# Scilab Textbook Companion for Materials Science and Engineering - A First Course by V. Raghavan[1]

Created by
Vareesh Pratap
B.Tech
Mechanical Engineering
Madan Mohan Malaviya University of Technology
College Teacher
None
Cross-Checked by
Aviral Yadav

June 2, 2016

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

# Book Description

**Title:** Materials Science and Engineering - A First Course

**Author:** V. Raghavan

**Publisher:** Prentice Hall, India

**Edition:** 5

**Year:** 2007

**ISBN:** 9788120324558

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 2

# Equilibrium and Kinetics

**Scilab code Exa 2.1** Calculate the entropy increase

```
1 // Calculate the entropy increase
2 clc
3 del_h = 6.02 // Heat added in kJ/mol
4 t_m = 273.15 // mean temperature in kelvin
5 printf("\n Example 2.1")
6 del_s = del_h*1e3/t_m
7 printf("\n Increase in entropy is %.2f J molˆ-1 Kˆ-1
    ",del_s)
```

**Scilab code Exa 2.2** Calculation of fraction of atoms with energy equal to or greater than 1eV at temperatures

```
1 // Calculation of fraction of atoms with energy
    equal to or greater than 1eV at temperatures
2 clc
3 E = 1 // energy in electron volt
4 e = 1.6e-19 // charge on electron
```

```
5  k = 1.38e-23 // constant
6  t1 = 300 // temperature in K
7  t2 = 1500 // temperature in K
8  printf("\n Example 2.2")
9  printf("\n\n Part A:")
10 n_N = exp(-(e/(k*t1)))
11 printf("\n Fraction of atoms with energy equal to or
       greater than 1eV at temperature %d K is %.2e ",
     t1,n_N) // numerical value of answer in book is 2
     e-17
12 printf("\n\n Part B:")
13 n_N = exp(-(e/(k*t2)))
14 printf("\n Fraction of atoms with energy equal to or
       greater than 1eV at temperature %d K is %.2e ",
     t2,n_N) // numerical value of answer in book is
     4.3e4
```

---

# Chapter 3

# Crystal Geometry and Structure Determination

**Scilab code Exa 3.2** Calculate effective number of lattice point in three cubic space lattice

```
1 // Calculate effective number of lattice point in
     three cubic space lattice
2 clc
3 sc_n = 1/8 // sharing of one lattice point in a unit
      cell
4 sc_N = 8 // Number of lattice points in Simple cubic
5 bcc_n_e = 1/4// sharing of one edge lattice point in
      a BCC
6 bcc_N_e = 4// Number of edge lattice point in a BCC
7 bcc_n_c  = 1// sharing of one body center lattice
     point in a BCC
8 bcc_N_c = 1// Number of body center lattice point in
      a BCC
9 fcc_n_e = 1/8 // sharing of one corner lattice point
      in a FCC
10 fcc_N_e = 8// Number of corner lattice point in a
     FCC
11 fcc_n_f = 1/2 // sharing of one face center lattice
```

```
    point in a FCC
12 fcc_N_f = 6// Number of face center lattice point in
    a FCC
13 printf("\n Example 3.2 ")
14 sc_net = sc_n*sc_N
15 bcc_net = bcc_n_e*bcc_N_e+bcc_n_c*bcc_N_c
16 fcc_net = fcc_n_e*fcc_N_e+fcc_n_f*fcc_N_f
17 printf("\n Effective number of lattice points are as
    :")
18 printf("\n\n Space lattice \t Abbreviation \t
    Effective number of lattice point in unit cell \n
    ")
19 printf("\n Simple cubic \t\tSC \t\t\t\t %d\n Body
    center cubic\tBCC \t\t\t\t %d\n Face centered
    cubic\tFCC \t\t\t\t %d ",sc_net,bcc_net,fcc_net)
```

**Scilab code Exa 3.7** Determine interplanar spacing and miller indices

```
1  //  Determine Interplanar spacing and miller indices
2  clc
3  a = 3.16 // lattice parameter in angstrom
4  l1 = 1 // line number
5  l2 = 2 // line number
6  l3 = 3 // line number
7  l4 = 4 // line number
8  theta1 = 20.3 // angle for line 1
9  theta2 = 29.2// angle for line 2
10 theta3 = 36.7// angle for line 3
11 theta4 = 43.6// angle for line 4
12 n = 1 // order
13 lambda = 1.54 // wavelength in angstrom
14 printf("\n Example 3.7")
15 d1 = lambda/(2*sin(theta1*%pi/180))
```

```
16 d2 = lambda/(2*sin(theta2*%pi/180))
17 d3 = lambda/(2*sin(theta3*%pi/180))
18 d4 = lambda/(2*sin(theta4*%pi/180))
19 x1 = a^2/d1^2
20 x2 = a^2/d2^2
21 x3 = a^2/d3^2
22 x4 = a^2/d4^2 // where x is function of h,k and l
23 printf("\n Interplanar spacing is %.3f angstrom ",
     x1) // answer in book is 2.220 angstrom
24 if  floor(x1) == 2 then
25     printf("\n\n For a^2/d^2 = %d \t Reflection
         plane is {110}",x1)
26 end
27
28 if floor(x2) == 4 then
29         printf("\n For a^2/d^2 = %d \t Reflection
             plane is {200}",x2)
30 end
31
32  if floor(x3) == 6 then
33     printf("\n For a^2/d^2 = %d \t Reflection plane
         is {211}",x3)
34 end
35
36  if floor(x4) == 8 then
37     printf("\n For a^2/d^2 = %d \t Reflection plane
         is {220}",x4)
38 end
```

