Scilab Textbook Companion for Applied Physics For Engineers by N. Mehta[1]

Created by
Divya Sharma
Modern Physics
Physics
Shri Mata Vaishno Devi University
College Teacher
Mr. Pankaj Biswas
Cross-Checked by
Chaya Ravindra

March 14, 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 For Engineers

Author: N. Mehta

Publisher: Phi Learning Pvt. Ltd., New Delhi

Edition: 1

Year: 2011

ISBN: 978-81-203-4242-2

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

Relativistic Mechanics

Scilab code Exa 1.2 Lorentz transformations for space and time

```
// Scilab Code Ex1.2: Page:26 (2011)
clc;clear;
c = 3e+008;    // Speed of light in vacuum, m/s
delta_x = 2.45e+03;    // Space difference, m
delta_t = 5.35e-06;    // Time difference, s
v = 0.855*c;    // Speed of frame S_prime, m/s
delta_x_prime = 1/sqrt(1-v^2/c^2)*(delta_x - v*(
    delta_t))*1e-03;    // Distance between two
    flashes as measured in S_prime frame, km
delta_t_prime = 1/sqrt(1-v^2/c^2)*(delta_t - v/c^2*
    delta_x)*1e+006;    // Time between two flashes
    as measured in S_prime
printf("\nThe distance between two flashes as
    measured in S_prime frame = %4.2f km",
    delta_x_prime);
printf("\nThe time between two flashes as measured
    in S_prime frame = %4.2f micro-second",
    delta_t_prime);

// Result
// The distance between two flashes as measured in
```

```
    S_prime frame = 2.08 km
// The time between two flashes as measured in
    S_prime frame = −3.15 micro−second
```

Scilab code Exa 1.4 Relative speed of one photon with respect to another

```
// Scilab Code Ex1.4: Page:27 (2011)
clc;clear;
c = 1;....// Speed of light in vacuum, m/s
u_x_prime = c;    // Velocity of photon as measured
    in S_prime frame, m/s
v = c;    // Velocity of frame S_prime relative to S
     frame, m/s
u_x = (u_x_prime + v)/(1+v*u_x_prime/c^2);
if u_x == 1 then
    ux = 'c';
else
    ux = string(u_x)+'c';
end
printf("\nThe speed of one photon as observed by the
     other is %c", ux);

// Result
// The speed of one photon as observed by the other
    is c
```

Scilab code Exa 1.6 Areal contraction of moving circular lamina

```
// Scilab Code Ex1.6 : Page:28 (2011)
clc;clear;
a = 1;    // For simplicity assume length of semi
    minor axis to be unity, m
c = 3e+08;    // Speed of light, m/s
```

```
v = poly(0, 'v');     // Declare velocity variable, m/s
// As b = a*sqrt(1-v^2/c^2), length of semi-major axis
// Also A_c = %pi*a^2, area of the lamina in its own frame and
// A_e = %pi*a*b, area of the lamina in stationary frame S, so with A_c = A_e
v = roots(1-v^2/c^2 - 1/4);     // Velocity at which surface area of lamina reduces to half in S-frame, m/s
printf("\nThe velocity at which surface area of lamina reduces to half in S-frame = %4.2e", v(1));

// Result
// The velocity at which surface area of lamina reduces to half in S-frame = 2.60e+008
```

Scilab code Exa 1.7 Length of a one metre stick moving parallel to its length

```
// Scilab Code Ex1.7 : Page:29 (2011)
clc;clear;
m0 = 1;     // For simplicity assume the rest mass of stick to be unity, kg
m = 1.5*m0;     // Mass of the moving stick, kg
L0 = 1;     // Assume resting length of the stick to be unity, m
// As m = m0/sqrt(1-v^2/c^2) = m0*gama, solving for gama
gama = m/m0;     // Relativistic factor
L = L0/gama;     // Contracted length of the metre stick, m
printf("\nThe contracted length of the metre stick =
```

```
    %4.2 f  m" , L);

// Result
// The contracted length of the metre stick = 0.67 m
```

Scilab code Exa 1.8 Mean lifetime of meson in motion

```
// Scilab Code Ex1.8: Page:29 (2011)
clc;clear;
c = 3e+008;    // Speed of light in vacuum, m/s
tau0 = 2e-008;    // Mean lifetime of meson at rest,
     m/s
v = 0.8*c;    // Velocity of moving meason, m/s
tau = tau0/sqrt(1-v^2/c^2);    // Mean lifetime of
    meson in motion, m/s
printf("\nThe mean lifetime of meson in motion = %4
    .2e s", tau);

// Result
// The mean lifetime of meson in motion = 3.33e-008
    s
```

Scilab code Exa 1.9 Speed at which a moving clock ticks slow

```
// Scilab Code Ex1.9: Page:30 (2011)
clc;clear;
c = 3e+008;    // Speed of light in vacuum, m/s
delta_t0 = 59;    // Reading of the moving clock for
     each hour, min
delta_t = 60;    // Reading of the stationary clock
    for each hour, min
// As from Time Dilation, delta_t = delta_t0/sqrt(1-
    v^2/c^2), solving for v
```

```
v = c*sqrt(1-(delta_t0/delta_t)^2);
printf("\nThe speed at which the moving clock ticks slow = %4.2e m/s", v);

// Result
// The speed at which the moving clock ticks slow = 5.45e+007 m/s
```

Scilab code Exa 1.10 Distance that the meson beam can travel before reduction in its flux

```
// Scilab Code Ex1.10: Page:30 (2011)
clc;clear;
c = 3e+008;     // Speed of light in vacuum, m/s
tau0 = 2.5e-008;     // Mean lifetime of meson at rest, m/s
v = 0.8*c;     // Velocity of moving meason, m/s
tau = tau0/sqrt(1-v^2/c^2);     // Mean lifetime of meson in motion, m/s
N0 = 1;     // Assume initial flux of meson beam to be unity, watt/Sq.m
N = N0*%e^(-2);     // Meson flux after time t, watt/Sq.m
// As N = N0*e^(-t/tau), which on comparing gives
t = 2*tau;     // Time during which the meson beam flux reduces, s
d = 0.8*c*t;     // The distance that the meson beam can travel before reduction in its flux, m
printf("\nThe distance that the meson beam can travel before reduction in its flux = %2d m", d);

// Result
// The distance that the meson beam can travel before reduction in its flux = 20 m
```

Scilab code Exa 1.11 Velocity of the particle when its total energy is thrice its rest energy

```
// Scilab Code Ex1.11: Page:31 (2011)
clc;clear;
c = 3e+008;     // Speed of light in vacuum, m/s
E0 = 1;     // Rest energy of particle , unit
E = 3*E0;     // Energy of relativistically moving
    particle , unit
// E = m*c^2 and E0 = m0*c^2
// With m = m0/sqrt(1-v^2/c^2), we have
v = c*sqrt(1-(E0/E)^2);     // Velocity of the moving
     particle , m/s
printf("\nThe velocity of the moving particle = %4.2
    e m/s", v);

// Result
// The velocity of the moving particle = 2.83e+008 m
    /s
```

Scilab code Exa 1.12 Kinetic energy and momentum of moving electron

```
// Scilab Code Ex1.12: Page:32 (2011)
clc;clear;
c = 3e+008;     // Speed of light in vacuum, m/s
m0 = 9.1e-031;     // Rest mass of electron , kg
m = 11*m0;     // Mass of relativistically moving
    electron , kg
E_k = (m-m0)*c^2/(1.6e-019*1e+06);     // Kinetic
    energy of moving electron , MeV
// As m = m0/sqrt(1-v^2/c^2), solving for v
```

```
v = c*sqrt(1-(m0/m)^2);      // The velocity of the moving electron, m/s
p = m*v;      // Momentum of moving electron, kg-m/s
printf("\nThe kinetic energy of moving electron = %4.2f MeV", E_k);
printf("\nThe momentum of moving electron = %4.2e kg-m/s", p);

// Result
// The kinetic energy of moving electron = 5.12 MeV
// The momentum of moving electron = 2.99e-021 kg-m/s
```

Scilab code Exa 1.13 Amount of work to be done to increase the speed of an electron

```
// Scilab Code Ex1.13: Page:32 (2011)
clc;clear;
c = 3e+008;      // Speed of light in vacuum, m/s
E0 = 0.5;      // Rest energy of the electron, MeV
v1 = 0.6*c;      // Initial velocity of the electron, m/s
v2 = 0.8*c;      // Final velocity of the electron, m/s
W = (1/sqrt(1-v2^2/c^2)-1/sqrt(1-v1^2/c^2))*E0;    // The amount of work to be done to increase the speed of the electron, MeV
printf("\nThe amount of work to be done to increase the speed of an electron = %4.2e J", W*1e+06*1.6e-019);

// Result
// The amount of work to be done to increase the speed of an electron = 3.33e-014 J
```

