //Calculations
9 C=-(e**2/(4*%pi*E*r*e))/10**10 //Coulomb energy
10 E_c=I-A+C //Energy
    required
11
12 //Result
13 printf('Coulomb energy =%0.2f eV\n',C)
14 printf('Energy required =%0.2f eV',E_c')

```

Scilab code Exa 1.5 Find the Distance of separation

```

1 //variable declaration
2 I=5.14; //Ionization energy
3 A=3.65; //Electron Affinity
4 e=(1.6*10**-19);
5 E=8.85*10**-12;
6 //calculations
7 E_c=I-A //Energy required
8 r=e**2/(4*%pi*E*E_c*e) //Distance of separation
9
10 //Result
11 printf('Energy required=%0.2f eV \n",E_c)
12 printf('Distance of separation =%0.2f Angstrom",r
    /10**-10)

```

Scilab code Exa 1.6 Find the Bond Energy

```

1 //variable declaration
2 I=5.14; //Ionization energy
3 A=3.65; //Electron Affinity
4 e=(1.6*10**-19);
5 E=8.85*10**-12;
6 r=236*10**-12;
7
8 //Calculations
9 E_c=I-A //Energy required
10 C=-(e**2/(4*%pi*E*r*e)) //Potential energy in
    eV
11 BE=-(E_c+C) //Bond Energy
12 //Result
13 printf('Energy required= %0.2f eV\n',E_c)
14 printf('Energy required =%0.1f eV\n',C)
15 printf('Bond Energy =%0.2f eV',BE)

```

Scilab code Exa 1.7 Find the density

```

1 //variable declaration
2 d=2.351 //bond length
3 N=6.02*10**26 //Avagadro number
4 n=8 //number of atoms in unit
    cell
5 A=28.09 //Atomin mass of silicon
6 m=6.02*10**26 //1mole
7
8 //Calculations
9 a=(4*d)/sqrt(3)
10 p=(n*A)/((a*10**-10)*m) //density
11
12 //Result
13 printf('a=%0.2fAngstorm\n',a)
14 printf('density =%0.2f kg/m**3\n',(p*10**16))
15 printf("//Answer given in the textbook is wrong")

```

Scilab code Exa 1.8 Find the radius of sphere

```
1 //Variable declaration
2
3
4 //Calculation
5 a1=4/sqrt(3);
6 R1=(a1/2)-1;           //radius of largest sphere
7 a2=4/sqrt(2);
8 R2=(a2/2)-1;         //maximum radius of sphere
9
10 //Result
11 printf('radius of largest sphere is %f*r\n',R1)
12 printf('maximum radius of sphere is %f*r',R2 )
```

Scilab code Exa 1.9 increase of density or the decrease of volume

```
1 //variable declaration
2 r1=1.258           //Atomic radius of BCC
3 r2=1.292           //Atomic radius of FCC
4
5 //calculations
6 a1=(4*r1)/sqrt(3) //in BCC
7 b1=((a1)**3)*10**-30 //Unit cell volume
8 v1=(b1)/2         //Volume occupied by
   one atom
9 a2=2*sqrt(2)*r2   //in FCC
10 b2=(a2)**3*10**-30 //Unit cell
   volume
11 v2=(b2)/4         //Volume occupied by
   one atom
```

```

12 v_c=((v1)-(v2))*100/(v1) //Volume Change in %
13 d_c=((v1)-(v2))*100/(v2) //Density Change in %
14
15 //Results
16 printf('a1=%0.3 f Angstrom\n\n',(a1))
17 printf('Unit cell volume =a1**3 =%0.3 f *10**-30 m
    **3\n',((b1)/10**-30))
18 printf('Volume occupied by one atom =%0.2 f *10**-30
    m**3\n',(v1/10**-30))
19 printf('a2=%0.2 f\n Angstorm\n',(a2))
20 printf('Unit cell volume =a2**3 =%0.3 f *10**-30 m
    **3\n',((b2)/10**-30))
21 printf('Volume occupied by one atom =%0.3 f*10**-30 m
    **3\n',(v2/10**-30))
22 printf('Volume Change in percentage =%0.3 f\n',(v_c)
    )
23 printf('Density Change in percentage =%0.3 f\n',(d_c)
    )
24 printf('Thus the increase of density or the decrease
    of volume is about 0.5 percentage")

```

Scilab code Exa 1.10 spacing between the nearest neighbouring ions

```

1 //variable declaration
2 n=4
3 M=58.5 //Molecular wt. of NaCl
4 N=6.02*10**26 //Avagadro number
5 rho=2180 //density
6
7 //Calculations
8 a=((n*M)/(N*rho))**(1/3)
9 s=a/2
10
11 //Result
12 printf('a=%0.3 f*10**-9 metre\n",(a/10**-9))

```

```
13 printf('spacing between the nearest neighbouring
    ions =%0.3f nm',(s/10**-9))
```

Scilab code Exa 1.11 lattice constant

```
1 //variable declaration
2 n=4
3 A=63.55 //Atomic wt. of NaCl
4 N=6.02*10**26 //Avagadro number
5 rho=8930 //density
6
7 //Calculations
8 a=((n*A)/(N*rho))**(1/3) //Lattice Constant
9
10 //Result
11 printf('lattice constant , a=%0.3f nm',(a*10**9))
```

Scilab code Exa 1.12 Density of iron

```
1 //variable declaration
2 r=0.123 //Atomic radius
3 n=4
4 A=55.8 //Atomic wt
5 a=2*sqrt(2)
6 N=6.02*10**26 //Avagadro number
7
8 //Calculations
9 rho=(n*A)/((a*r*10**-9)**3*N)
10
11 //Result
12 printf('Density of iron =%0.3fkg/m**-3',rho)
```

Chapter 2

Crystal Planes and Xray Diffraction

Scilab code Exa 2.1 Number of atoms per unit area

```
1  clc
2
3  //Variable declaration
4  a=mulf('2', 'R')
5
6  //Results
7  printf('i.Number of atoms per unit area of (100)
   plane= 1/(%d*R**2) ",2**2)
8  printf(' \nii.Number of atoms per unit area of (110)
   plane=%f*R**2",2**2/(sqrt(2)))
9  printf(' \niii.Number of atoms per unit area of (111)
   plane=%f*R**2",2**2/(sqrt(3)))
```

Scilab code Exa 2.2 Surface area of surfaces

```
1  clc
```

```

2
3
4
5 //Variable declaration
6 a=3.61*10**-7
7 BC=sqrt(2)/2
8 AD=(sqrt(6))/2
9 //Result
10 printf('i.Surface area of the face ABCD =%0.3f
      *10**-14 mm**2\n',(a**2*10**14))
11 printf('ii.Surface area of plane (110) =%0.3f*10**13
      atoms/mm**2\n',((2/(a*sqrt(2)*a)/10**13)))
12 printf('iii.Surface area of pane(111)=%0.3f*10**13
      atoms/mm**2", (2/(BC*AD*a**2)*10**-13))

```

Scilab code Exa 2.3 Calculate ratio

```

1 h1=1
2 k1=0
3 l1=0
4 h2=1
5 k2=1
6 l2=0
7 h3=1
8 k3=1
9 l3=1
10 a=1
11
12 //Calculations
13 d1=a/(sqrt(h1**2+k1**2+l1**2))
14 d2=a/(sqrt(h2**2+k2**2+l2**2))
15 d3=a/(sqrt(h3**2+k3**2+l3**2))
16
17 //Result
18 printf('d1 =%0.2f\n',d1 )

```