**Scilab code Exa 3.8** Determine structure and lattice parameter of material

```
1 // Determine structure and lattice parameter of
```

```
    material
clc
d = 114.6 // diameter of power camera in angstrom
lambda = 1.54 // wavelength in angstrom
s1 = 86
s2 = 100
s3 = 148
s4 = 180
s5 = 188
s6 = 232
s7 = 272
printf("\n Example 3.8")
R = d/2       // Radius
if R==57.3 then
    k = 1/4 // Bragg angle factor
end
a1 = (sin(s1*k*%pi/180))^2
a2 = (sin(s2*k*%pi/180))^2
a3 = (sin(s3*k*%pi/180))^2
a4 = (sin(s4*k*%pi/180))^2
a5 = (sin(s5*k*%pi/180))^2
a6 = (sin(s6*k*%pi/180))^2
a7 = (sin(s7*k*%pi/180))^2
c = 22 // constant to convert into integral number

printf("\n Within experimental error, values are as
    in integral ratio are as: \n %d:%d:%d:%d:%d:%d:%d
    ",ceil(c*a1),ceil(c*a2),ceil(c*a3),ceil(c*a4),
    ceil(c*a5),ceil(c*a6),ceil(c*a7))
printf("\n So, this structure is FCC and material is
    copper with 3.62 angstrom lattice parameter")
```

# Chapter 4

# Atomic Structure and Chemical Bonding

**Scilab code Exa 4.1** Calculate frequency and wavelength of radiation

```
1 // Calculate frequency and wavelength of radiation
2 clc
3 E = 1.64e-18 // energy difference between two states
      in J
4 h= 6.626e-34 // planks constant
5 c = 2.998e8 // speed of light in m/s
6 printf("\n Example 4.1")
7 nu = E/h
8 lambda = c/nu
9 printf("\n Frequency of emitted radiation is %.2e Hz
     ",nu)
10 printf("\n Wavelength of emitted radiation is %.2e m
      \n\t \tor\t\t %d angstrom",lambda,lambda*1e10)//
      answer in book is 1210 angstrom
```

**Scilab code Exa 4.3** Reconcile the difference of energy

```
1 // Reconcile the difference of energy
2 clc
3 e_a = 713 // enthalpy of atomization in kJ/mol
4 e_b = 347 // bond energy in kJ/mol
5 a = 4 // total number of atoms in single crystal
     structure
6 b = 2 // number of atoms in a bond
7 printf("\n Example 4.3")
8 k = a/b // effective number of bond per atom
9 e = k*e_b
10 printf("\n %d kJ should be the enthalpy of
     atomization of diamond", e)
11 printf("\n However, %d kJ is very close to %d kJ",e,
     e_a)
```

**Scilab code Exa 4.4** Calculation of fraction of hydrogen bonds which breaks during ice melting

```
1 // Calculation of fraction of hydrogen bonds which
     breaks during ice melting
2 clc
3 del_h = 6.02 // enthalpy of fusion in kJ/mol
4 n = 2 // number of hydrogen atom in 1 water atom
5 del_b = 20.5 // hydrogen bond energy in kJ/mol
6 printf("\n Example 4.4")
7 f = del_h/(n*del_b)
8 printf("\n Fraction of hydrogen bonds which broken
     is %.2f",f)
```

# Chapter 5

# Structure of Solid

**Scilab code Exa 5.1** Calculate packing efficiency and density of diamond

```
1 // Calculate packing efficiency and density of
    diamond
2 clc
3 n_c = 1/8 // sharing of corner atom in a unit cell
4 N_c = 8 // Number of corner atoms in unit cell
5 n_b = 1 // sharing of body centered atom in a unit
    cell
6 N_b = 4 // Number of body centered atoms in unit
    cell
7 n_f = 0.5// sharing of face centered atom in a unit
    cell
8 N_f = 6// Number of face centered atoms in unit cell
9 a = 1 // let lattice parameter
10 m = 12 // mass of carbon
11 printf("\n Example 5.1")
12 printf("\n Part A:")
13 N = n_c*N_c+n_b*N_b+n_f*N_f // effective number of
    atoms
14 r = a*sqrt(3)/8
15 p_e = N*4/3*%pi*r^3/a^3 // packing efficiency
16
```

```
17 printf("\n Packing efficiency of diamond is %.2f",
     p_e)
18 printf("\n\n Part B:")
19 a = 3.57 // lattice parameter in angstrom
20 d = m*1.66e-27*N/(a*1e-10)^3
21 printf("\n Density of diamond is %d Kg/m^3",d)//
     numerical answer in book is 3500
22 printf("\n Density of diamond is %.1f g/cm^3",d
     /1000)
```

**Scilab code Exa 5.3** Calculate the cation to anion ratio for an ideally close packed HCP crystal

```
1 // calculate the c/a ratio for an ideally close
     packed HCP crystal
2 clc
3 a = 1 // let
4 PR = a
5 printf("\ Example 5.3")
6 RT = a/sqrt(3)
7 PT = sqrt(PR^2-RT^2)
8 c_a = 2*PT/PR
9 // Calculations are made on the crystal structure
     drawn in book
10 printf("\n c/a ratio for an ideally close packed HCP
     crystal is %0.3f ",c_a)
```