Scilab code Exa 1.14 Particle moving with relativistic speed

```
// Scilab Code Ex1.14: Page:33 (2011)
clc;clear;
c = 1;    // Assume speed of light in vacuum to be
    unity, unit
m0 = 1;    // For simplicity assume rest mass of the
     particle to be unity, unit
v = c/sqrt(2);    // Given speed of the particle, m/
    s
gama = 1/sqrt(1-v^2/c^2);    // Relativistic factor
m = gama*m0;    // The relativistic mass of the
    particle, unit
p = m*v;    // The relativistic momentum of the
    particle, unit
E = m*c^2;    // The relativistic total eneryg of
    the particle, unit
E_k = (m-m0)*c^2;    // The relativistic kinetic
    energy of the particle, unit
printf("\nThe relativistic mass of the particle = %5
    .3fm0", m);
printf("\nThe relativistic momentum of the particle
    = %1.0gm0c", p);
printf("\nThe relativistic total energy of the
    particle = %5.3fm0c^2", E);
printf("\nThe relativistic kinetic energy of the
    particle = %5.3fm0c^2", E_k);

// Result
// The relativistic mass of the particle = 1.414m0
// The relativistic momentum of the particle = 1m0c
// The relativistic total energy of the particle =
    1.414m0c^2
// The relativistic kinetic energy of the particle =
```

```
0.414m0c^2
```

Scilab code Exa 1.15 Speed of the electron in order to have its mass equal to mass of a proton

```
// Scilab Code Ex1.15: Page:34 (2011)
clc;clear;
c = 3e+008;     // Speed of light in vacuum, unit
m0 = 9.1e-031;     // Rest mass of the electron, kg
m = 1.67e-027;     // Rest mass of the proton, kg
// As m = m0/sqrt(1-v^2/c^2), solving for v
v = c*sqrt(1-(m0/m)^2);     // Velocity of the electron, m/s
printf("\nThe velocity of the electron to have its mass equal to mass of the proton = %5.3e m/s", v);

// Result
// The velocity of the electron to have its mass equal to mass of the proton = 3.000e+008 m/s
```

Scilab code Exa 1.17 Classical and relativistic speed of an electron of given kinetic energy

```
// Scilab Code Ex1.17: Page:35 (2011)
clc;clear;
c = 3e+008;     // Speed of light in vacuum, unit
m0 = 9.1e-031;     // Rest mass of the electron, kg
E_k = 0.1*1e+006*1.6e-019;     // Kinetic energy of the electron, J
v = sqrt(2*E_k/m0);     // Classical speed of the electron, m/s
```

```
printf("\nThe classical speed of the electron = %5.3e m/s", v);
// As E_k = (m-m0)*c^2 = (1/sqrt(1-v^2/c^2)-1)*m0*c^2, solving for v
v = c*sqrt(1-(m0*c^2/(E_k+m0*c^2))^2);     // Relativistic speed of the electron, m/s
printf("\nThe relativistic speed of the electron = %5.3e m/s", v);

// Result
// The classical speed of the electron = 1.875e+008 m/s
// The relativistic speed of the electron = 1.644e+008 m/s
```

Chapter 2

Quantum Mechanics

Scilab code Exa 2.1 de broglie wavelength of an electron

```
// Scilab Code Ex2.1: Page:79 (2011)
clc;clear;
V = 50;....// Given potential difference, V
lambda = 12.24/sqrt(V);....// Wavelength of the
    light, angstrom
printf("\nThe de-broglie wavelength of electron = %4
    .2f angstrom", lambda);

// Result
// The de-broglie wavelength of electron = 1.73
    angstrom
```

Scilab code Exa 2.2 de broglie wavelength associated with a proton

```
// Scilab Code Ex2.2: Page:79 (2011)
clc;clear;
h = 6.62e-34;    // Planck's constant, J-s
m0 = 1.6e-27;    // Rest mass of proton, kg
```

```
c = 3e+8;     // Speed of light, in m/s
v = c/20;     // Velocity of the proton, in m/s
lambda = (h*sqrt(1-v^2/c^2))/(m0*v);
printf("\nThe de broglie wavelength associated with
    the proton = %4.2e m",lambda);

// Result
//  The de broglie wavelength associated with the
    proton = 2.75e-14 m
```

Scilab code Exa 2.3 Wavelength of the matter wave associated with a proton

```
// Scilab Code Ex2.3: Page:79 (2011)
clc;clear;
c = 3e+8;....// Speed of light, m/s
v = 2e+8;....// Velocity of the proton, m/s
m0 = 1.6e-27;....// Rest mass of proton, kg
h = 6.62e-34;....// Plancks constant,J-s
lambda = (h*sqrt(1-v^2/c^2))/(m0*v);
printf("\nThe wavelength of matter wave associated
    with the proton = %5.3e m", lambda);

// Result
//  The wavelength of matter wave associated with
    the proton = 1.542e-15 m
```

Scilab code Exa 2.5 Uncertainity in determining the position of the electron

```
// Scilab Code Ex2.5: Page:80 (2011)
clc;clear;
a = 0.003;....// Accuracy of the electron,in percent
```

```
s = 5e+03;....// Speed of the electron ,in m/s
del_v = (a/100)*s;....// Change in velocity ,in m/s
m0 = 9.1e-31;....// Rest mass of the electron ,in kg
hcut = 1.054e-34;....// Plancks constant ,J-s
del_x = hcut/(2*del_v*m0);
printf("\nThe uncertainity in the position of the
    electron = %4.2e m", del_x);

// Result
// The uncertainity in the position of the electron
   = 3.86e-004 m
```

Scilab code Exa 2.6 Minimum error in measurement of lifetime of excited state of hydrogen atom

```
// Scilab Code Ex2.6 : Page:81 (2011)
clc;clear;
del_t = 2.5e-14;....// Lifetime of the hydrogen atom
     in excited state
hcut = 1.054e-34;....// Planck's constant ,in J-s
e = 1.6e-19;....// Charge on electron ,in C
del_E = hcut/(2*del_t*e);....// Energy of the state ,
     in eV
printf("\nThe minimum error in measurement of
    lifetime of excited state of hydrogen atom = %6.4
    f eV",del_E);

// Result
// The minimum error in measurement of lifetime of
    excited state of hydrogen atom = 0.0132 eV
```

Scilab code Exa 2.7 Uncertainity in the velocity of an electron

```
// Scilab Code Ex2.7 : Page:81 (2011)
clc;clear;
del_x = 1e-09;      // Uncertainty in position of the
    electron , m
m0 = 9.1e-031;....// Rest mass of an electron , kg
hcut = 1.054e-034;....// Planck's constant ,in J-s
del_v = hcut/(2*del_x*m0);....// Uncertainity in
   velocity of the electron
printf("\nThe uncertainity in the velocity of an
   electron = %4.2e m/s",del_v);

// Result
// The uncertainity in the velocity of an electron =
    5.79e+04 m/s
```

Scilab code Exa 2.8 Smallest possible uncertainity in position of an electron

```
// Scilab Code Ex2.8 : Page:81 (2011)
clc;clear;
hcut = 1.054e-34;    // Reduced Planck's constant , Js
v = 3e+07;....// Velocity of the electron , m/s
c = 3e+08;....// Speed of light in vacuum, m/s
m0 = 9.1e-31;....// Rest mass of an electron , kg
del_v = 3e+08;....// Uncertainty in velocity of the
   electron , m/s
del_x = (hcut*sqrt(1-v^2/c^2))/(2*m0*del_v);
printf("\nThe smallest possible uncertainity in
   position of the electron = %6.4f angstrom", del_x
   /1e-010);

// Result
// The smallest possible uncertainity in position of
    the electron = 0.0019 angstrom
```

Scilab code Exa 2.9 Energy of an electron moving in 1D infinetly high potential box

```
//Scilab Code Ex2.9 :Page:82 (2011)
clc;clear;
n = 1;
m0 = 9.1e-031;....// Mass of the electron , kg
a = 1e-10;....// Width of the box , m
h = 6.63e-034;....// Planck's constant , J-s
E = n^2*h^2/(8*m0*a^2);
printf("\n The energy of the electron moving in 1D
    infinetly high potential box = %5.2e J", E);

// Result
//  The energy of the electron moving in 1D
    infinetly high potential box = 6.04e-18 J
```