```
19 printf( 'd2 =%0.2f\n' ,(d2))
20 printf( 'd3 =%0.2f\n' ,(d3))
21 printf( 'd1:d2:d3 =%0.2f:%0.2f:%0.2f' ,d1 ,( d2) ,d3)
```

Scilab code Exa 2.4 Calculatate length

```
1
2 clc
3
4
5
6 //Variable declaration
7 h=2
8 k=2
9 l=0
10 a=450
11
12 //Calculations
13 d=a/(sqrt(h**2+k**2+l**2))
14
15 //Result
16 printf( 'd(220) =%0.3fpm\n' ,(d))
```

Scilab code Exa 2.5 Calculatate two lengths

```
1
2 clc
3
4
5
6 //Variable declaration
7 a=3.615
8 r=1.278
```

```

9 h=1
10 k=1
11 l=1
12
13 // Calculations
14 a=(4*r)/sqrt(2)
15 d=a/(sqrt(h**2+k**2+l**2))
16
17 // Result
18 printf('a =%0.3fAngstroms\n',(a))
19 printf('d =%0.3fAngstroms\n',(d))

```

Scilab code Exa 2.7 Calculate lengths of two spheres

```

1 clc
2
3
4
5 // Variable declaration
6 n=1
7 lamda=1.54
8 theta=32*%pi/180
9 h=2
10 k=2
11 l=0
12
13 // Calculations
14 d=(n*lamda*10**-10)/(2*sin(theta)) //derived from
    2dsin(theta)=n*l
15 a=d*(sqrt(h**2+k**2+l**2))
16
17 // Results
18 printf('d =%0.3f *10**-10 m\n',(d*10**10))
19 printf('a =%0.3f *10**-10 m\n',(a*10**10))

```

Scilab code Exa 2.8 length by diffraction given by angles

```
1  clc
2
3
4
5  //Variable declaration
6  lamda=0.58
7  theta1=6.45*%pi/180
8  theta2=9.15*%pi/180
9  theta3=13*%pi/180
10
11 //Calculations
12 dbyn1=lamda/(2*(sin(theta1)))
13 dbyn2=lamda/(2*sin(theta2))
14 dbyn3=lamda/(2*sin(theta3))
15
16 //Results
17 printf('i. d/n =%0.3 f Angstroms\n', (dbyn1))
18 printf('ii. d/n =%0.3 f Angstroms\n', (dbyn2))
19 printf('iii. d/n =%0.3 f Angstroms\n', (dbyn3))
```

Scilab code Exa 2.9 Given wavelength calculate n

```
1  clc
2
3
4
5  //Variable declaration
6  d=1.18
7  theta=90*%pi/180
8  lamda=1.540
```

```

9
10 // Calculations
11 n=(2*d*sin(theta))/lamda
12
13 // Result
14 printf('n =%0.3 f \n', (n))

```

Scilab code Exa 2.10 Calculate length

```

1 clc
2
3
4
5 // Variable declaration
6 lamda=0.58
7 theta=9.5*%pi/180
8 n=1
9 d=0.5 // d200=a/sqrt(2**2+0**2+0**2)=0.5 a
10 // Calculations
11 a=n*lamda/(2*d*sin(theta)) // 2*d*sin(theta)=n*
    lamda
12
13 // Result
14 printf('a =%0.3 f Angstroms\n', (a))

```

Scilab code Exa 2.11 Calculate the angle

```

1
2 clc
3
4
5
6 // Variable declaration

```

```

7 lamda=0.842
8 n1=1
9 q=(8+(35/60))*(%pi/180)
10 n2=3
11 d=1
12 // Calculations
13 //n*lamda=2*d*sin(theta)
14 //n1*0.842=2*d*sin(q)
15 //n3*0.842=2*d*sin(theta3)
16 //Dividing both the equations, we get
17 //((n2*lamda)/(n1*lamda))=2*d*sin(theta3)/2*d*sin(q)
18 theta3=asin(((n2*lamda)/(n1*lamda))*(2*d*sin(q))
    /(2*d))
19 d=theta3*180/%pi;
20 a_d=int(d);
21 a_m=(d-int(d))*60
22
23 // Result
24 printf('sin(theta3) =%0.3f %0.3f',a_d,a_m)

```

Scilab code Exa 2.12 Calculate h and k values

```

1 clc
2 //Variable declaration
3 a=3.16
4 lamda=1.54
5 n=1
6 theta=20.3*%pi/180
7
8 // Calculations
9 d=(n*lamda)/(2*sin(theta))
10 x=a/d //let sqrt(h**2+k
    **2+l**2)=x
11
12 // Result

```

```

13 printf('d =%0.3f Angstroms\n',(d))
14 printf('sqrt(h**2+k**2+l**2) =%0.3f \n',(x))
15 printf('Therefore , h**2+k**2+l**2 =sqrt(2)\n')
16 printf('h =1, k=1')

```

Scilab code Exa 2.13 Calculate wavelength and energy

```

1
2 //Variable declaration
3 n=4
4 A=107.87
5 rho=10500
6 N=6.02*10**26
7 h=1;
8 k=1;
9 l=1;
10 H=6.625*10**-34
11 e=1.6*10**-19
12 theta=(19+(12/60))*%pi/180
13 C=3*10**8
14 //Calculations
15 a=((n*A)/(rho*N))**(1/3)*10**10
16 d=a/sqrt(h**2+k**2+l**2)
17 lamda=2*d*sin(theta)
18 E=(H*C)/(lamda*10**-10*e)
19
20 //Result
21 printf('a =%0.3f Angstroms \n',(a))
22 printf('d =%0.3f Angstroms\n',(d))
23 printf('lamda =%0.3f Angstroms\n',(lamda))
24 printf('E =%0.3f *10**3 eV\n',(E/10**3))

```

Scilab code Exa 2.14 Calculate wavelength and angle

```
1
2 //Variable declaration
3 a=4.57
4 h=1
5 k=1
6 l=1
7 lamda=1.52
8 twotheta=33.5*%pi/180
9 r=5 //radius
10 //Calculations
11 d=a/(h**2+k**2+l**2)**(1/2)
12 sintheta=lamda/(2*d)
13 X=r/tan(twotheta)
14
15 //Result
16 printf('d =%0.3f Angstorms\n',(d))
17 printf('sin(theta)=%0.3f \n',(sintheta))
18 printf('X =%0.3f cm\n',(X))
```

Chapter 3

Defects In Solids

Scilab code Exa 3.1 The number of vacancies per kilomole of copper

```
1 //Variable declaration
2 N=6.023*10**26
3 deltaHv=120
4 B=1.38*10**-23
5 k=6.023*10**23
6
7 //Calculations
8 n0=0 // 0
   in denominator
9 n300=N*exp(-deltaHv*10**3/(k*B*300)) //The
   number of vacancies per kilomole of copper
10 n900=N*exp(-(deltaHv*10**3)/(k*B*900))
11
12 //Results
13 printf('at 0K, The number of vacancies per kilomole
   of copper is %0.3f' ,n0)
14 printf('at 300K, The number of vacancies per
   kilomole of copper is %0.3f *10**5\n',(n300
   /10**5))
15 printf('at 900K, The number of vacancies per
   kilomole of copper is %0.3f *10**19\n',(n900
```



```
/10**19))
```

Scilab code Exa 3.2 Fraction of vacancies at 1000 degree

```
1
2 //Variable declaration
3 F_500=1*10**-10
4
5 T1=500+273
6 T2=1000+273
7
8
9 //Calculations
10 lnx=log(F_500)*T1/T2;
11 x=exp(lnx)
12
13 printf('Fraction of vacancies at 1000 degrees C =%0
    .3f *10**-7\n',(x*10**7))
```

Scilab code Exa 3.3 The concentration of Schottky defects