**Scilab code Exa 5.4** find the size of largest sphere that can fit into a tetrahedral void

```
1 // find the size of largest sphere that can fit into
      a tetrahedral void
2 clc
3 r = 1 // let
4 a = 3/4
5 printf("\n Example 5.4")
6 pt = 2*sqrt(2/3)*r
7 s = a*pt-r // size of sphere
8 printf("\n Size of largest sphere that can fit into
     a tetrahedral void is %.3fr",s)
```

**Scilab code Exa 5.5** Find critical radius ratio for triangular coordination

```
1 // find critical radius ratio for triangular
     coordination
2 clc
3 theta = 60 // angle in degree
4 printf("\n Example 5.5")
5 r_c_a = (2/3*2*sin(theta*%pi/180))-1 // ratio
     calculation
6 printf("\n Critical radius ratio for triangular
     coordination is %0.3f ",r_c_a)
```

**Scilab code Exa 5.6** Calculate density of MgO

```
1 // Calculate density of MgO
2 clc
3 r_mg = 0.78 // radius of magnesium cation in
     angstrom
```

```
r_o = 1.32 // radius of oxygen anion in angstrom
n = 4 // effective number of unit cell
m_o = 16 // mass of oxygen
m_mg = 24.3 // mass of magnesium
printf("\n Example 5.6")
a = 2*(r_mg+r_o)// lattice parameter
d = (m_o+m_mg)*1.66e-27*n/(a*1e-10)^3// density
printf("\n Density of MgO is %d Kg/m^3",d) // answer
    is 3610 kg/m^3
printf("\n Density of MgO is %0.2f g/cm^3",d/1000)
```

# Chapter 6

# Crystal Imperfection

**Scilab code Exa 6.1** Find equilibrium concentration of vacancies in metals at given temperature

```
1 // Find equilibrium concentration of vacancies in
     metals at given temperature
2 clc
3 t1 = 0 // temperature in kelvin
4 t2 = 300 // temperature in kelvin
5 t3 = 900// temperature in kelvin
6 R = 8.314 // universal gas constant
7 del_hf_al = 68 // Enthalpy of formation of aluminium
      crystal in KJ
8 del_hf_ni = 168 // Enthalpy of formation of nickel
     crystal in KJ
9 printf("\n Example 6.1")
10
11 printf("\n Equilibrium concentration of vacancies of
      aluminium at %dK is 0",t1)
12 n_N = exp(-del_hf_al*1e3/(R*t2))
13 printf("\n Equilibrium concentration of vacancies of
      aluminium at %dK is %.2e",t2,n_N) // answer in
     book is 1.45e-12
14 n_N = exp(-del_hf_al*1e3/(R*t3))
```

```
15 printf("\n Equilibrium concentration of vacancies of
       aluminium at %dK is %.2e",t3,n_N) // answer in
   book is 1.12e-4
16
17 printf("\n\n Equilibrium concentration of vacancies
   of Nickel at %dK is 0",t1)
18 n_N = exp(-del_hf_ni*1e3/(R*t2))
19 printf("\n Equilibrium concentration of vacancies of
       Nickel at %dK is %.2e",t2,n_N)
20 n_N = exp(-del_hf_ni*1e3/(R*t3))
21 printf("\n Equilibrium concentration of vacancies of
       Nickel at %dK is %.2e",t3,n_N) // answer in book
       is 1.78e-10
```

**Scilab code Exa 6.2** Compute the line energy of dislocation

```
1 // Compute the line energy of dislocation
2 clc
3 a = 2.87 // lattice parameter in angstrom
4 b= 2.49 // magnitude of burgers vector in angstrom
5 G = 80.2 // shear modulus in GN
6 printf("\n Example 6.2")
7 E = G*1e9*(b*1e-10)^2*1/2
8 printf("\n Line energy of dislocation is %.2e J m^-1
   ",E)
```

**Scilab code Exa 6.4** Calculation of down climb of crystal on heating

```
1 // calculation of down climb of crystal on heating
```

```
clc
a = 1e10// total number of edge dislocation
N = 6.023e23 // Avogadro number
R = 8.314 // Universal gas constant
t1 = 0 // initial temperature in K
t2 = 1000 // Final temperature in K
del_hf = 100 // Enthalpy of vacancy formation in KJ
d = 2 // length of one step in angstrom
v = 5.5e-6 // volume of one mole crystal
printf("\n Example 6.4")
n = N*exp(-(del_hf*1e3)/(R*(t2-t1)))/v
k = 1/(d*1e-10) // atoms required for 1 m climb
b = n/(k*a)// average amount of climb
c = b*d*1e-10

printf("\n Average down climb of crystal is %.2em",c
   )
```

**Scilab code Exa 6.5** Calculate surface energy of copper crystal of type 111

```
// Calculate surface energy of copper crystal of
   type {111}
clc
E = 56.4 // bond energy in KJ
N_a = 6.023e23 // Avogadro s number
n = 12 // number of bonds
m = 3 // number of broken bonds
N = 1.77e19 // number of atoms in copper crystal of
   type {111} per m^2
printf("\n Example 6.5")
b_e = 1/2*E*1e3*n/N_a // bond energy per atom
e_b = b_e*m/n // energy of broken bond at surface
s_e = e_b*N // surface enthalpy of copper
```

```
12 printf("\n Surface enthalpy of copper is %0.2f J m
   ^-2",s_e)
13 printf("\n Surface enthalpy of copper is %d erg cm
   ^-2",s_e*1e3)
14 // Answer in book is 2490 erg cm^-2
```