Scilab code Exa 2.10 Lowest two permitted energy values of an electron

```
// Scilab Code Ex2.10: Page:83 (2011)
clc;clear;
n = [1,2];....// Shell numbers for two lowest
    permitted energy of the electron
m0 = 9.1e-31;....// Mass of the electron , kg
a = 2.5e-10;....// Width of the box , m
h = 6.63e-34;....// Planck's constant , J-s
e = 1.6e-19;....// Charge on electron , C
E = (n^2*h^2)/(8*m0*a^2*e);
printf("\nThe lowest two permitted energy values of
    an electron are");
printf(" %d eV and %d eV respectively", E(1), E(2));

```

```
// Result
// The lowest two permitted energy values of an
   electron are 6 eV and 24 eV respectively
```

Scilab code Exa 2.11 Lowest energy of the neutron confined to the nucleus

```
// Scilab Code Ex2.11: Page:83 (2011)
clc;clear;
m0 = 1.67e-27;....// Rest mass,in kg
a = 1e-14;....//  Size of the box
h = 6.63e-34;....// Planck's constant,in J-s
n = 1;  // Quantum number for lowest energy state
E_n = n^2*h^2/(8*m0*a^2);
printf("\nThe lowest energy of the neutron confined
   to the nucleus = %4.2e J", E_n);

// Result
// The lowest energy of the neutron confined to the
   nucleus = 3.29e-13 J
```

Scilab code Exa 2.12 Energy difference between the ground state and the first excited state for an electron in 1D box

```
// Scilab Code Ex2.12: Page:83 (2011)
clc;clear;
m0 = 9.1e-31;....// Rest mass, kg
a = 1e-10;....// Length of the box, m
h = 6.62e-34;....// Planck's constat, J-s
n1 = 1;....// Ground state
n2 = 2;....// First excited state
e = 1.6e-19;....// Charge on electron, C
E1 = (n1^2*h^2)/(8*m0*a^2*e);
```

```
E2 = (n2^2*h^2)/(8*m0*a^2*e);
del_E = E2-E1;
printf("\nThe energy difference between the ground
   state and the first excited state = %5.1f eV",
   del_E);

//Result
// The energy difference between the ground state
   and the first excited state = 112.9 eV
```

Chapter 3

Statistical Mechanics

Scilab code Exa 3.1 Probability of existence of oxygen molecules within the given velocity range

```
// Scilab Code Ex3.1: Page:132 (2011)
clc;clear;
m = 5.32e-26;    // Mass of one oxygen molecule, kg
k_B = 1.38e-23;   // Boltzmann constant, J/K
T = 200;     // Temperature of the system, K
v = 100;     // Speed of the oxygen molecules, m/s
dv = 1;      // Increase in speed of the oxygen
   molecules, m/s
P = 4*%pi*(m/(2*%pi*k_B*T))^(3/2)*exp(-m*v^2/(2*k_B*
   T))*v^2*dv;
printf("\nThe probability that the speed of oxygen
   molecule is %4.2e", P) ;

// Result
// The probability that the speed of oxygen molecule
   is 6.13e-04
```

Scilab code Exa 3.2 Probability that the speed of oxygen molecules

```
// Scilab Code Ex3.2 : Page:132 (2011)
clc;clear;
A = 32;      // Gram atomic mass of oxygen, g/mol
N_A = 6.023e+026;    // Avogadro's number, per kmol
m = A/N_A;....//mass of the molecule, kg
k_B = 1.38e-23;....// Boltzmann constant, J/K
T = 273;....// Temperature of the gas, K
v_av = 1.59*sqrt(k_B*T/m);....// Average speed of
    oxygen molecule, m/s
printf("\nThe average speed of oxygen molecule is =
    %3d m/s", v_av);
v_rms = 1.73*sqrt(k_B*T/m);....// The mean square
    speed of oxygen molecule, m/s
printf("\nThe root mean square speed of oxygen gas
    molecule is = %3d m/s", ceil(v_rms))
v_mp = 1.41*sqrt(k_B*T/m);....// The most probable
    speed of oxygen molecule, m/s
printf("\nThe most probable speed of oxygen molecule
     is = %3d m/s", ceil(v_mp));

// Result
// The average speed of oxygen molecule is = 423 m/s
// The root mean square speed of oxygen gas molecule
     is = 461 m/s
// The most probable speed of oxygen molecule is =
    376 m/s
```

Scilab code Exa 3.3 Temperature to produce invariant average speed of hydrogen molecule

```
// Scilab Code Ex3.3: Page:133 (2011)
clc;clear;
m_H = 2;     // Gram molecular mass of hydrogen, g
m_O = 32;    // Gram molecular mass of oxygen, g
k_B = 1.38e-23;....// Boltzmann constant, J/K
```

```
v_avO = 1;....// For simplicity average speed of oxygen gas molecule is assumed to be unity, m/s
v_avH = 2*v_avO;....// The average speed of hydrrogen gas molecule, m/s
T_O = 300;  // Temperature of oxygen gas, K
// As v_avO/v_av_H = sqrt(T_O/T_H)*sqrt(m_H/m_O), solving for T_H
T_H = (v_avH/v_avO*sqrt(m_H/m_O)*sqrt(T_O))^2; // Temperature at which the average speed of hydrogen gas molecules is the same as that of oxygen gas molecules, K
printf("\nTemperature at which the average speed of hydrogen gas molecules is the same as that of oxygen gas molecules at 300 K = %2d", T_H);

// Result
// Temperature at which the average speed of hydrogen gas molecules is the same as that of oxygen gas molecules at 300 K = 75
```

Scilab code Exa 3.4 Fraction of oxygen gas molecules within one percent of most probable speed

```
// Scilab Code Ex3.4: Page:133 (2011)
clc;clear;
v_mp = 1;   // Most probable speed of gas molecules, m/s
dv = 1.01*v_mp-0.99*v_mp;   // Change in most probable speed, m/s
v = v_mp;   // Speed of the gas molecules, m/s
Frac = 4/sqrt(%pi)*1/v_mp^3*exp(-v^2/v_mp^2)*v^2*dv;
printf("\nThe fraction of oxygen gas molecules within one percent of most probable speed = %5.3f", Frac);

```

```
// Result
// The fraction of oxygen gas molecules within one
   percent of most probable speed = 0.017
```

Scilab code Exa 3.5 Most probable distribution of 5 distinguishable particles among 3 cells

```
// Scilab Code Ex3.5: Page:134 (2011)
clc;clear;
n = 5;  // Number of distinguishable particles which
    are to be distributed among cells
n1 = [5 4 3 3 2];   // Possible occupancy of
   particles in first cell
n2 = [0 1 2 1 2];   // Possible occupancy of
   particles in second cell
n3 = [0 0 0 1 1];   // Possible occupancy of
   particles in third cell
BIG_W = 0;
printf("\n_____________________________________");
printf("\nn1      n2      n3      5/(n1!n2!n3!)");
printf("\n_____________________________________");
for i = 1:1:5
W = factorial(n)/(factorial(n1(i))*factorial(n2(i))*
   factorial(n3(i)));
if BIG_W < W then
    BIG_W = W;
    ms = [n1(i) n2(i) n3(i)];
end
printf("\n%d      %d      %d            %d", n1(i), n2
   (i), n3(i), W);
end
printf("\n_____________________________________");
printf("\nThe macrostates of most probable
   distribution with thermodynamic probability %d
   are:", BIG_W);
```

```
printf("\n(%d, %d, %d), (%d, %d, %d) and (%d, %d, %d
    )", ms(1), ms(2), ms(3), ms(2), ms(3), ms(1),ms
    (3), ms(1), ms(2));

// Result
// -------------------------------------
// n1       n2       n3       5/(n1!n2!n3!)
// -------------------------------------
//5        0        0            1
// 4        1        0            5
// 3        2        0            10
// 3        1        1            20
// 2        2        1            30
// -------------------------------------
// The macrostates of most probable distribution
    with thermodynamic probability 30 are:
// (2, 2, 1), (2, 1, 2) and (1, 2, 2)
```

Scilab code Exa 3.6 Thermodynamic probability of the macrostate of distributing 8 distinguishable particles in 2 compartments

```
// Scilab Code Ex3.6: Page:135 (2011)
clc;clear;
g1 = 4;     // Intrinsic probability of first cell
g2 = 2;     // Intrinsic probability of second cell
k = 2;      // Number of cells
n = 8;     // Number of distinguishable particles
n1 = 8;     // Number of cells in first compartment
n2 = n - n1;     // Number of cells in second
    compartment
W = factorial(n)*1/factorial(n1)*1/factorial(n2)*(g1
    )^n1*(g2)^n2;
printf("\nThe thermodynamic probability of the
    macrostate (8,0) = %5d", W);

```

```
// Result
// The thermodynamic probability of the macrostate
   (8,0) = 65536
```