```
1 //Variable declaration
2 a=(2*2.82*10**-10)
3 delta_Hs=1.971*1.6*10**-19
4 k=1.38*10**-23
5 T=300
6 e=2.718281
7 //Calculations
8 V=a**3 //Volume of unit
    cell of NaCl
9 N=4/V //Total number of
    ion pairs
10 n=N*e**-(delta_Hs/(2*k*T))
```

```

11
12 //Result
13 printf('Volume of unit cell of NaCl =%0.3f *10**-28
      m**3 \n',(V*10**28))
14 printf('Total number of ion pairs N =%0.3f *10**28\n
      ',(N/10**28))
15 printf('The concentration of Schottky defects per m
      **3 at 300K =%0.3f *10**11\n',(n/10**11))

```

Scilab code Exa 3.4 amount of climb down by the dislocation

```

1 //Variable declaration
2 N=6.023*10**23
3 delta_Hv=1.6*10**-19
4 k=1.38*10**-23
5 T=500
6 mv=5.55; //molar volume
7 x=2*10**-8; //number of cm in 1 angstrom
8
9 //Calculations
10 n=N*exp(-delta_Hv/(k*T))/mv
11 a=(n/(5*10**7*10**6))*x;
12
13 //Result
14 printf('The number that must be created on heating
      from 0 to 500K is n=%0.3f *10**12 per cm**3\n',(n
      /10**12)) //into cm**3
15 printf('As one step is 2 Angstroms, 5*10**7
      vacancies are required for 1cm')
16 printf('The amount of climb down by the dislocation
      is %0.3f cm',a*10**8)

```

Scilab code Exa 3.5 Velocity and wavelength

```

1 //Variable declaration
2 KE=10 //Kinetic Energy of neutron in keV
3 m=1.675*10**-27
4 h=6.625*10**-34
5 //Calculations
6 KE=10**4*1.6*10**-19 //in joule
7 v=((2*KE)/m)**(1/2) //derived from KE=1/2*m*v
   **2
8 lamda=h/(m*v)
9 //Results
10 printf('Velocity =%0.3 f *10**6 m/s \n ',(v/10**6))
11 printf('Wavelength =%0.3 f Angstorm \n ',(lamda
   *10**10))

```

Scilab code Exa 3.6 Momentum and de Brolie wavelength

```

1
2 //Variable declaration
3 E=2*1000*1.6*10**-19 //in joules
4 m=9.1*10**-31
5 h=6.6*10*10**-34
6
7 //Calculations
8 p=sqrt(2*m*E)
9 lamda= h/p
10
11 //Result
12 printf('Momentum%0.3 f \n ',(p*10**23))
13 printf('de Brolie wavelength =%0.3 f *10**-11 m \n ',
   ,(lamda*10**10))

```

Scilab code Exa 3.7 wavelength

```

1 //Variable declaration
2 M=1.676*10**-27           //Mass of neutron
3 m=0.025
4 v=1.602*10**-19
5 h=6.62*10**-34
6
7 //Calculations
8 mv=(2*m*v)**(1/2)
9 lamda=h/(mv*M**(1/2))
10
11 //Result
12 printf('wavelength =%0.3f  Angstorm \n ',(lamda
    *10**10))

```

Scilab code Exa 3.8 Wavelength

```

1 //Variable declaration
2 V=10000
3
4 //Calculation
5 lamda=12.26/sqrt(V)
6
7 //Result
8 printf('Wavelength =%0.3f  Angstorm ' ,lamda)

```

Scilab code Exa 3.9 The permitted electron energies

```

1 //Variable declaration
2 e=1.6*10**-19;           //charge of electron(coulomb)
3 L=10**-10                 //1Angstrom=10**-10 m
4 n1=1;
5 n2=2;
6 n3=3;

```

```

7 h=6.626*10**-34
8 m=9.1*10**-31
9 L=10**-10
10
11 // Calculations
12 E1=(h**2)/(8*m*L**2*e)
13 E2=4*E1
14 E3=9*E1
15 // Result
16 printf('The permitted electron energies =%0.3f *n
    **2 eV \n ',(E1))
17 printf('E1=%0.3f eV \n ',(E1))
18 printf('E2=%0.3f eV \n ',(E2))
19 printf('E3=%0.3f eV \n ',(E3))
20 printf('//Answer varies due to ing of numbers")

```

Scilab code Exa 3.10 Calculate delta

```

1 //Variable declaration
2 i=1*10**-10; //interval
3 L=10*10**-10; //width
4
5 // Calculations
6 si2=2*i/L;
7
8 // Result
9 printf('si**2 delta(x)=%0.3f ', si2)

```

Scilab code Exa 3.11 Calculate energy difference

```

1 //Variable declaration
2 nx=1
3 ny=1

```

```

4  nz=1
5  a=1
6  h=6.63*10**-34
7  m=9.1*10**-31
8
9  // Calculations
10 E1=h**2*(nx**2+ny**2+nz**2)/(8*m*a**2)
11 E2=(h**2*6)/(8*m*a**2)           //nx**2+ny**2+nz
    **2=6
12 diff=E2-E1
13 // Result
14 printf('E1 =%0.3 f *10**-37 Joule \n ',(E1*10**37))
15 printf('E2 =%0.3 f *10**-37 Joule \n ',(E2*10**37))
16 printf('E2-E1 =%0.3 f *10**-37 J \n ',(diff*10**37))

```

Scilab code Exa 3.12 Calculate energy

```

1  // Variable declaration
2  m=1.67*10**-27
3  a=10**-14
4  h=1.054*10**-34
5
6  // Calculations
7  E1=(1*pi*h)**2/(2*m*a**2)
8
9  // Result
10 printf('E1 =%0.3 f *10**-13 J \n ',(E1*10**13))

```

Scilab code Exa 3.13 Integration

```

1  // Variable declarations
2  k=1;
3

```

```
4 // Calculations
5
6 a=integrate('2*k*exp(-2*k*x)', 'x', 2/k, 3/k)
7 // Result
8 printf('a=%0.3f \n ', (a))
```

Chapter 4

Electron Theory of Metals

Scilab code Exa 4.1 energy difference

```
1
2
3 //Variable declaration
4 m=9.1*10**-31; //mass(kg)
5 nx=1;
6 ny=1
7 nz=1
8 n=6;
9 a=1; //edge(m)
10 h=6.63*10**-34; //planck's constant
11 k=1.38
12 //Calculation
13 E1=h**2*(nx**2+ny**2+nz**2)/(8*m*a**2);
14 E2=h**2*n/(8*m*a**2);
15 E=E2-E1; //energy difference(J)
16 T=(2*E2*10**37)/(3*k*10**-23)
17 //Result
18 printf('energy difference is%0.3f *10**-37 J \n ',(
    E*10**37))
19 printf('3/2*k*T = E2 =%0.3f *10**-37 J \n ',(E2
    *10**37))
```



```
20 printf('T =%0.3f *10**-14 K \n ',(T/10**23))
```

Scilab code Exa 4.2 Calculate temperature

```
1 //Variable declaration
2 y=1/100; //percentage of probability
3 x=0.5*1.6*10**-19; //energy(J)
4 k=1.38*10**-23; //boltzmann constant
5
6 //Calculation
7 xbykT=log((1/y)-1);
8 T=x/(k*xbykT); //temperature(K)
9
10 //Result
11 printf('temperature is %0.3f K ',int(T))
12 printf('answer varies due to ing off errors')
```

Scilab code Exa 4.3 Calculate fermi energy