**Scilab code Exa 6.6** Compute the angle at the bottom of groove of a boundary

```
1 // Compute the angle at the bottom of groove of a
   boundary
2 clc
3 Gamma_gb = 1 // let, energy of grain boundary
4 Gamma_s = 3* Gamma_gb// energy of free surface
5 printf("\n Example 6.6")
6 theta = 2*acos(1/2*Gamma_gb/Gamma_s)
7 printf("\n Angle at the bottom of groove of a
   boundary is %d degrees.",ceil(theta*180/%pi))
```

# Chapter 7

# Phase Diagram

**Scilab code Exa 7.1** Find degrees of freedom of a system of two components

```
1 // Find degrees of freedom of a system of two
     components
2 clc
3 c = 2 // number of components
4 printf("\n Example 7.1")
5 for n = 1:4
6     p = (c-1) +2 // Total variables
7     f = c-n+2 // degree of freedom
8     printf("\n\n Degree of freedom for two
         components when \n number of phases is %d is
         %d",n,f)
9 end
```

**Scilab code Exa 7.2** Find minimum number of component in system

```
// Find minimum number of component in system
clc
p = 4 // number of phases of system
f = 0 // number of degree of system

printf("\n Example 7.2")
C = f+p-1 // components number
printf("\n Minimum number of components in system is
    %d",C)
```

**Scilab code Exa 7.3** Calculate amount of pure water that can be extracted from sea water

```
// Calculate amount of pure water that can be
   extracted from sea water
clc
L = 23.3 // % composition of L
a = 3.5 // concentration of Nacl in sea water
ice = 0 // % composition of ice
printf("\n Example 7.3")
f_ice = (L-a)/(L-ice)
printf("\n Fractional amount of pure water that can
   be extracted from sea water is %0.2f",f_ice)
```

**Scilab code Exa 7.5** Calculate proeutectoid ferrite and eutectoid ferrite in 6 tenth percent steel

```
// Calculate proeutectoid ferrite and eutectoid
   ferrite in 0.6% steel
```

```
clc
a = 0 // limiting value
b = 0.8 // limiting value
c = 0.6 // percentage composition of carbon
f = 0.88 // fraction of ferrite in a eutectoid steel
printf("\n Example 7.5")
f_pro_alpha = (b-c)/(b-a)
f_perlite = 1 - f_pro_alpha
f_eut = f*f_perlite
printf("\n Composition of proeutectoid ferrite is %0
   .2f",f_pro_alpha)
printf("\n Composition of eutectoid ferrite is %0.2f
   ",f_eut)
```

# Chapter 8

# Diffusion in Solids

**Scilab code Exa 8.1** Calculate the rate at which hydrogen escapes through the walls of the steel tank

```
1 // calculate the rate at which hydrogen escapes
     through the walls of the steel tank
2 clc
3 t = 5 // thickness in mm
4 c = 10 // concentration
5 D = 1e-9 // diffusion coefficient
6 printf("\n Example 8.1")
7 j  = D*c/(t*1e-3)
8 printf("\n Outward flux is %.0e kg m^-2 s^-1",j)
```

**Scilab code Exa 8.2** Calculate maximum time till which material can be kept at 550 degree Celsius

```
1 // Calculate maximum time till which material can be
     kept at 550 degree Celsius
2 clc
```

```
3  D_0 = 0.24e-4 // diffusion coefficient
4  Q = 121e3
5  R = 8.314// Universal gas constant
6  T = 550 // temperature in Celsius
7  k = 0.2 // thickness of pure Al sheet in mm
8  d = 0.1 // penetration depth in mm
9  c_x = 0.4 // concentration in percentage
10 A  = 2 // Constant in percentage
11 B = 2// Constant in percentage
12 printf("\n Example 8.2")
13 x = d-k
14 D_cu_al = D_0*exp(-Q/(R*(T+273)))
15 k = (A-c_x)/B
16 if k ==0.8 then
17     z = 0.9 // from table
18 end
19 t = (x*1e-3)^2/(z^2*4*D_cu_al)// time in sec
20
21 printf("\n Material can be kept at %d degree Celsius
       for nearly %d minute",T,t/60)// answer in book
     is 100 min
```

**Scilab code Exa 8.3** Calculate minimum depth up to which post machining is to be done

```
1  // Calculate minimum depth up to which post
     machining is to be done
2  clc
3  D_0 = 0.7e-4 // diffusion coefficient
4  Q = 157 // Energy in kJ mol^-1, considered from
     table 8.2
5  R = 8.314// Universal gas constant
6  T = 950 // temperature in Celsius
```

```
c2 = 0.8 // concentration in percentage
cs = 0 // concentration in percentage
c_x = 0.6// concentration in percentage
t = 4 // time in hours
a = 1 //let
printf("\n Example 8.3")
A = cs
B = c2-cs
D = D_0*exp(-Q*1e3/(R*(T+273)))
k = erf(((A-c_x)/B))*-1
if k >0.7 then
    if k<0.712 then
            z = 0.81 // from table
    end

end
x = z*2*sqrt(D*t*3600)

printf("\n Depth up to which machining is required
   is nearly %.2f mm",x*1e3)
// numerical value of answer in book is 0.75
```