Scilab code Exa 3.7 Three particles obeying Bose Einstein statistics distributed in three cells

```
// Scilab Code Ex3.7 : Page:135 (2011)
clc;clear;
function str = st(val)
    str = emptystr();
    if val == 3 then
        str = 'aaa';
    elseif val == 2 then
        str = 'aa';
    elseif val == 1 then
        str = 'a';
    elseif val == 0 then
        str = '0';
    end
endfunction

g = 3; // Number of cells in first compartment
n = 3; // Number of bosons
p = 3;
r = 1;   // Index for number of rows
clc;
printf("\nAll possible meaningful arrangements of
   three particles in three cells are:")
printf("\n__________________________");
printf("\nCell 1     Cell 2     Cell 3");
printf("\n__________________________");
for i = 0:1:g
    for j = 0:1:n
        for k = 0:1:p
```

```
            if (i+j+k == 3) then
      printf("\n%4s      %4s       %4s", st(i), st(j), st(k));
            end
        end
    end
end
printf("\n___________________________");

// Result
// All possible meaningful arrangements of three particles in three cells are:
// ---------------------------
// Cell 1     Cell 2     Cell 3
// ---------------------------
//   0          0         aaa
//   0          a          aa
//   0          aa          a
//   0         aaa          0
//   a          0          aa
//   a          a           a
//   a          aa          0
//   aa         0           a
//   aa         a           0
//  aaa         0           0
// ---------------------------
```

Scilab code Exa 3.8 Probability of macrostate

```
// Scilab Code Ex3.8 : Page:136 (2011)
clc;clear;
g1 = 3; // Number of cells in first compartment
g2 = 4; // Number of cells in second compartment
k = 2;  // Number of compartments
n1 = 5; // Number of bosons
```

```
n2 = 0; // Number of with no bosons
W_50 = factorial(g1+n1-1)*factorial(g2+n2-1)/(
    factorial(n1)*factorial(g1-1)*factorial(n2)*
    factorial(g2-1));
printf("\nThe probability for the macrostate (5,0)
    is = %2d", W_50);

// Result
// The probability for the macrostate (5,0) is = 21
```

Scilab code Exa 3.11 Fermi energy of the Na at absolute zero

```
// Scilab Code Ex3.11: Page:138 (2011)
clc;clear;
r = 1.86e-10;....// Radius of Na, angstrom
m = 9.1e-31;....// Mass of electron ,in kg
h = 6.62e-34;....// Planck's constant , J-s
N = 2;....// Number of free electrons in a unit cell
     of Na
a = 4*r/sqrt(3);....// Volume of Na, m
V = a^3;....// Volume of the unit cell of Na, meter
    cube
E = h^2/(2*m)*(3*N/(8*%pi*V))^(2/3);
printf("\nThe fermi energy of the Na at absolute
    zero is = %4.2e J", E);

// Result
// The fermi energy of the Na at absolute zero =
    5.02e-019 J
```

Scilab code Exa 3.12 Fermi energy of silver in metallic state

```
// Scilab Code Ex3.12:Page-139 (2011)
```

```
clc;clear;
m = 9.1e-31;....// mass of electron, kg
h = 6.62e-34;....// Planck's constant, J-s
V = 108/10.5*1e-06;....// Volume of 1 gm mole of
    silver, metre-cube
N = 6.023e+023;     // Avogadro's number
E_F = h^2/(2*m)*(3*N/(8*%pi*V))^(2/3);     // Fermi
    energy at absolute zero, J
printf("\nThe fermi energy of the silver at absolute
     zero = %4.2e J",E_F);

// Result
// The fermi energy of the silver at absolute zero =
     8.80e-019 J
```

Scilab code Exa 3.13 Fermi energy of free electrons in cesium

```
// Scilab Code Ex3.14: Electron density in lithium
    at absolute zero: Page:140 (2011)
clc;clear;
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
m = 9.1e-31;....// Mass of the elecron, kg
h = 6.63e-34;     // Planck's constant, Js
EF = 4.72*e;....// Fermi energy of free electrons in
     Li, J
rho = 8*%pi/3*(2*m*EF/h^2)^(3/2);     // Electron
    density at absolute zero, electrons/metre-cube
printf("\nThe electron density in lithium at
    absolute zero = %4.2e electrons/metre-cube", rho)
    ;

// Result
// The electron density in lithium at absolute zero
   = 4.63e+028 electrons/metre-cube
```

Scilab code Exa 3.15 Temperature at which the level above the fermi level is occupied by the electron

```
// Scilab Code Ex3.15: Page:140 (2011)
clc;clear;
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
k_B = 1.38e-023;     // Boltzmann constant, J/K
f_E = 0.01;..../ Probability that a state with energy 0.5 eV above the Fermi energy is occupied by an electron, eV
delta_E = 0.5;     // Energy difference (E-Ef)of fermi energy, eV
// Since f_E = 1/(exp((E-Ef)/(k_B*T))+1), solvinf for T
T = delta_E/(log((1-f_E)/f_E)*k_B/e);   // Temperature at which the level above the fermi level is occupied by the electron, K

printf("\nThe temperature at which the level above the fermi level is occupied by the electron = %4d K", ceil(T));

// Result
// The temperature at which the level above the fermi level is occupied by the electron = 1262 K
```

Chapter 4

Geometrical Optics

Scilab code Exa 4.1 Actual path of light using Fermat principle

```
// Scilab Code Ex4.1: Page:189 (2011)
clc;clear;
// Declare cosine function
function r = cosine(t)
    t = poly(0,t);
    r = 1-t^2/factorial(2)+t^4/factorial(4)-t^6/
       factorial(6)+t^8/factorial(8)-t^10/factorial
       (10)+t^12/factorial(12)-t^14/factorial(14);
endfunction

// Declare sine function
function r = sine(t)
    t = poly(0,t);
    r = t-t^3/factorial(3)+t^5/factorial(5)-t^7/
       factorial(7)+t^9/factorial(9)-t^11/factorial
       (11)+t^13/factorial(13)-t^15/factorial(15);
endfunction

r = 1;  // For convenience assume radius of the
   circle to be unity, unit
thet = poly(0,'thet');  // Declare a variable
```

```
l = 2*r*(cosine('thet')+sine('thet'));   // Length of
    actual path, unit
theta = 45*%pi/180;      // Angle which the chord PQ
    makes with the diameter, radian
d_diff = derivat(derivat(l));    // Double derivative
    of 'l' w.r.t. theta
printf("\nl = %5.3fr", horner(l,theta));
printf("\nDouble_diff of l at theta = 45 degrees =
    %5.3fr \nwhich is negative, so the actual path is
    maximum", horner(d_diff, theta));

// Result
// l = 2.828r
// Double_diff of l at theta = 45 degrees = -2.828r
// which is negative, so the actual path is maximum
```

Scilab code Exa 4.2 Light reflected from the inner surface of spherical shell

```
// Scilab Code Ex4.2: Page:191 (2011)
clc;clear;
r = 1;  // For convenience assume radius of the
    circle to be unity, unit
alpha = 0.8*r;  // Distance of light source from the
    centre of the spherical shell, unit
cos_phi_by_2 = sqrt((alpha+1)/(4*alpha));
printf("\ncos(phi/2) = %d/4", cos_phi_by_2*4);

// Result
// alpha^2+1-2*alpha*cosine('phi')
```

Scilab code Exa 4.3 Equivalent focal length of the combinations of lenses

```
// Scilab Code Ex4.3: Page:193 (2011)
clc;clear;
f1 = 5;....// Focal length of thin convex lens, cm
f2 = 3;....// Focal length of thin convex lens, cm
d = 2;....// Separation between the lenses, cm
F = f1*f2/(f1+f2-d);....// Equivalent focal length
    of a combination of the two lenses, cm
printf("\nThe equivalent focal length of the
    combination of lenses = %3.1f cm", F)

// Result
// The equivalent focal length of the combination of
    lenses = 2.5 cm
```

Scilab code Exa 4.4 Focal length of the combination of lenses of given powers

```
// Scilab Code Ex4.4: Page:194 (2011)
clc;clear;
P1 = 5;....// Power of first converging lens,
    diopter
P2 = 4;....// Power of second converging lens,
    diopter
d = 0.1;....// Separation distance between two
    lenses, cm
P = P1+P2-d*P1*P2;
f = 1/P*100;....// The corresponding value of the
    focal length of the lens combination, cm
printf("\nThe focal length of the combination of
    lenses of given powers = %5.2f cm", f);

// Result
// The focal length of the combination of lenses of
    given powers = 14.29 cm
```