```
1 //Variable declaration
2 d=970; //density(kg/m**3)
3 Na=6.02*10**26; //avagadro number
4 w=23; //atomic weight
5 m=9.1*10**-31; //mass(kg)
6 h=6.62*10**-34; //planck's constant
7
8 //Calculation
9 N=d*Na/w; //number of atoms/m**3
10 x=h**2/(8*m);
11 y=(3*N/%pi)**(2/3)
12 EF=x*y; //fermi energy(J)
13
14 //Result
```

```
15 printf('fermi energy is %0.3f eV \n ',(EF
    /(1.6*10**-19)))
```

Scilab code Exa 4.4 Find energy

```
1 //Variable declaration
2 kT=1;
3 E_EF=1;
4
5 //Calculations
6 p_E=1/(1+exp(E_EF/kT))
7
8 //Result
9 printf('p(E) =%0.3f \n ',(p_E))
```

Scilab code Exa 4.5 Number of states

```
1 //Variable declarations
2 m=9.1*10**-31
3 h=6.626*10**-34
4 Ef=3.1
5 Ef1=Ef+0.02
6 e=1.6*10**-19
7 //Calculations
8
9 N=integrate('%pi*((8*m)**(3/2))*(E**(1/2))*e
    *(3/2))/(2*(h**3))', 'E', Ef, Ef1)
10
11 //Result
12 printf('N =%0.3f *10**26 states \n ',(N*10**-26))
```

Scilab code Exa 4.6 mean free collision time

```
1 //Variable declaration
2 N=6.023*10**26 //Avagadro number
3 D=8960 //density
4 F_e=1 //no.of free
   electrons per atom
5 W=63.54 //Atomic weight
6 i=10
7 e=1.602*10**-19
8 m=9.1*10**-31
9 rho=2*10**-8
10 Cbar=1.6*10**6 //mean thermal
   velocity (m/s)
11
12 //Calculations
13 n=(N*D*F_e)/W
14 A=%pi*0.08**2*10**-4
15 Vd=i/(A*n*e) //Drift speed
16 Tc=m/(n*(e**2)*rho)
17 lamda=Tc*Cbar
18
19 //Result
20 printf('n =%0.3f *10**28 /m**3 \n ',(n/10**28))
21 printf('The drift speed Vd =%0.3f *10**-5 m/s \n ',
   ,(Vd*10**5))
22 printf('The mean free collision time Tc =%0.3f
   *10**-14 seconds \n ',(Tc*10**14))
23 printf('Mean free path =%0.3f *10**-8 m""(answer
   varies due to ing off errors) \n ',(lamda*10**8)
   )
```

Scilab code Exa 4.7 The mean free collision time

```
1 //Variable declaration
```

```

2 n=8.5*10**28
3 e=1.602*10**-19
4 t=2*10**-14
5 m=9.1*10**-31
6
7 // Calculations
8 Tc=n*(e**2)*t/m
9
10 // Result
11 printf('The mean free collision time =%0.3f *10**7
        ohm**-1 m**-1 \n ',(Tc/10**7))

```

Scilab code Exa 4.8 Relaxation time

```

1 // Variable declaration
2 e=1.6*10**-19
3 E=1 // (V/m)
4 rho=1.54*10**-8
5 n=5.8*10**28
6 m=9.1*10**-31
7 // Calculations
8 T=m/(rho*n*e**2)
9 Me=(e*T)/m
10 Vd=Me*E
11
12 // Result
13 printf('Relaxation time =%0.3f *10**-14 second \n ',
        ,(T*10**14))
14 printf('Mobility =%0.3f *10**-3 m**2/volt-s \n ',(
        Me*10**3))
15 printf('Drift Velocity=%0.3f m/s \n ',(Vd*100))

```

Scilab code Exa 4.9 Temperature coefficient of resistivity

```

1
2 //Variable declaration
3 rho_r=0
4 T=300
5 rho=1.7*10**-18
6
7 //Calculations
8 a=rho/T
9 rho_973=a*973
10
11 //Results
12 printf('Temperature coefficient of resistivity ,a =%0
    .3f \n ',(a*10**21))
13 printf('rho_973 =%0.3f *10**-8 ohm-m \n ',(rho_973
    *10**18))

```

Scilab code Exa 4.10 Increase in resistivity in copper

```

1 //Variable declaration
2 rho1=1.2*10**-8
3 p1=0.4
4 rho2=0.12*10**-8
5 p2=0.5
6 rho3=1.5*10**-8
7 //Calculations
8 R=(rho1*p1)+(rho2*p2)
9 R_c=R+rho3
10
11 //Results
12 printf('Increase in resistivity in copper =%0.3f
    *10**-8 ohm m \n ',(R*10**8))
13 printf('Total resistivity of copper alloy =%0.3f
    *10**-8 ohm m \n ',(R_c*10**8))
14 printf('The resistivity of alloy at 3K =%0.3f
    *10**-8 ohm m \n ',(R*10**8))

```


Chapter 5

Dielectric Properties and Magnetic Properties

Scilab code Exa 5.1 insulation resistance

```
1 //Variable declaration
2 rho=5*10**16; //resistivity (ohm m)
3 l=5*10**-2; //thickness (m)
4 b=8*10**-2; //length (m)
5 w=3*10**-2; //width (m)
6
7 //Calculation
8 A=b*w; //area (m**2)
9 Rv=rho*l/A;
10 X=l+b; //length (m)
11 Y=w; //perpendicular (m)
12 Rs=Rv*X/Y;
13 Ri=Rs*Rv/(Rs+Rv); //insulation resistance (ohm)
14
15 printf('insulation resistance is %0.3f *10**18 ohm'
16 , (Ri/10**18))
16 printf('answer varies due to rounding off errors')
```

Scilab code Exa 5.2 polarisability of He

```
1 //Variable declaration
2 epsilon0=8.84*10**-12;
3 R=0.55*10**-10; //radius (m)
4 N=2.7*10**25; //number of atoms
5
6 //Calculation
7 alpha_e=4*pi*epsilon0*R**3; //polarisability of
  He(farad m**2)
8 epsilon_r=1+(N*alpha_e/epsilon0); //relative
  permittivity
9
10 //Result
11 printf('polarisability of He is %0.3f *10**-40
  farad m**2\n',(alpha_e*10**40))
12 printf('relative permittivity is %0.3f \n',(
  epsilon_r))
13 printf('answer varies due to ing off errors')
```

Scilab code Exa 5.3 total dipole moment

```
1 //Variable declaration
2 A=360*10**-4; //area (m**2)
3 V=15; //voltage (V)
4 C=6*10**-6; //capacitance (farad)
5 epsilon_r=8;
6 epsilon0=8.84*10**-12;
7
8 //Calculation
9 E=V*C/(epsilon0*epsilon_r*A); //field strength (V/
  m)
```



```

10 dm=epsilon0*(epsilon0r-1)*V*A;    //total dipole
    moment(Cm)
11
12 //Result
13 printf('field strength is %0.3f *10**7 V/m\n',(E
    /10**7))
14 printf('total dipole moment is %0.3f *10**-12 Cm\n'
    ,(dm*10**12))

```

Scilab code Exa 5.4 the complex polarizability