**Scilab code Exa 8.4** Calculate time required to get required boron concentration

```
// Calculate time required to get required boron
   concentration
clc
D = 4e-17 // diffusion coefficient
c1 = 0
cs = 3e26
c_x = 1e23 // number of atoms
x = 2e-6 // depth in m
```

```
printf("\n Example 8.4")
A = cs
B = cs - c1
k = (A-c_x)/B
if k >0.99966 then
    if k< 0.9997 then
        z = 2.55 // from table
    end
end
t = x^2/(z^2*4*D)// time in sec

printf("\n Time required to get required boron
   concentration is %d sec",t)// answer in book is
   3845 sec
```

**Scilab code Exa 8.5** Calculate ratio of cross sectional areas

```
// Calculate ratio of cross sectional areas
clc
r = 10 // radius in mm
t = 4 // thickness in angstrom
printf("\n Example 8.5")
r = 2*%pi*r*1e-3*t*1e-10/(%pi*(r*1e-3)^2)
printf("\n Ratio of cross sectional areas is %.0e ",
   r)
```

# Chapter 9

# Phase Transformation

**Scilab code Exa 9.1** Calculate the critical free energy of nucleation of ice from water and critical radius

```
1 // Calculate the critical free energy of nucleation
     of ice from water and critical radius
2 clc
3 del_t1 = 0// temperature difference in degree
     Celsius
4 del_t2 = -5 // temperature difference in degree
     Celsius
5 del_t3 = -40 // temperature difference in degree
     Celsius
6 del_h = 6.02 // enthalpy of fusion in kJ/mol
7 T_m = 273 // mean temperature
8 Gamma = 0.076 // energy of ice water interface in J
     /m^2
9 v = 19 // molar volume of ice
10 printf("\n Example 9.1")
11 printf("\n Part A")
12 printf("\n At del_t = %d, there is no supercooling.
     So there is no critical radius",del_t1)
13 printf("\n\n Part B")
14 del_f = 16/3*%pi*(Gamma)^3*T_m^2/((del_h*1e3*1e6/v)
```

```
    ^2*del_t2^2)
15 r = 2*T_m*Gamma/(-del_h*1e3*1e6/v*del_t2)
16 printf("\n Critical free energy of nucleation is %.1
    eJ",del_f)
17 printf("\n Critical radius is %d angstrom",ceil(r*1
    e10))
18 printf("\n\n Part C")
19 temp_r = del_t3/del_t2
20 del_f_ = del_f/temp_r^2
21 r_ = r/temp_r
22
23  printf("\n Critical free energy of nucleation is %
    .1eJ",del_f_)
24 printf("\n Critical radius is %d angstrom.", ceil(r_
    *1e10))
```

**Scilab code Exa 9.2** Calculate the change in delta f required to increase nucleation rate

```
1 // Calculate the change in del_f required to
    increase nucleation rate
2 clc
3 T= 300 // temperature in kelvin
4 R = 8.314 // universal gas constant
5 k = 2.303 // conversion factor
6 a1 = 1e42
7 a2 = 1e6 // nucleation rate
8 a3 = 1e10 // nucleation rate
9 printf("Example 9.2")
10 I1 = (log10(a1)-log10(a2))*k*R*T // exponent factor
11 I2 = (log10(a1)-log10(a3))*k*R*T// exponent factor
12 del_f = I1-I2 // difference
13 a = 10^(log10(a3)-log10(a2))
```

```
14
15 printf("\n A change of %d KJ mol^-1 energy is
      required to increase nucleation factor from \n %
      .0e m^-3 s^-1 to %.0e m^-3 s^-1 ",ceil(del_f/1
      e3),a,a3)
```

**Scilab code Exa 9.4** Calculate delta f of heterogeneous as a fraction of delta f of homogeneous

```
1 // Calculate del_f_het as a fraction of del_f_homo
2 clc
3 Gamma_alpha_del = 0.5 // in J m^-2
4 Gamma_alpha_beta = 0.5 // in J m^-2
5 Gamma_beta_del = 0.01 // in J m^-2
6
7 printf("\n Example 9.4")
8 theta = acos((Gamma_alpha_del -Gamma_beta_del)/
      Gamma_alpha_beta)
9 del_f_ratio = 1/4*(2-3*cos(theta)+(cos(theta))^3)
10
11
12 printf("\n del_f_het is %0.4fth fraction of
      del_f_homo.",del_f_ratio)
```

**Scilab code Exa 9.6** Calculate the free energy change during recrystallization

```
1 // Calculate the free energy change during
      recrystallization
```

```
clc
mu = 45.5e9
b = 2.55e-10
n1 = 1e9 // initial dislocation density
n2 = 1e13 // final dislocation density
printf("\n Example 9.6")
E = 1/2*mu*b^2*n2
del_g = E // as difference between initial and final
    dislocation energy is four order magnitude
printf("\n Free energy change during
   recrystallization is %d J m^-3",-del_g)
// Numerical value of answer in book is 14800
```