Scilab code Exa 4.5 Focal length of the combination of coaxially placed thi convex lenses

```
// Scilab Code Ex4.5: Page:194 (2011)
clc;clear;
f1 = 30;....// Focal length first convex lens , cm
f2 = -50;....// Focal length of second convex lens ,
    cm
d = 20;....// Separation distance between lenses , cm
F = f1*f2/(f1+f2-d);....// Equivalent focal length
    of a combination of the two lenses , cm
printf("\nThe equivalent focal length of the
    combination = %4.1f cm", F);

// Result
// The equivalent focal length of the combination =
    37.5 cm
```

Scilab code Exa 4.7 Locations of principal points and focal points

```
// Scilab Code Ex4.7 : Page-195
clc;clear;
f1 = 4;....// Focal length of thin convex lens , cm
f2 = 12;....// Focal length of thin convex lens , cm
d = 8;....// Separation distance between the lenses ,
    cm
F = f1*f2/(f1+f2-d);....// Equivalent focal length
    of the combination , cm
L1H1 = d*F/f2;     // Distance of first principal
    point H1 from first lens , cm
printf("\nThe distance of the first principal point
    H1 from the first lens = %d cm", L1H1);
```

```
L2H2 = -d*F/f1;     // Distance of first principal point H2 from second lens, cm
printf("\nThe distance of the second principal point H2 from the second lens = %d cm", L2H2);
L1F1 = -F*(1-d/f2);     // Distance of first focal point F1 from first lens, cm
printf("\nThe distance of the first focal point F1 from the first lens = %d cm", L1F1);
L2F2 = F*(1-d/f1);     // Distance of second focal point F2 from first lens, cm
printf("\nThe distance of the second focal point F2 from the second lens= %d cm", L2F2);

// Result
// The distance of the first principal point H1 from the first lens = 4 cm
// The distance of the second principal point H2 from the second lens = -12 cm
// The distance of the first focal point F1 from the first lens = -2 cm
// The distance of the second focal point F2 from the second lens= -6 cm
```

Scilab code Exa 4.8 Position of principal points and focal points for two coaxially placed lenses

```
// Scilab Code Ex4.8: Page-195 (2011)
clc;clear;
f1 = 25;....// Focal length of thin convex lens, cm
f2 = -15;....// Focal length of thin concave lens, cm
d = 15;....// Separation distance between the lenses, cm
// We know that, F = f1*f2/f1+f2-d then
F = f1*f2/(f1+f2-d);....// The equivalent focal
```

```
length of the combination
L1H1 = d*F/f2;     // The distance of the first principal point H1 from the first lens, cm
printf("\nThe distance of the first principal point H1 from the first lens = %d cm", L1H1);
L2H2 = -d*F/f1;     // The distance of the second principal point H2 from the first lens, cm
printf("\nThe distance of the second principal point H2 from the second lens = %d cm", L2H2);
L1F1 = -F*(1-d/f2);     // The distance of the first focal point F1 from the first lens, cm
printf("\nThe distance of the first focal point H1 from the first lens = %d cm", L1F1);
L2F2 = F*(1-d/f1);     // The distance of the second principal point F2 from the first lens, cm
printf("\nThe distance of the second focal point H2 from the second lens= %d cm", L2F2);

//Result
// The distance of the first principal point H1 from the first lens = -75 cm
// The distance of the second principal point H2 from the second lens = -45 cm
// The distance of the first focal point H1 from the first lens = -150 cm
// The distance of the second focal point H2 from the second lens= 30 cm
```

Scilab code Exa 4.9 Focal lengths from dispersive powers of achromatic combination of lenses

```
// Scilab Code Ex4.9 : Page-196
clc;clear;
w1 = 0.024;....// Magnitude of the dispersive power of first lens
```

```
w2 = 0.036;....// Magnitude of the dispersive power of second lens
// Let 1/f1 = x and 1/f2 = y, then
// The condition for achromatic combination of two lenses, w1/f1 + w2/f2 = 0 => w1*x + w2*y = 0 --- (I)
F = 90;....// Given focal length, cm
// Also F = 1/f1 + 1/f2 => F = x + y    ---- (II)
A = [w1 w2; 1 1];       // Square matrix
B = [0;1/F];     // Column vector
X = inv(A)*B;     // Characteristic roots of the simultaneous equations, cm
f1 = 1/X(1);     // Focal length of convex lens, cm
f2 = 1/X(2);     // Focal length of concave lens, cm

printf("\nThe focal length of convex lens = %2d cm", ceil(f1));
printf("\nThe focal length of concave lens = %2d cm", ceil(f2));

// Result
// The focal length of convex lens = 30 cm
// The focal length of concave lens = -44 cm
```

Scilab code Exa 4.10 Focal length of the two component lens of an achromatic doublet

```
// Scilab Code Ex4.10: Page-197
clc;clear;
w1 = 0.02;....// Magnitude of the dispersive power of first lens
w2 = 0.04;....// Magnitude of the dispersive power of second lens
// Let 1/f1 = x and 1/f2 = y, then
// The condition for achromatic combination of two
```

```
    lenses , w1/f1 + w2/f2 = 0 => w1*x + w2*y = 0
    --- (I)
F = 20;....// Given focal length of achromatic
    doublet , cm
// Also F = 1/f1 + 1/f2 => F = x + y    ---- (II)
A = [w1 w2; 1 1];       // Square matrix
B = [0;1/F];     // Column vector
X = inv(A)*B;     // Characteristic roots of the
    simultaneous equations , cm
f1 = 1/X(1);     // Focal length of convex lens , cm
f2 = 1/X(2);     // Focal length of concave lens , cm

printf("\nThe focal length of convex lens = %2d cm",
     ceil(f1));
printf("\nThe focal length of concave lens = %2d cm"
    , ceil(f2));

// Result
// The focal length of convex lens = 10 cm
// The focal length of concave lens = -20 cm
```

Scilab code Exa 4.11 Radii of curvature of the second surface of each of the lens of achromatic doublet

```
// Scilab Code Ex4.11: Page-197
clc;clear;
w1 = 0.017;....// Magnitude of the dispersive power
    of first lens
w2 = 0.034;....// Magnitude of the dispersive power
    of second lens
// Let 1/f1 = x and 1/f2 = y, then
// The condition for achromatic combination of two
    lenses , w1/f1 + w2/f2 = 0 => w1*x + w2*y = 0
    --- (I)
F = 40;....// Given focal length of achromatic
```

```
doublet, cm
// Also F = 1/f1 + 1/f2 => F = x + y    ---- (II)
A = [w1 w2; 1 1];        // Square matrix
B = [0;1/F];     // Column vector
X = inv(A)*B;     // Characteristic roots of the simultaneous equations, cm
f1 = 1/X(1);     // Focal length of convex lens, cm
f2 = 1/X(2);     // Focal length of concave lens, cm
// For the convex lens
R2 = -25;     // Radius of curvature of the contact surface, cm
mu = 1.5;     // Mean refractive index of crown glass
// From the Lens Maker formula, 1/f = (mu - 1)*(1/R1 -1/R2), solving for R1
f = f1;
R1 = 1/(1/(f*(mu-1))+1/R2);     // Radius of curvature of second surface of first lens, cm
printf("\nThe radius of curvature of second surface of first lens = %5.2f cm", R1);
// For the concave lens
R1 = -25;     // Radius of curvature of the contact surface, cm
mu = 1.7;     // Mean refractive index of flint glass
// From the Lens Maker formula, 1/f = (mu - 1)*(1/R1 -1/R2), solving for R1
f = f2;
R2 = 1/(1/R1-1/(f*(mu-1)));     // Radius of curvature of second surface of second lens, cm
printf("\nThe radius of curvature of second surface of second lens = %5.2f cm", R2);

// Result
// The radius of curvature of second surface of first lens = 16.67 cm
// The radius of curvature of second surface of second lens = -233.33 cm
```

Scilab code Exa 4.12 Focal length of the convergent lens for C line

```
// Scilab Code Ex4.12: Page-199
clc;clear;
// For flint glass
mu_C = 1.665;     // Refractive index of flint glass
    for C line
mu_F = 1.700;     // Refractive index of flint glass
    for F line
mu_D = (mu_F+mu_C)/2;     // Refractive index of
    flint glass for D line
w2 = (mu_F-mu_C)/(mu_D-1);....// Magnitude of the
    dispersive power of second lens of flint glass
// For crown glass
mu_C = 1.510;     // Refractive index of crown glass
    for C line
mu_F = 1.536;     // Refractive index of crown glass
    for F line
mu_D = (mu_F+mu_C)/2;     // Refractive index of
    flint glass for D line
w1 = (mu_F-mu_C)/(mu_D-1);....// Magnitude of the
    dispersive power of second lens of crown glass
f = 50;    // Focal length of acromatic doublet, cm
FD = f*(w2-w1)/w2;    // Focal length of D line of
    the Fraunhofer spectrum due to convex lens of
    crown glass
FC = FD*(mu_D - 1)/(mu_C - 1);    // Focal length of
     C component of converging lens, cm
printf("\nThe focal length of C component of
    converging lens = %4.2f cm", FC);

// Result
// The focal length of C component of converging
    lens = 1.57 cm
```