```

1 //Variable declaration
2 epsilon0r=4.36;    //dielectric constant
3 t=2.8*10**-2;    //loss tangent(t)
4 N=4*10**28;    //number of electrons
5 epsilon0=8.84*10**-12;
6
7 //Calculation
8 epsilon_r = epsilon0r*t;
9 epsilonstar = (complex(epsilon0r,-epsilon_r));
10 alphastar = (epsilonstar-1)/(epsilonstar+2);
11 alpha_star = 3*epsilon0*alphastar/N;    //
    complex polarizability(Fm**2)
12
13 //Result
14 printf('the complex polarizability is %0.3f
    *10**-40 F-m**2 \n',alpha_star*10**40)
15 printf('answer cant be rounded off to 2 decimals as
    given in the textbook. Since it is a complex
    number and complex cant be converted to float')

```

Scilab code Exa 5.5 temperature rise

```

1 //Variable declaration
2 E1=10**-2*50; //energy loss(J)
3 H=E1*60; //heat produced(J)
4 d=7.7*10**3; //iron rod(kg/m**3)
5 s=0.462*10**-3; //specific heat(J/kg K)
6
7 //Calculation
8 theta=H/(d*s); //temperature rise(K)
9
10 //Result
11 printf('temperature rise is %0.3f K \n',(theta))

```

Scilab code Exa 5.6 magnetic field at the centre

```

1 //Variable declaration
2 e=1.6*10**-19; //charge(coulomb)
3 new=6.8*10**15; //frequency(revolutions per second
)
4 mew0=4*%pi*10**-7;
5 R=5.1*10**-11; //radius(m)
6
7 //Calculation
8 i=(e*new); //current(ampere)
9 B=mew0*i/(2*R); //magnetic field at the centre(
weber/m**2)
10 A=%pi*R**2;
11 d=i*A; //dipole moment(ampere/m**2)
12
13 //Result
14 printf('magnetic field at the centre is %0.3f weber/
m**2\n',(B))
15 printf('dipole moment is %0.3f *10**-24 ampere/m**2\
n',(d*10**24))

```

Scilab code Exa 5.7 intensity of magnetisation

```
1 //Variable declaration
2 chi=0.5*10**-5; //magnetic susceptibility
3 H=10**6; //field strength(ampere/m)
4 mew0=4*%pi*10**-7;
5
6 //Calculation
7 I=chi*H; //intensity of magnetisation(ampere/m)
8 B=mew0*(I+H); //flux density in material(weber/m
  **2)
9
10 //Result
11 printf('intensity of magnetisation is %0.3f ampere
  /m \n',I)
12 printf('flux density in material is %0.3f weber/m**2
  \n',(B))
```

Scilab code Exa 5.8 number of Bohr magnetons

```
1 //Variable declaration
2 B=9.27*10**-24; //bohr magneton(ampere m**2)
3 a=2.86*10**-10; //edge(m)
4 Is=1.76*10**6; //saturation value of
  magnetisation(ampere/m)
5
6 //Calculation
7 N=2/a**3;
8 mew_bar=Is/N; //number of Bohr magnetons(ampere
  m**2)
9 mew_bar=mew_bar/B; //number of Bohr magnetons(
  bohr magneon/atom)
```

```

10
11 //Result
12 printf('number of Bohr magnetons is %0.3f bohr
    magneon/atom\n',(mew_bar))

```

Scilab code Exa 5.9 average magnetic moment

```

1 //Variable declaration
2 mew0=4*%pi*10**-7;
3 H=9.27*10**-24; //bohr magneton(ampere m**2)
4 beta=10**6; //field (ampere/m)
5 k=1.38*10**-23; //boltzmann constant
6 T=303; //temperature(K)
7
8 //Calculation
9 mm=mew0*H*beta/(k*T); //average magnetic moment(
    bohr magneton/spin)
10
11 //Result
12 printf('average magnetic moment is %0.3f *10**-3
    bohr magneton/spin\n',(mm*10**3))

```

Scilab code Exa 5.10 hysteresis loss

```

1 //Variable declaration
2 A=94; //area(m**2)
3 vy=0.1; //value of length(weber/m**2)
4 vx=20; //value of unit length
5 n=50; //number of magnetization cycles
6 d=7650; //density(kg/m**3)
7
8 //Calculation
9 h=A*vy*vx; //hysteresis loss per cycle(J/m**3)

```

```
10 hs=h*n;          //hysteresis loss per second(watt/m
    **3)
11 pl=hs/d;        //power loss(watt/kg)
12
13 //Result
14 printf('hysteresis loss per cycle is %0.3f    J/m
    **3 \n',h)
15 printf('hysteresis loss per second is %0.3f    watt/
    m**3 \n',hs)
16 printf('power loss is %0.3f watt/kg\n',(pl))
```

Chapter 6

Semiconductors and Superconductivity

Scilab code Exa 6.1 number of electron hole pairs

```
1 //Variable declaration
2 ni1=2.5*10**19; //number of electron hole pairs
3 T1=300; //temperature(K)
4 Eg1=0.72*1.6*10**-19; //energy gap(J)
5 k=1.38*10**-23; //boltzmann constant
6 T2=310; //temperature(K)
7 Eg2=1.12*1.6*10**-19; //energy gap(J)
8
9 //Calculation
10 x1=-Eg1/(2*k*T1);
11 y1=(T1**(3/2))*exp(x1);
12 x2=-Eg2/(2*k*T2);
13 y2=(T2**(3/2))*exp(x2);
14 ni=ni1*(y2/y1); //number of electron hole
    pairs
15
16 //Result
17 printf('number of electron hole pairs is %0.3f
    *10**16 per cubic metre \n',(ni/10**16))
```

```
18 printf('answer varies due to ing off errors')
```

Scilab code Exa 6.2 intrinsic conductivity

```
1 //Variable declaration
2 w=72.6; //atomic weight
3 d=5400; //density (kg/m**3)
4 Na=6.025*10**26; //avagadro number
5 mew_e=0.4; //mobility of electron (m**2/Vs)
6 mew_h=0.2; //mobility of holes (m**2/Vs)
7 e=1.6*10**-19;
8 m=9.108*10**-31; //mass (kg)
9 ni=2.1*10**19; //number of electron hole pairs
10 Eg=0.7; //band gap (eV)
11 k=1.38*10**-23; //boltzmann constant
12 h=6.625*10**-34; //plancks constant
13 T=300; //temperature (K)
14
15 //Calculation
16 sigmab=ni*e*(mew_e+mew_h); //intrinsic
    conductivity (ohm-1 m-1)
17 rhob=1/sigmab; //resistivity (ohm m)
18 n=Na*d/w; //number of germanium atoms per m**3
19 p=n/10**5; //boron density
20 sigma=p*e*mew_h;
21 rho=1/sigma;
22
23 //Result
24 printf('intrinsic conductivity is %0.3f *10**4 ohm
    -1 m-1 \n',(sigma/10**4))
25 printf('intrinsic resistivity is %0.3f *10**-4
    ohm m \n',(rho*10**4))
26 printf('answer varies due to ing off errors')
27 printf('number of germanium atoms per m**3 is %0.3f
    *10**28 \n',(n/10**28))
```

Scilab code Exa 6.3 charge carrier density

```
1
2 //Variable declaration
3 e=1.6*10**-19;
4 RH=3.66*10**-4; //hall coefficient(m**3/coulomb)
5 sigma=112; //conductivity(ohm-1 m-1)
6
7 //Calculation
8 ne=3*pi/(8*RH*e); //charge carrier density(per m
   **3)
9 mew_e=sigma/(e*ne); //electron mobility(m**2/Vs
   )
10
11 //Result
12 printf('charge carrier density is %0.3f *10**22 per
   m**3 \n',int(ne/10**22))
13 printf('electron mobility is %0.3f m**2/Vs \n',
   ,(mew_e))
```

Scilab code Exa 6.4 conductivity during donor impurity