# Chapter 10

# Elastic Anelastic and Viscoelastic Behaviour

**Scilab code Exa 10.1** Estimate Youngs modulus of material

```
1 // Estimate young s modulus of material
2 clc
3 n = 1
4 m = 9
5 A = 7.68e-29 // Constant having unit J m
6 r_0 = 2.5e-10 // bonding distance in m
7 printf("\n Example 10.1")
8 B = A*r_0^8/9
9
10 Y = (90*B/(r_0)^11-2*A/(r_0)^3)/r_0
11
12 printf("\n Young s modulus of material is %d GN m
   ^-2",Y/1e9)
```

**Scilab code Exa 10.2** Calculation of stress in fibers

```
1 // Calculation of stress in fibers
2 clc
3 Y_f = 440
4 Y_m = 71
5 sigma_total= 100 // total load
6 printf("\n Example 10.2")
7 r = Y_f/Y_m
8 sigma_f = r*(sigma_total/0.7)/(1+r*3/7)
9 printf("\n Part A:")
10 printf("\n When load is applied parallel to fiber
    then, stress in fiber is %d MN m^-2",sigma_f)
11
12 printf("\n\n Part B:")
13 printf("\n When load is applied perpendicular to
    fiber then, stress in fiber and matrix is same i.
    e. %d MN m^-2",sigma_total)
```

**Scilab code Exa 10.3** Estimate diffusion coefficient

```
1 // Estimate diffusion coefficient
2 clc
3 t_r = 100 // relaxation time in s
4 d = 2.5 // distance in angstrom
5 printf("\n Example 10.3")
6 f = 1/t_r // jump frequency
7 D = (d*1e-10)^2*f
8 printf("\n Diffusion coefficient is %.2e m^2 s^-1",D
    )
```

# Chapter 11

# Plastic Deformation and Creep in Crystalline Materials

**Scilab code Exa 11.2** Calculate the stress required to move the dislocation at given temperature

```
1 // Calculate the stress required to move the
     dislocation at given temperature
2 clc
3 b = 2 // burger vector in angstrom
4 v = 20*b^3 // activation volume
5 tau_pn = 1000 // P-N stress of crystal in MNm^-2
6 k = 1.38e-23 // physical constant
7 t1 = 0 // temperature in K
8 t2 = 100// temperature in K
9 t3 = 300// temperature in K
10 t4 = 500// temperature in K
11 printf("\n Example 11.2")
12 printf("\n\n Part A:")
13 T = t1
14 tau_app = tau_pn - 40*k*T/(v*1e-30)
15 printf("\n The stress required to move the
     dislocation at temperature %dK is %d MNm^-2",T,
     tau_app)
```

```
16 printf("\n\n Part B:")
17 T = t2
18 tau_app = tau_pn - 40*k*T/(v*1e-30*1e6)
19 printf("\n The stress required to move the
      dislocation at temperature %dK is %d MNm^-2",T,
      tau_app)
20 printf("\n\n Part C:")
21 T = t3
22 tau_app = tau_pn - 40*k*T/(v*1e-30*1e6)
23 if tau_app<0 then
24     printf("\n Stress to be applied is zero")
25     printf("\n The stress required to move the
          dislocation at temperature %dK entirely
          overcome by thermal fluctuations", T)
26 end
27 printf("\n\n Part D:")
28 T = t4
29 tau_app = tau_pn - 40*k*T/(v*1e-30*1e6)
30 if tau_app<0 then
31     printf("\n Stress to be applied is zero")
32     printf("\n The stress required to move the
          dislocation at temperature %dK entirely
          overcome by thermal fluctuations", T)
33 end
```

**Scilab code Exa 11.3** Calculate the dislocation density in copper

```
1 // Calculate the dislocation density in copper
2 clc
3 mu = 44 // shear modulus of copper in GN m^-2
4 b = 2.55 // burgers vector in angstrom
5 tau = 35 // shear stress in MN m^-2
6 printf("Example 11.3")
```

```
l = mu*1e9*b*1e-10/(tau*1e6)
rho = 1/l^2

printf("\n Dislocation density in copper is %.1e m^-2",rho)
// Answer in book is 1e12 m^-2
```

**Scilab code Exa 11.4** Find the yield stress for a grain size of ASTM 9

```
//  Find the yield stress for a grain size of ASTM 9
clc
sigma1 = 120 // initial yield strength of material in MNm^-2
sigma2 = 220 // Final yield strength of material in MN m^-2
d1 = 0.04 // initial diameter in mm
d2 = 0.01 // final diameter in mm
n = 9 // astm number
printf("Example 11.4")
k = (sigma2-sigma1)*1e6/(1/sqrt(d2*1e-3)-1/sqrt(d1*1e-3))
sigma_i = sigma1*1e6 -k/sqrt((d1*1e-3))
d = 1/sqrt(2^(n-1)*1e4/645)
sigma_y = sigma_i+k*(d*1e-3)^(-0.5)

printf("\n Yield stress for a grain size of ASTM 9 is %d MN m^-2",ceil(sigma_y/1e6))
```

**Scilab code Exa 11.5** Estimate the change in yield strength

```
// Estimate the change in yield strength
clc
n1 = 1e6 // initial number of particles
n2 = 1e3 // final number of particle
printf("\n Example 11.5")
k = (n1/n2)^(1/3)
printf("\n Yield strength would have decreased to
   %d%% of its initial value.",100/k)
```