Scilab code Exa 4.13 Focal length of two lenses with no aberration

```
// Scilab Code Ex4.13 Page-200
clc;clear;
F = 50;....// Equivalent focal length of combination of two lenses, cm
//d = f1+f2/2, condition for no chromatic aberration ....(1)
//d = f2-f1, condition for minimum spherical aberration....(2)
// From (1) and (2), f1 = 3*d/2, f2 = d/2
// As 1/F = 1/f1 + 1/f2 - d/(f1*f2), solving for d
d = 4/3*50;    // Distance of separation betwen two lenses, cm
f1 = 3*d/2, f2 = d/2;
printf("\nf1 = %3d cm, f2 = %5.2f cm", ceil(f1), f2);

// Result
// f1 = 100 cm, f2 = 33.33 cm
```

Scilab code Exa 4.14 Longitudinal chromatic aberration for an object at infinity

```
// Scilab Code Ex4.14 : Page-200 (2011)
clc;clear;
mu_R = 1.5230;    // Refractive index for red wavelength
mu_V = 1.5145;    // Refractive index for violet wavelength
R1 = 40;    // Radius of curvature for red wavelength, cm
```

```
R2 = -10;      // Radius of curvature for violet
    wavelength, cm
// As 1/f = (mu - 1)*(1/R1 - 1/R2), solving for fV
    and fR
fV = 1/((mu_V-1)*(1/R1-1/R2));     // Focal length
    for violet wavelength, cm
fR = 1/((mu_R-1)*(1/R1-1/R2));     // Focal length
    for violet wavelength, cm
l = fR - fV;     // Longitudinal chromatic aberration
    , cm
printf("\nThe longitudinal chromatic aberration = %5
    .3f cm", abs(l));

// Result
// The longitudinal chromatic aberration = 0.253 cm
```

Scilab code Exa 4.15 Focal length of component lenses of a convergent doublet

```
// Scilab Code Ex4.15: Page-202 (2011)
clc;clear;
F = 10;....// Equivalent focal length of a
    combination of two lenses, cm
d = 2;....// Separation distance between two lenses,
     cm
// As d = f1-f2, condition for minimum spherical
    aberration => f1 = d+f2
// and F = f1*f2/(f1+f2-d), so solving for f2
f2 = 2*F-d;     // Focal length of second lens, cm
f1 = d+f2;     // Focal length of first lens, cm
printf("\nf1 = %2d cm, f2 = %2d cm", f1, f2);

// Result
// f1 = 20 cm, f2 = 18 cm
```

Scilab code Exa 4.16 Radii of Aplanatic surfaces and lateral magnification of the image

```
// Scilab Code Ex4.16: Page-202 (2011)
clc;clear;
mu = 1.6;....// Refractive index of aplanatic
    surface
R = 3.2;....// Radius of curvature, cm
R1 = R/mu;....// First radius of the aplanatic
    surface, cm
printf("\nR1 = %3.1f cm", R1);
R2 = R*mu;....// Second radius of the aplanatic
    surface, cm
printf("\nR2 = %4.2f cm", R2);
//Since the image of an object at one aplanatic
    point will be formed by the sphere at the other
    aplantic point,so the  is
m = mu^2;    // The lateral magnification of the
    image
printf("\nThe lateral magnification of the image =
    %4.2f", m);

// Result
// R1 = 2.0 cm
// R2 = 5.12 cm
// The lateral magnification of the image = 2.56
```

Scilab code Exa 4.17 Aplanatic surface

```
// Scilab Code Ex4.17: Page-203 (2011)
clc;clear;
```

```
mu = 1.52;....// Refractive index of aplanatic surface
R = 30;....// Radius of curvature, cm
R1 = R/mu;....// First radius of the aplanatic surface, cm
printf("\nR1 = %5.2f cm", R1);
R2 = R*mu;....// Second radius of the aplanatic surface, cm
printf("\nR2 = %4.1f cm", R2);
//Since the image of an object at one aplanatic point will be formed by the sphere at the other aplantic point,so the  is
m = mu^2;    // The lateral magnification of the image
printf("\nThe lateral magnification of the image = %4.2f", m);

// Result
// R1 = 19.74 cm
// R2 = 45.6 cm
// The lateral magnification of the image = 2.31
```

Scilab code Exa 4.18 Focal length of the field lens

```
// Scilab Code Ex4.18: Page-203  (2011)
clc;clear;
F = 5;....// Equivalent focal length of Huygens eyepiece, cm
// as f1 = 3*f, f2 = f and d = 2*f, therefore
f = 2/3*F;    // Focal length of base lens, cm
f1 = 3*f;    // Focal length of field lens, cm
printf("\nThe focal length of the field lens = %2d cm", f1);

// Result
```

```
// The focal length of the field lens = 10 cm
```

Scilab code Exa 4.19 Equivalent focal length of a Ramsden eyepiece

```
// Scilab Code Ex4.19: Page-204 (2011)
clc;clear;
f = 10;....// Given focal length of each lens, cm
f1 = f;    // Focal length of first lens, cm
f2 = f;    // Focal length of second lens, cm
d = 2/3*f;  // Separation distance between two
    lenses, cm
F = f1*f2/(f1+f2-d);     // Equivalent focal length
    of Ramsden eyepiece, cm
printf("\nThe equivalent focal length of the field
    lenses is = %3.1f cm", F);

// Result
// The equivalent focal length of the field lenses
    is = 7.5 cm
```

Scilab code Exa 4.20 Focal lengths of the lenses and the eyepiece

```
// Scilab Code Ex4.20 : Page-204 (2011)
clc;clear;
d = 10;....// Distance between the two thin plano
    convex lenses in the Huygens eyepiece,
f = d/2;    // Base focal length
f1 = 3*f;   // Focal length of the first component
    lens, cm
printf("\nf1 = %d cm", f1);
f2 = f;     // Focal length of the second component
    lens, cm
printf("\nf2 = %d cm", f2);
```

```
F = 3/2*f;  // Equivalent focal length of the lens,
    cm
printf("\nF = %3.1f cm", F);

// Result
// f1 = 15 cm
// f2 = 5 cm
// F = 7.5 cm
```

Scilab code Exa 4.21 Focal length of the component lenses and the sepration between them

```
// Scilab Code Ex4.21: Page-204 (2011)
clc;clear;
F = 4.2;....// Equivalent focal length of Ramsden
    eyepiece, cm
//F = 3/4*f, Equivalent focal length of Ramsden
    eyepiece,
f = 5.6;....//focal length,in cm
f1 = f;
f2 = f;
printf("\nf1 = %3.1f cm", f1);
printf("\nf2 = %3.1f cm", f2);
d = 2/3*f;
printf("\nd = %4.2f cm", d);

// Result
// f1 = 5.6 cm
// f2 = 5.6 cm
// d = 3.73 cm
```

Chapter 5

Physical Optics

Scilab code Exa 5.1 Order of interference maximum with different wavelength

```
// Scilab Code Ex5.1: Page:297 (2011)
clc;clear;
n1 = 10;....// Order of interference maximum for
   lambda = 7000 angstrom
lambda1 = 7000;....// Wavelength of the light,
   angstrom
lambda2 = 5000;....// Wavelength of the light,
   angstrom
// As W = D*lambda/(2*d) then, x = n1*D*lambda1/(2*d
   ) = n2*D*lambda2/(2*d), solving for n2
n2 = n1*lambda1/lambda2;    // Order of interference
    maximum for lambda = 5000 angstrom
printf("\nThe order of interference maximum for
   wavelength of 5000 angstrom = %2d ", n2);

// Result
// The order of interference maximum for wavelength
   of 5000 angstrom = 14
```

Scilab code Exa 5.2 Angle of the biprism

```
// Scilab Code Ex5.2: Page:297 (2011)
clc;clear;
D = 1.6;....// Distance between the slit and the
    screen , m
a = 0.4;....// Distance between the slit and the
    biprism , m
mu = 1.52;....// Refractive index of the material of
    biprism
W = 1e-004;     // Fringe width , m
lambda = 5.893e-007;....// Wavelength of light used ,
    m
// As W = lambda*D/(2*a(mu-1)*alpha then
alpha = ((lambda*D)/(2*a*(mu-1)*W))*180/%pi;    //
    Angle of biprism , degrees
printf("\nThe angle of the biprism = %3.1f degrees",
    alpha);

// Result
// The angle of the biprism = 1.3 degrees
```