```
1 //Variable declaration
2 mew_e=0.13; //mobility of electron(m**2/Vs)
3 mew_h=0.05; //mobility of holes(m**2/Vs)
4 e=1.6*10**-19;
5 ni=1.5*10**16; //number of electron hole pairs
6 N=5*10**28;
7
8 //Calculation
9 sigma1=ni*e*(mew_e+mew_h); //intrinsic
   conductivity(ohm-1 m-1)
```



```

10 ND=N/10**8;
11 n=ni**2/ND;
12 sigma2=ND*e*mew_e;      //conductivity (ohm-1 m-1)
13 sigma3=ND*e*mew_h;      //conductivity (ohm-1 m-1)
14
15 //Result
16 printf('intrinsic conductivity is %0.3f *10**-3 ohm
    -1 m-1 %0.3f \n',(sigma1*10**3),sigma2)
17 printf('conductivity during donor impurity is %0.3f
    ohm-1 m-1 \n',sigma2)
18 printf('conductivity during acceptor impurity is %0
    .3f ohm-1 m-1',int(sigma3))

```

Scilab code Exa 6.5 conductivity

```

1 //Variable declaration
2 e=1.6*10**-19;
3 Eg=0.72;      //band gap(eV)
4 k=1.38*10**-23;    //boltzmann constant
5 T1=293;      //temperature(K)
6 T2=313;      //temperature(K)
7 sigma1=2;     //conductivity(mho m-1)
8
9 //Calculation
10 x=(Eg*e/(2*k))*((1/T1)-(1/T2));
11 y=(x/2.303);
12 z=(log10(sigma1));
13 log_sigma2=y+z;
14 sigma2=10**log_sigma2;    //conductivity(mho m-1)
15
16 //Result
17 printf('conductivity is %0.3f    mho m-1    \n',(
    sigma2))

```

Scilab code Exa 6.6 Concentration

```
1 //Variable declaration
2 ni=1.5*10**16
3 mu_n=1300*10**-4
4 mu_p=500*10**-4
5 e=1.6*10**-19
6 sigma=3*10**4
7
8 //Calculations
9 //Concentration in N-type
10 n1=sigma/(e*mu_n)
11 p1=ni**2/n1
12 //Concentration in P-type
13 p=sigma/(e*mu_p)
14 n2=(ni**2)/p
15
16 //Result
17 printf('a) Concentration in N-type\n ')
18 printf('n = %0.3f *10**24 m**-3 \n ',(n1
    *10**-24))
19 printf('Hence p = %0.3f *10**8 m**-3 \n ',(p1
    /10**8))
20 printf('b) Concentration in P-type\n ')
21 printf('p = %0.3f *10**24 m**-3 \n ',(p/10**24)
    )
22 printf('Hence n = %0.3f *10**8 m**-3 \n ',(n2
    /10**8))
```

Scilab code Exa 6.7 Current density

```
1 //Variable declaration
```

```

2 i=10**-2
3 A=0.01*0.001
4 RH=3.66*10**-4
5 Bz=0.5
6
7 // Calculations
8 Jx=i/A
9 Ey=RH*(Bz*Jx)
10 Vy=Ey*0.01
11
12 // Result
13 printf('Jx = %0.3 f      ampere/m**2  \n', Jx)
14 printf('Ey = %0.3 f      V/m      \n', (Ey))
15 printf('Vy = %0.3 f      mV      \n', (Vy*10**3))

```

Scilab code Exa 6.8 Position of fermi level

```

1 // Variable declaration
2 Ev=0
3 Ec=1.12
4 k=1.38*10**-23
5 T=300
6 mh=0.28
7 mc=0.12
8 e=1.6*10**-19
9 // Calculations
10 Ef=((Ec+Ev)/2)+((3*k*T)/(4*e))*log(mh/mc)
11
12 // Result
13 printf('Position of fermi level = %0.3 f      eV      \n',
      ,(Ef))

```

Scilab code Exa 6.9 Conductivity of intrinsic germanium at 300K

```

1 //Variable declaration
2 ni=2.5*10**19
3 mu_e=0.38
4 mu_h=0.18
5 e=1.6*10**-19
6
7 //Calculations
8 sigmai=ni*e*(mu_e+mu_h)
9
10 //Result
11 printf('Conductivity of intrinsic germanium at 300K
    = %0.3f      ohm**-1 m**-1      \n',(sigmai))

```

Scilab code Exa 6.10 Conductivity

```

1 //Variable declaration
2 m=9.1*10**-31
3 k=1.38*10**-23
4 T=300
5 h=6.626*10**-34
6 Eg=1.1
7 e=1.6*10**-19
8 mu_e=0.48
9 mu_h=0.013
10 //Calculations
11 ni=2*((2*%pi*m*k*T)/h**2)**(3/2)*exp(-(Eg*e)/(2*k*T)
    )
12 sigma=ni*e*(mu_e+mu_h)
13
14 //Result
15 printf('Conductivity = %0.3f      *10**-3 ohm**-1 m
    **-1      \n',(sigma*10**3))

```

Scilab code Exa 6.11 The electron concentration

```
1 //Variable declaration
2 Na=5*10**23
3 Nd=3*10**23
4 ni=2*10**16
5 //Calculations
6 p=((Na-Nd)+(Na-Nd))/2
7
8 //Result
9 printf('p = %0.3f      *10**23 m**-3 \n',p*10**-23)
10 printf('The electron concentration is given by n =
      %0.3f      *10**9 m**-3 \n',ni**2/p*10**-9)
```

Scilab code Exa 6.12 resistance

```
1
2 //Variable declaration
3 Vh=37*10**-6
4 thick=1*10**-3
5 width=5
6 Iy=20*10**-3
7 Bz=0.5
8
9 //Calculations
10 Rh=(Vh*width*thick)/(width*Iy*Bz)
11
12 //Result
13 printf('Rh = %0.3f      *10**-6 C**-1 m**3 \n', (Rh
      *10**6))
```

Scilab code Exa 6.13 Calculate Dn and Dp

```

1
2 //Variable declaration
3 Vt=0.0258
4 mu_n=1300
5 mu_p=500
6
7 //Calculations
8 Dn=Vt*mu_n
9 Dp=Vt*mu_p
10
11 //Result
12 printf('Dn = %0.3 f      cm**2 s**-1   \n',Dn)
13 printf('Dp = %0.3 f cm**2 s**-1      \n',Dp)

```

Scilab code Exa 6.14 Electrical Conductivity

```

1
2 //Variable declaration
3 ni=1.5*10**16
4 Nd=2*10**19
5 e=1.602*100**-19
6 mu_n=0.12
7
8 //Calculations
9 p=ni**2/Nd
10 E_c=e*Nd*mu_n
11
12 //Result
13 printf('The hole concentration p = %0.3 f      *10**13
        /m**3   \n',(p*10**-13))
14 printf('n= Nd = %0.3 f      *10**19   \n',(Nd*10**-19)
        )
15 printf('Electrical Conductivity = %0.3 f      ohm**-1
        m**-1   \n',(E_c*10**19))

```

Scilab code Exa 6.15 Current density