# Chapter 12

# Fracture

**Scilab code Exa 12.1** Estimate fracture strength

```
1 // Estimate fracture strength
2 clc
3 c = 2 // crack of half length in micro meter
4 Y = 70 // young s modulus in GN m^-2
5 Gamma = 1 // specific surface energy
6 printf("\n Example 12.1")
7 sigma_f = sqrt(2*Gamma*Y*1e9/(%pi*c*1e-6))/1e6
8 r = Y*1e3/sigma_f
9 printf("\n Fracture strength of %d MN m^-2 is 1/%dth
    of young s modulus. ",ceil(sigma_f),ceil(r
   /100)*100)
10 printf("\n Thus Griffiths criterion bridges the gap
   between the \n observed and ideal strengths of
   brittle material")
```

**Scilab code Exa 12.2** Estimate the brittle fracture strength at low temperatures

```
// Estimate the brittle fracture strength at low
   temperatures
clc
Gamma = 1.5// specific surface energy in J/m^2
Y = 200 // Young s modulus in GN/m^2
c = 2 // half length of crack

printf("\n Example 12.2")
sigma_f = sqrt(2*Gamma*Y*1e9/(%pi*c*1e-6))

printf("\n Brittle fracture strength at low
   temperatures is %d MNm^-2 ",sigma_f/1e6)// answer
    in book is 310MNm^-2
```

**Scilab code Exa 12.3** Estimate the temperature at which the ductility of brittle transition occurs at given strain rates

```
// Estimate the temperature at which the ductility
   of brittle transition occurs at given strain
   rates
clc
Gamma = 2// specific surface energy in J/m^2
Y = 350 // Young s modulus in GN/m^2
c = 2 // half length of crack
de_dt1 = 1e-2 // strain rate
de_dt2 = 1e-5 // strain rate
printf("\n Example 12.3")
printf("\n Part A:")
sigma_f = sqrt(2*Gamma*Y*1e9/(%pi*c*1e-6))
sigma_y = sigma_f/1e6
T = 173600/(sigma_y-20.6-61.3*log10(de_dt1))//
   temperature calculation

```

```
14 printf("\n Transition temperature for strain rate %
      .0e s^-1 is %d K",de_dt1,T)// answer in book is
      300 K
15 printf("\n\n Part B:")
16
17 T = 173600/(sigma_y-20.6-61.3*log10(de_dt2))//
      temperature calculation
18
19 printf("\n Transition temperature for strain rate %
      .0e s^-1 is %d K",de_dt2,T)// answer in book is
      230 K
20 // Solution in book for two parts is divided into
      three parts
```

# Chapter 13

# Oxidation and Corrosion

**Scilab code Exa 13.2** Calculation of required quantity of magnesium

```
1 // Calculation of required quantity of magnesium
2 clc
3 j = 15 // current density in mA m^-2
4 m = 0.0243 // molar mass of magnesium
5 F = 96490 // farad charge
6 n = 2 // charge on ion
7 t = 10 // time in years
8 printf("\n Example 13.2")
9 a = m*j*1e-3*(t*365*24*3600)/(n*F)
10 printf("\n Amount of magnesium required is %0.1f kg
   m^-2",a)
```

# Chapter 14

# Conductors and Resistors

**Scilab code Exa 14.1** Calculate energy difference

```
1 // Calculate energy difference
2 clc
3 n_x1 = 1 // atomic level
4 n_y1 = 1// atomic level
5 n_z1 = 1// atomic level
6 n_x2 = 2// atomic level
7 L = 10 // lattice parameter in mm
8 h = 6.626e-34 // plank constant
9 m_e = 9.109e-31 // mass of electron in kg
10 printf("\n Example 14.1")
11 E1 = h^2*(n_x1^2+n_y1^2+n_z1^2)/(8*m_e*(L*1e-3)^2)
12 E2 = h^2*(n_x2^2+n_y1^2+n_z1^2)/(8*m_e*(L*1e-3)^2)
13 E = E2-E1 // energy difference
14 printf("\n Energy difference is %.2e J ",E)
```

**Scilab code Exa 14.2** Calculate conductivity of copper at 300 K

```
// Calculate conductivity of copper at 300 K
clc
tau = 2e-14 // collision time of electron in s
e = 1.602e-19 // charge on electron
m_e = 9.1e-31 // mass of electron in kg

printf("\n Example 14.2")
n = 6.023e23*8960/0.06354

sigma= n*e^2*tau/m_e
printf("\n Conductivity of copper at 300 K is %.1e
   ohm^-1 m^-1 ",sigma)
```

**Scilab code Exa 14.3** Estimation of resistivity due to impurity scattering of 1 percent of Nickel in copper lattice

```
// Estimation of resistivity due to impurity
   scattering of 1% of Nickel in copper lattice
clc
r_cu = 1.8e-8 // resistivity of pure copper in ohm-m
r_Ni_cu = 7e-8 //resistivity of copper 4% Ni in ohm-
   m
per1 = 4//impurity in percent
per2 = 1 // impurity in percent
printf("\n Example 14.3")
r = (r_Ni_cu-r_cu)*per2/per1 // resistivity of
   copper 1% Ni in ohm-m
printf("\n Resistivity due to impurity scattering of
    1 %% of Nickel in copper lattice is %.1e ohm-m",
   r)
```