Scilab code Exa 5.3 Thickness of the mica sheet

```
// Scilab Code Ex5.3 : Page:298 (2011)
clc;clear;
lambda = 5.890e-7;....// Wavelength of source of
    light , m
mu = 1.6;....//refractive index of the mica sheet
// As del_x = W*(mu-1)*t/lambda, where del_x = 3*W,
    solving for t
```

```
t = 3*lambda/(mu-1);     // Thickness of the mica
   sheet, m
printf("\nThe thickness of the mica sheet = %5.3e cm
   ", t/1e-02);

// Result
// The thickness of the mica plate is = 2.945e-004
   cm
```

Scilab code Exa 5.4 Distance between the two coherent sources

```
//Scilab Code Ex5.4: Page:298 (2011)
clc;clear;
lambda = 6.0e-7;....// Wavelength of the
   monochromatic light, m
D = 1;....// Distance between the screen and the two
    coherent sources, m
W = 5e-004;....// Fringe width, m
d = lambda*D/(W*1e-03);     // Distance between two
   coherent sources, mm
printf("\nThe distance between the two coherent
   sources = %3.1f mm", d);

// Result
// The distance between the two coherent sources =
   1.2 mm
```

Scilab code Exa 5.5 Wavelength of light used in a biprism experiment

```
// Scilab Code Ex5.5: Page:298 (2011)
clc;clear;
D = 1;....// Distance between slits and the screen,
   m
```

```
mu = 1.5;     // Refractive index of the material of
    biprism
a = 0.5;....// The distance between the slit and the
     biprism , m
W = 1.35e-004;....// Width of the fringes , m
alpha = (180-179)/2*%pi/180;     // Acute angle of
    biprism , radian
lambda = 2*a*(mu-1)*alpha*W/D;     // Wavelength of
    light used , m
printf("\nThe wavelength of light used = %4d
    angstrom", lambda/1e-10);

// Result
// The wavelength of light used = 5890 angstrom
```

Scilab code Exa 5.6 Distance between the slits

```
// Scilab Code Ex5.6: Page:299 (2011)
clc;clear;
lambda = 6.328e-007;....// Wavelength of the
    monochromatic light , m
D = 40;....// Distance between the slits and the
    screen , m
W = 0.1;....// Distance between the interference
    maxima , m
d = lambda*D/W;     // Distance between the slits , m
printf("\nThe distance between the slits = %6.4f mm"
    ,d/1e-03);

// Result
// The distance between the slits = 0.2531 mm
```

Scilab code Exa 5.7 Lateral shift of central maximum

```
// Scilab Code Ex5.7: Page:299 (2011)
clc;clear;
lambda = 5.0e-007;....// Wavelength of the
    monochromatic light, m
D = 1;....// Distance between the silts and the
    screen, m
d = 5e-004/2;....// Half of the distance between the
     two slits, m
mu = 1.5;....// Refractive index of glass
t = 1.5e-006;....// Thickness of thin glass plate, m
del_x = D*(mu-1)*t/(2*d);
printf("\nThe lateral shift of central maximum = %3
    .1f m", del_x/1e-03);

// Result
// The lateral shift of central maximum = 1.5 m
```

Scilab code Exa 5.8 Thickness of a soap bubble film

```
// Scilab Code Ex5.8: Page:300 (2011)
clc;clear;
lambda = 6.0e-007;....// Wavelength of the light, m
mu = 1.463;....// Refrctive index of a soap bubble
    film
n = 0;     // Value of n for smallest thickness
r = 0;     // Angle of refraction for normal
    incidence
// As 2*mu*t*cos(r) = (2*n+1)*lambda/2, solving for
    t
t = (2*n+1)*lambda/(4*mu*cos(r));     // The
    thickness of a soap bubble film, m
printf("\nThe thickness of a soap bubble film = %5.1
    f angstrom", t/1e-010);

// Result
```

```
// The thickness of a soap bubble film = 1025.3 angstrom
```

Scilab code Exa 5.9 Wavelength of the light used in Newton rings experiment

```
// Scilab Code Ex5.9: Page:300 (2011)
clc;clear;
D5 = 3.36e-003;....// Diameter of Newton's 5th ring, m
D15 = 5.90e-003;....// Diameter of Newton's 15th ring, m
m = 10;     // Number of ring
R = 1;....// Radius of the plano-convex lens, m
lambda = (D15^2-D5^2)/(4*m*R);
printf("\nThe wavelength of the light used = %4d angstrom", lambda/1e-010);

// Result
// The wavelength of the light used = 5880 angstrom
```

Scilab code Exa 5.10 Radius of the curvature of the lens and the thickness of the air film

```
// Scilab Code Ex5.10: Page:301 (2011)
clc;clear;
D10 = 0.005;....// Diameter of Newton's 5th ring, m
n = 10;....// Order of the ring
lambda = 6.0e-007;....// Wavelength of the light used, m
R = (D10^2)/(4*n*lambda);     // Radius of the curvature of the lens, m
```

```
printf("\nThe radius of the curvature of the lens =
    %6.4f m", R);
t = D10^2/(8*R);
printf("\nThe thickness of the corresponding air
    film = %3.1e m",t);

// Result
// The radius of the curvature of the lens = 1.0417
    m
// The thickness of the corresponding air film = 3.0
    e-006 m m
```

Scilab code Exa 5.11 Thickness of the soap film

```
// Scilab Code Ex5.11: Page-301 (2011)
clc;clear;
mu = 1.43;....// Refractive index of the soap film
n = 0;     // Order of fringes for smallest thickness
i = 30;     // Angle of incidence, degrees
// As sin(i)/sin(r) = mu, cos(r)
cosr = sqrt(1-(sind(i)/mu)^2);     // Cosine of angle
    r
lambda = 6.0e-007;....// Wavelength of the light, m
t = (2*n+1)*lambda/(4*mu*cosr);....// Thickness of
    the soap film, m
printf("\nThe thickness of the soap film = %4.2e m",
    t);

// Result
// The thickness of the soap film = 1.12e-007 m
```

Scilab code Exa 5.12 Least thickness of the soap film that will appear bright dark

```
// Scilab Code Ex5.12: Page:301 (2011)
clc;clear;
lambda = 5.893e-007;....// Wavelength of the sodium light , m
mu = 1.42;....// Refractive index of the soap film
r = 0;     // Angle of refraction , degrees
n = 0;     // Order of diffraction for least thickness of dark film
t = (2*n+1)*lambda/(4*mu*cosd(r));     // Least thickness of the film that will apear bright , m
printf("\nThe least thickness of the film that will appear bright = %5.1f m", t/1e-010);
n = 1;     // Order of diffraction for least thickness of bright film
t = n*lambda/(2*mu*cosd(r));     // Least thickness of the film that will apear dark , m
printf("\nThe least thickness of the film that will appear dark = %6.2f m",t/1e-010);

// Result
// The least thickness of the film that will appear bright = 1037.5 m
// The least thickness of the film that will appear dark = 2075.00 m
```

Scilab code Exa 5.13 Thickness of the wire separating edges of two plane glass surfaces

```
// Scilab Code Ex5.13: Page:302 (2011)
clc;clear;
lambda = 5.893e-007;....// Wavelength of the sodium light , m
// As fringe width of the thin wedge-shaped air film is
// W = lambda/(2*t/20*W), solving for t
```

```
t = (10*lambda);     // Thickness of the wire separating edges of two plane glass surfaces, m
printf("\nThe thickness of the wire = %5.3e m", t);

// Result
// The thickness of the wire = 5.893e-006 m
```

Scilab code Exa 5.14 Radius of curvature of the lens and the thickness of the corresponding air film

```
// Scilab Code Ex5.14: Page:303 (2011)
clc;clear;
lambda = 5.9e-007;....// Wavelength of the reflected light, m
n = 10;....// Order of the ring
D10 = 0.005;....// Diameter of the 10th ring,in m
R = (D10^2)/(4*n*lambda);     // Radius of curvature of the lens, m
printf("\nThe radius of curvature of the lens = %6.4f m", R);
t = (D10^2)/(8*R);     // Thickness of the corresponding air film, m
printf("\nThe thickness of the corresponding air film = %4.2e m",t);

// Result
// The radius of curvature of the lens = 1.0593 m
// The thickness of the corresponding air film = 2.95e-006 m
```

Scilab code Exa 5.16 Angles at which first and second order maxima can be observed

```
// Scilab Code Ex5.16: Page:304 (2011)
clc;clear;
lambda = 6.328e-007;....// Wavelength of
    monochromatic light from He laser, m
n1 = 1;....// First order
n2 = 2;....// Second order
l = 6000;....// Lines/cm of the diffraction grating
A= 1.66e-6;
theta = asind(n1*lambda/A);
printf("\n The first order maximum angle = %4.1f
    degrees",theta);
theta = asind(n2*lambda/A);
printf("\n The second order maximum angle = %4.1f
    degrees",theta);

// Result
// The first order maximum angle = 22.4 degrees
// The second order maximum angle = 49.7 degrees
```