```
1
2 //Variable declaration
3 N=1/60
4 e=1.6*10**-19
5 ni=2.5*10**13
6 b=5*10**13
7 E=2
8
9 //Calculations
10 n=(b+sqrt(2*b**2))/2
11 mu_p=N/(3*e*ni)
12 mu_i=2*mu_p
13 np=ni**2
14 p=(ni**2)/n
15 e=1.6*10**-19
16 E=2
17 J=(e*E)*((n*mu_i)+(p*mu_p))
18 //Result
19 printf('mu_p= %0.3 f      cm**2/V-s      \n',(mu_p))
20 printf('n= %0.3 f      *10**13/cm**3      \n',(n/10**13))
21 printf('p= %0.3 f      *10**13/cm**3      \n',(p*10**-13)
22 )
23 printf('J= %0.3 f      A/m**2      \n',(J*10**4))
24 printf('//Answer varies due to ing of numbers')
```

Scilab code Exa 6.16 Drift velocity

```
1
2 //Variable declaration
3 rho=47*10**-2
```

```

4 e=1.6*10**-19
5 mu_n=0.39
6 mu_p=0.19
7 E=10**4
8
9 // Calculations
10 ni=1/(rho*e*(mu_n+mu_p))
11 Dh=mu_p*E
12 De=mu_n*E
13
14 // Results
15 printf('ni = %0.3f      *10**19 /m**3   \n',(ni
      /10**19))
16 printf('Drift velocity of holes %0.3f   ms**-1   \n',
      Dh)
17 printf('Drift velocity of electrons= %0.3f   ms**-1
      \n',De)

```

Scilab code Exa 6.17 critical field

```

1 // Variable declaration
2 H0=64*10**3; // initial field (ampere/m)
3 T=5; // temperature (K)
4 Tc=7.26; // transition temperature (K)
5
6 // Calculation
7 H=H0*(1-(T/Tc)**2); // critical field (ampere/m)
8
9 // Result
10 printf('critical field is %0.3f      *10**3 ampere/m
      \n',(H/10**3))

```

Scilab code Exa 6.18 Frequency of generated microwaves


```

1 //Variable declaration
2 e=1.6*10**-19
3 V=1*10
4 h=6.625*10**-34
5
6 //Calculations
7 v=(2*e*V**-3)/h
8
9 //Result
10 printf('Frequency of generated microwaves= %0.3 f
        *10**9 Hz      \n',(v/10**9))

```

Scilab code Exa 6.19 Penetration depth

```

1 //Variable declaration
2 d=7300 //density in (kg/m**3)
3 N=6.02*10**26 //Avagadro Number
4 A=118.7 //Atomic Weight
5 E=1.9 //Effective mass
6 e=1.6*10**-19
7
8 //Calculations
9 n=(d*N)/A
10 m=E*9.1*10**-31
11 x=4*%pi*10**-7*n*e**2
12 lamda_L=sqrt(m/x)
13
14 //Result
15 printf('Number of electrons per unit volume = %0.3 f
        *10**28/m**3      \n',(n/10**28))
16 printf('Effective mass of electron m = %0.3 f
        *10**-31 kg      \n',(m*10**31))
17 printf('Penetration depth = %0.3 f      Angstroms \n',
        lamda_L*10**8)
18 printf('//The answer given in the text book is wrong

```

’)

Scilab code Exa 6.20 Calculate wavelength

```
1
2 //Variable declaration
3 lamda_L1=39.6*10**-9
4 lamda_L2=173*10**-9
5 T1=7.1
6 T2=3
7
8 //Calculations
9 x=(lamda_L1/lamda_L2)**2
10 Tc4=(T1**4)-((T2**4)*x)/(1-x)
11 Tc=(Tc4)**(1/4)
12 printf('Tc = %0.3 f      K      \n', (Tc))
13 printf('lamda0= %0.3 f      nm      \n', ((sqrt(1-(T2/Tc)
      **4)*lamda_L1)*10**9))
```

Scilab code Exa 6.21 Critical current density

```
1
2 //Variable declaration
3 H0=6.5*10**4 // (ampere/metre)
4 T=4.2 //K
5 Tc=7.18 //K
6 r=0.5*10**-3
7
8 //Calculations
9 Hc=H0*(1-(T/Tc)**2)
10 Ic=(2*pi*r)*Hc
11 A=pi*r**2
12 Jc=Ic/A //Critical current density
```

```
13
14 //Result
15 printf('Hc = %0.3 f      *10**4 \n', (Hc/10**4))
16 printf('Critical current density, Jc = %0.3 f *10**8
      ampere/metre**2      \n', (Jc/10**8))
```

Scilab code Exa 6.22 New critical temperature for mercury

```
1
2 //Variable declaration
3 Tc1=4.185
4 M1=199.5
5 M2=203.4
6
7 //Calculations
8 Tc2=Tc1*(M1/M2)**(1/2)
9
10 //Result
11 printf('New critical temperature for mercury = %0.3 f
      K \n', (Tc2))
```

Chapter 7

Lasers

Scilab code Exa 7.1 Calculatate Divergence

```
1
2 //variable declaration
3 r1 = 7;           //in radians
4 r2 = 3;           //in radians
5 d1 = 4;           //Converting from mm to
   radians
6 d2 = 6;           //Converting from mm to
   radians
7
8 //calculations
9 D = (r2-r1)/(d2*10**3-d1*10**3) //Divergence
10
11 //Result
12 printf('Divergence = %0.3f *10**-3 radian \n',
   (D*10**3))
```

Scilab code Exa 7.2 Relative Population

```

1
2 //variable declaration
3 C=3*10**8 //The speed of light
4 Lamda=6943 //Wavelength
5 T=300 //Temperature in Kelvin
6 h=6.626*10**-34 //Planck constant
7 k=1.38*10**-23 //Boltzmann's constant
8
9 //Calculations
10
11 V=(C)/(Lamda*10**-10) //Frequency
12 R=exp(h*V/(k*T)) //Relative population
13
14 //Result
15 printf('Frequency (V) = %0.3f *10**14 Hz \n',(
    V/10**14))
16 printf('Relative Population= %0.3f *10**30 \n',
    ,(R/10**30))

```

Scilab code Exa 7.3 Power density

```

1
2 //variable declaration
3 C=3*10**8 //Velocity of light
4 W=632.8*10**-9 //wavelength
5 P=2.3
6 t=1
7 h=6.626*10**-34 //Planck constant
8 S=1*10**-6
9
10 //Calculations
11 V=C/W //Frequency
12 n=((P*10**-3)*t)/(h*V) //no. of photons emitted
13 PD=P*10**-3/S //Power density
14

```

```

15 //Result
16 printf('Frequency= %0.3 f      *10**14 Hz      \n', (V
    /10**14))
17 printf('no. of photons emitted= %0.3 f      *10**15
    photons/sec      \n', (n/10**15))
18 printf('Power density = %0.3 f      kWm**-2      \n', (PD
    /1000))

```

Scilab code Exa 7.4 Wavelength

```

1
2 //variable declaration
3 h=6.626*10**-34           //Planck constant
4 C=3*10**8                 //Velocity of light
5 E_g=1.44                  //bandgap
6
7 //calculations
8 lamda=(h*C)*10**10/(E_g*1.6*10**-19)      //
    Wavelength
9
10 //Result
11 printf('Wavelength = %0.3 f      Angstrom      \n', (lamda
    ))

```

Scilab code Exa 7.5 Band gap

```

1
2 //variable declaration
3 W=1.55                    //wavelength
4
5 //Calculations
6 E_g=(1.24)/W              //Bandgap in eV
7

```

```
8 //Result
9 printf('Band gap = %0.3 f      eV \n',E_g)
```

Chapter 8

Fiber Optics

Scilab code Exa 1.1 Critical angle