# Chapter 15

# Semiconductors

**Scilab code Exa 15.1** Calculate intrinsic carrier density

```
1 // Calculate intrinsic carrier density
2 clc
3 rho = 3000 // resistivity in ohm m
4 mu_e = 0.14
5 mu_h = 0.05
6 e = 1.602e-19 // charge on electron
7 printf("\n Example 15.1")
8 sigma = 1/rho
9 n = sigma/((mu_e+mu_h)*e)
10 printf("\n Intrinsic carrier density is %.3e m^-3",n
   )
```

# Chapter 16

# Magnetic Materials

**Scilab code Exa 16.1** Calculate the net magnetic moment per iron atom in crystal

```
1 // Calculate the net magnetic moment per iron atom
     in crystal
2 clc
3 a = 2.87 // lattice parameter in angstrom
4 n = 2 // number of atoms per unit cell
5 m = 1750 // Saturation magnetization in kAmˆ-1
6 mu = 9.273e-24 // bohr magneton
7 printf("\n Example 16.1")
8 m_atom = m*1e3*(a*1e-10)^3 /n
9 mu_b = m_atom/mu
10
11 printf("\n Net magnetic moment per iron atom in
     crystal is %.3e Amˆ2",m_atom)
12 printf("\n In unit of mu_b, Net magnetic moment per
     iron atom in crystal is %.1f ",mu_b)
```

**Scilab code Exa 16.2** Comparison of saturation temperatures

```
1 // Comparison of saturation temperatures
2 clc
3 t1 = 0 // temperature in kelvin
4 t2 = 300 // temperature in kelvin
5 m_net_Gd = 7 // net magnetic moment of gadolinium
6 m_net_Co = 1.7 // net magnetic moment of cobalt
7 t_c_Gd = 289 // curie temperature for Gd
8 printf("\n Example 16.2")
9 printf("\n Part A:")
10 if m_net_Gd> m_net_Co then
11 printf("\n At %d K, Net magnetic moment of
   gadolinium i.e. %d is greater than net magnetic
   moment of cobalt i.e. %.1f ",t1,m_net_Gd,m_net_Co
   )
12 printf("\n So, Gd will have higher saturation
   magnetization")
13 end
14 printf("\n\n Part B:")
15 if t_c_Gd<t2 then
16     printf("\n At temperature %d K, Gd is above its
       curie temperature of %dK",t2,t_c_Gd)
17     printf("\n Gd will be paramagnetic at %d K and
       will have negligible magnetization\n as
       compared to Co, which has higher curie
       temperature",t2)
18 end
```

**Scilab code Exa 16.4** Calculation of hysteresis loss

```
1 // Calculation of hysteresis loss
2 clc
```

```
3 v = 0.01 // volume in m^3
4 x = 1e-4 // axis intercept
5 y = 1e2 // axis intercept
6 a = 60000 // Hysteresis loop area
7 f = 50 // frequency in Hz
8 printf("\n Example 16.4")
9 e = x*y*a // Energy loss in one loop
10 E = e*v // energy loss in core in one cycle
11 P = E*f // Power loss
12 printf("\n Power loss due to hysteresis is %d W",P)
```

**Scilab code Exa 16.5** Calculation of eddy current loss at normal voltage and frequency

```
1 // Calculation of eddy current loss at normal
   voltage and frequency
2 clc
3 Total1 = 2300 // total iron loss in W at 440 V and
   50 Hz
4 Total2 = 750 // total iron loss in W at 220 V and 25
    Hz
5 printf("\n Example 16.5")
6 W_e = 1/2*(Total1-2*Total2)
7 printf("\n Eddy current loss at normal voltage and
   frequency is %dW",4*W_e)
```

# Chapter 17

# Dielectric Materials

**Scilab code Exa 17.1** Calculation of relative dielectric constant

```
1 // calculation of relative dielectric constant
2 clc
3 l= 10// length of capacitor in mm
4 b = 10 // width of capacitor in mm
5 d = 2 // distance of separation in mm
6 c = 1e-9 // capacitance in farad
7 epsilon_0 = 8.85e-12 // permittivity of free space
8 printf("\n Example 17.1")
9 epsilon_r = c*d*1e-3/(epsilon_0*l*1e-3*b*1e-3)
10
11 printf("\n Relative dielectric constant is %d",
   epsilon_r)
```

**Scilab code Exa 17.2** Calculate the polarization of a BaTio3 crystal

```
1 // calculate the polarization of a BaTio3 crystal
2 clc
```

```
Ti_shift= 0.06 // shift of TI ion in angstrom
O_shift = 0.06 // shift of oxygen ion of side face
    in angstrom
o_shift = 0.08 //shift of oxygen ion of top and
    bottom faces in angstrom
O_charge  = 2 // unit charge on oxygen ion of side
    face
o_charge  = 2 // unit charge on oxygen ion of top
    and bottom faces
Ti_charge  = 4 // unit charge on titanium ion
n_O = 2 // number of oxygen ion of side face
n_o = 1 // number of oxygen ion of top and bottom
    face
n_Ti = 1 // number of titanium ion
e = 1.6e-19 // amount of one unit charge in coulomb
printf("\n Example 17.2")
p_Ti = n_Ti*Ti_charge *e*Ti_shift*1e-10
p_O = n_O*O_charge*e*O_shift*1e-10
p_o = n_o*o_charge*e*o_shift*1e-10
Total = p_Ti+p_O+p_o
P = Total/(4.03*3.98^2*1e-30)
printf("\n Polarization of BaTiO3 crystal is %.2f Cm
    ^-2 ", P)
```