```
1 //variable declaration
2 n1=1.50           //Core refractive index
3 n2=1.47           //Cladding refractive index
4
5 //Calculations
6 C_a=asin(n2/n1)   //Critical angle
7 N_a=(n1**2-n2**2)**(1/2)
8 A_a=asin(N_a)
9
10 //Results
11 printf('The Critical angle =%0.3f degrees\n',(C_a
    *180/%pi))
12 printf('The numerical aperture =%0.3f \n',(N_a))
13 printf('The acceptance angle =%0.3f degrees\n',(A_a
    *180/%pi))
```

Scilab code Exa 8.2 No of modes propogated inside the fiber


```

1 //variable declaration
2 d=50 //diameter
3 N_a=0.2 //Numerical aperture
4 lamda=1 //wavelength
5
6 //Calculations
7 N=4.9*(((d*10**-6*N_a)/(lamda*10**-6))**2)
8
9 //Result
10 printf('N =%0.3f \n',N)
11 printf('Fiber can support%0.3f guided modes \n',N)
12 printf('In graded index fiber , No.of modes
propogated inside the fiber =%0.3f only',N/2)

```

Scilab code Exa 8.3 Numerical aperture

```

1 //variable declaration
2 d=50 //diameter
3 n1=1.450
4 n2=1.447
5 lamda=1 //wavelength
6
7 //Calculations
8 N_a=(n1**2-n2**2) //Numerical aperture
9 N=4.9*(((d*10**-6*N_a)/(lamda*10**-6))**2)
10
11 //Results
12 printf('Numerical aperture =%0.3f ',N_a)
13 printf('No. of modes that can be propogated =%0.3f \
n', (N))

```

Scilab code Exa 8.4 Numerical aperture

```

1 //variable declaration
2 delta=0.05
3 n1=1.46
4
5 //Calculation
6 N_a=n1*(2*delta)**(1/2) //Numerical aperture
7
8 //Result
9 printf('Numerical aperture =%0.3f \n',(N_a))

```

Scilab code Exa 8.5 maximum no of modes propogating through fiber

```

1 //variable declaration
2 a=5
3 n1=1.450
4 n2=1.447
5 lamda=1 //wavelength
6
7 //Calculations
8 N_a=(n1**2-n2**2) //Numerical aperture
9
10 N=4.9*((a*10**-6*sqrt(N_a)/(lamda*10**-6))**2)
11
12 //Result
13
14 printf('maximum no.of modes propogating through
15 fiber =%0.3f \n',(N))
16 printf('Correction needed')

```

Scilab code Exa 8.6 Number of modes

```

1 //variable declaration
2 a=100

```

```

3 N_a=0.3 //Numerical aperture
4 lamda=850 //wavelength
5
6 //Calculations
7 V_n=(2*(%pi**2*a**2*10**-12*N_a**2)/lamda
    **2*10**-18)
8 //Result
9 printf('Number of modes =%0.3f modes\n', (V_n
    /10**-36))
10 printf('No. of modes is doubled to account for the
    two possible polarisations')
11 printf('Total No. of modes =%0.3f \n', (V_n/10**-36)
    *2)

```

Scilab code Exa 8.7 Cutoff Wavelength

```

1 //variable declaration
2 a=5;
3 n1=1.48;
4 delta=0.01;
5 V=25;
6
7 //Calculation
8 lamda=(%pi*(a*10**-6)*n1*sqrt(2*delta))/V //
    Cutoff Wavelength
9
10 //Result
11 printf('Cutoff Wavelength =%0.3f micro m. \n', (
    lamda*10**7))

```

Scilab code Exa 8.8 Maximum core radius

```

1 //variable declaration

```

```

2 V=2.405
3 lamda=1.3
4 N_a=0.05
5
6 // Calculations
7 a_max=(V*lamda)/(2*pi*N_a)
8
9 // Result
10 printf('Maximum core radius=%0.3f micro m\n',(a_max)
    )

```

Scilab code Exa 8.9 Acceptance angle

```

1 //variable declaration
2 N_a=0.3
3 gamma=45
4
5 // Calculations
6 theta_a=asin(N_a)
7 theta_as=asin((N_a)/cos(gamma))
8
9 // Results
10 printf('Acceptance angle, theta_a =%0.3f degrees\n',
    ,(theta_a*180/pi))
11 printf('For skew rays, theta_as %0.3f degrees\n',(
    theta_as*180/pi))
12 printf('//Answer given in the textbook is wrong')

```

Scilab code Exa 8.10 Numerical aperture and Acceptance angle

```

1 //variable declaration
2 n1=1.53
3 delta=0.0196

```

```

4
5 // Calculations
6 N_a=n1*(2*delta)**(1/2)
7 A_a=asin(N_a)
8 // Result
9 printf('Numerical aperture =%0.3f \n',(N_a))
10 printf('Acceptance angle =%0.3f degrees \n',(A_a
    *180/%pi))

```

Scilab code Exa 8.11 Core radius

```

1 //variable declaration
2 n1=1.480
3 n2=1.465
4 V=2.405
5 lamda=850*10**-9
6
7 // Calculations
8 delta=(n1**2-n2**2)/(2*n1**2)
9 a=(V*lamda*10**-9)/(2*%pi*n1*sqrt(2*delta))
10
11 // Results
12 printf('delta =%0.3f \n',(delta))
13 printf('Core radius ,a =%0.3f micro m\n',(a*10**15))

```

Scilab code Exa 8.12 Total dist travelled by light over one metre of fiber

```

1 //variable declaration
2 n1=1.5
3 n2=1.49
4 a=25
5
6 // Calculations

```

```

7 C_a=asin(n2/n1)           // Critical angle
8 L=2*a*tan(C_a)
9 N_r=10**6/L
10
11 //Result
12 printf('Critical angle=%0.3f degrees\n',(C_a*180/%pi
   ))
13 printf('Fiber length covered in one reflection=%0.3f
   micro m\n',(L))
14 printf('Total no.of reflections per metre=%0.3f \n'
   ,(N_r))
15 printf('Since L=1m, Total dist. travelled by light
   over one metre of fiber =%0.3f m \n',(1/sin(C_a))
   )

```

Scilab code Exa 8.13 No of modes

```

1 //variable declaration
2 alpha=1.85
3 lamda=1.3*10**-6
4 a=25*10**-6
5 N_a=0.21
6
7 //Calculations
8 V_n=((2*pi**2)*a**2*N_a**2)/lamda**2
9 N_m=(alpha/(alpha+2))*V_n
10
11 printf('No. of modes =%0.3f =155(approx)\n',(N_m))
12 printf('Taking the two possible polarizations, Total
   No. of nodes =%0.3f \n',(N_m*2))

```

Scilab code Exa 8.14 signal attenuation per unit length

```

1 //variable declaration
2 P_i=100
3 P_o=2
4 L=10
5
6 //Calculations
7 S=(10/L)*log(P_i/P_o)
8 O=S*L
9
10 //Result
11 printf('a.Signal attention per unit length =%0.3f dB
        km**-1\n',(S))
12 printf('b.Overall signal attenuation =%0.3f dB\n',(O
        ))
13 printf('//Answer given in the textbook is wrong')

```

Scilab code Exa 8.15 Bandwidth length product

```

1 //variable declaration
2 L=10
3 n1=1.55
4 delta=0.026
5 C=3*10**5
6
7 //Calculations
8 delta_T=(L*n1*delta)/C
9 B_W=10/(2*delta_T)
10
11 //Result
12 printf('Total dispersion =%0.3f ns\n',(delta_T
        /10**-9))
13 printf('Bandwidth length product =%0.3f Hz-km\n',(
        B_W/10**5))
14 printf('//Answer given in the text book is wrong")

```