Scilab code Exa 5.17 Relation between two wavelengths illuminating a single slit due to Fraunhofer diffraction

```
// Scilab Code Ex5.17: Page:305 (2011)
clc;clear;
a = 1;     // For simplicity assume slit width to be
    unity, unit
theta = 1; // For simplicity assume diffraction
    angle to be unity, unit
// As a*sin(theta) = m*lambda, solving for lambdas
lambda1 = a*sin(theta);     // First wavelength,
    angstrom
lambda2 = a*sin(theta)/2;     // First wavelength,
    angstrom
printf("\nlambda1 = %d*lambda2", lambda1/lambda2);

```

```
// Result
// lambda1 = 2*lambda2
```

Scilab code Exa 5.18 Angular position of the first two minima on either side of a central maxima

```
// Scilab Code Ex5.18: Page:305 (2011)
clc;clear;
function [deg, minute] = deg2degmin(theta)
    deg = int(theta);
    minute = (theta - deg)*60;
endfunction

lambda = 5.5e-007;....// Wavelength of light , m
a = 2.2e-006;....// Width of the slit , m
l = 6000;....// Lines /cm of the diffraction grating
// In a single slit diffraction pattern the
   directions of minimum intensity are given by a*
   sintheta = m*lambda where m = 1 ,2 ,3....
// For m = 1
m = 1;....// First order
theta = asind(m*lambda/a);    // Angular position of
    first minima on either side of the central
   maxima, degrees
[deg, minute] = deg2degmin(theta);    // Degree to
   deg-min conversion
printf("\nThe angular position of first minima on
   either side of the central maxima = %2d degrees
   %2d minutes", deg, minute);
// For m = 2
m = 2;....// Second order
theta = asind(m*lambda/a);
[deg, minute] = deg2degmin(theta);    // Degree to
   deg-min conversion
printf("\nThe angular position of second minima on
```

```
    either side of the central maxima = %2d degrees
    %2d minutes", deg, minute);

// Result
//
```

Scilab code Exa 5.19 Wavelength of light and the missing order of Fraunhofer diffraction

```
// Scilab Code Ex5.19: Page:306 (2011)
clc;clear;
D = 1.7;....// Distance between the slit and the
    screen, m
W = 2.5e-003;....// Given fringe width, m
a = 8e-005;....// Width of the first slit, m
b = 4e-004;....// Width of the second slit, m
n = b;    //
p = [1 2 3 4 5 6];
// In a double slit experiment Fraunhoffer
    diffraction pattern,the fringe width is given by
    W = lambda*D/n
lambda = b*W/D;     // Wavelength of the light used,
    m
printf("\nThe wavelength of light = %4d angstrom",
    lambda/1e-010);
printf("\nThe missing orders are:\n");
for i = 1:1:6
s = [(a+b)/a]*p(i);
printf("\t%d", s);
end
printf("  etc.")
// Result
// The wavelength of light = 5882 angstrom
// The missing orders are:
//  6   12  18  24  30  36   etc.
```

Scilab code Exa 5.20 Deduction of wavelength of the light from given data

```
//Scilab Code Ex5.20: Page-306 (2011)
clc;clear;
D = 2;....// Distance of the screen from the slit, m
x = 1.6e-02;....// Position of centre of the second dark band, m
m = 2;     // Order of diffraction
a = 1.4e-04;....// Width of the slit, m
lambda = a*x/(m*D);     // Wavelength of light, m
printf("\n The wavelength of the light = %4d angstrom", ceil(lambda/1e-010));

// Result
// The wavelength of the light = 5600 angstrom
```

Scilab code Exa 5.21 Minimum number of lines in a grating

```
// Scilab Code Ex5.21: Page:307 (2011)
clc;clear;
lambda1 = 5890;....// Wavelength of the line, angstrom
lambda2 = 5896;....// Wavelength of the line, angstrom
d_lambda = lambda2 - lambda1;....// Wavelength difference, angstrom
n = 2;....// Order of diffraction
N = lambda2/(n*d_lambda);     // Minimum no. of lines in a grating
printf("\nThe minimum number of lines in the grating = %3d lines", N);
```

```
// Result
// The minimum number of lines in the grating = 491
   lines
```

Scilab code Exa 5.22 Maximum number of visible orders

```
// Scilab Code 5.22: Page:307 (2011)
clc;clear;
lambda = 5.0e-07;....// Wavelength of the radiation ,
    m
a_plus_b = 2.54e-02/2620;....// The grating element ,
    m
theta_max = 90;    // Maximum value of angle of
   diffraction , degrees
n_max = a_plus_b/lambda*sind(theta_max);    //
   Maximum number of visible orders
printf("\nThe number of visible orders = %2d ",
   n_max);

// Result
// The number of visible orders = 19
```

Scilab code Exa 5.23 Grating element of diffraction grating

```
// Scilab Code 5.23: Page:307 (2011)
clc;clear;
lambda1 = 6000;....// Wavelength of yellow line ,
   angstrom
lambda2 = 4800;....// Wavelength of blue line ,
   angstrom
theta = asin(3/4);    // Angle of diffraction ,
   radian
```

```
// As a_plus_b*sin_theta) = n*lambda, so n*lambda1 = (n+1)*lambda2, solving for n
n = poly(0, 'n');
n = roots(n*6000 - (n+1)*4800);     // Order of diffraction
a_plus_b = n*6000/sin(theta);     // Grating element of diffraction grating, m
printf("\nThe Grating element of diffraction grating = %3.1e m", a_plus_b*1e-010);

// Result
// The Grating element of diffraction grating = 3.2e-006 m
```

Scilab code Exa 5.26 Coinciding spectral lines

```
// Scilab Code Ex5.26: Page:310 (2011)
clc;clear;
n = 5;....// Order for given wavelength
m = [4 5 6 7 8];     // Orders of spectral lines in the visible range
lambda1 = 6000;....// Wavelength of the spectral line in visible range, angstrom
lambda2 = zeros(5);
printf("\n The spectral lines in visible ranges are :\n");
for i=1:1:5
l2 = (n*lambda1)/m(i);
lambda2(i) = l2;     // Preserve the lambda value
printf("%4d angstrom\n", ceil(l2));
end
printf("\nThe other spectral lines in the visible range 4000A to 7000A are");
for i=1:1:5
    if lambda2(i) < 7000 & lambda2(i) > 4000 then
```

```
        if lambda2(i) == 6000 then
            continue
        end
          printf("\n%4dA", ceil(lambda2(i)));
    end
end

// Result
// The spectral lines in visible ranges are:
// 7500 angstrom
// 6000 angstrom
// 5000 angstrom
// 4285 angstrom
// 3750 angstrom

// The other spectral lines in the visible range
   4000A to 7000A are
// 5000A
// 4286A
```

Scilab code Exa 5.27 Resolution of D1 and D2 lines of sodium

```
// Scilab Code Ex5.27: Page:310 (2011)
clc;clear;
N = 4500;....// Number of lines in grating
n = 2;....// Order of diffraction
lambda1 = 5890;....// Wavelength, angstrom
lambda2 = 5896;....// Wavelength, angstrom
RP2 = n*N;    // Resolving power of grating in the
   second order
lambda = (lambda1+lambda2)/2;    // Mean wavelength
   of sodium light, angstrom
d_lambda = lambda2 - lambda1;    // Wavelength
   difference, angstrom
RP = lambda/d_lambda;    // Calculated resolving
```

```
    power of grating
if RP2 <> RP then
    printf("\nThe D1 and D2 lines of Na light cannot be resolved in second order");
end

// Result
// The D1 and D2 lines of Na light cannot be resolved in second order
```

Scilab code Exa 5.28 Distance between two stars which are just resolved

```
// Scilab Code Ex5.28: Page:311 (2011)
clc;clear;
lambda = 5.5e-07;....// Wavelength of light used, m
f = 3.0;....// Focal length of telescope objective, m
a = 0.01;....// Diameter of the telescope objective, m
// As x/f = 1.22*lambda/a, the Rayleigh criterian for resolution, solving for x
x = 1.22*f*lambda/a;     // Distance between two stars just seen as separate, m
printf("\nThe distance between two stars just seen as separate = %3.1e m ", x);

// Result
// The distance between two stars just seen as separate = 2.0e-004 m
```

Scilab code Exa 5.29 Numerical aperature of the objective of microscope

```
// Scilab Code Ex5.29: Page:311 (2011)
```

```
clc;clear;
lambda = 5.461e-07;....// Wavelength of light used, m
d = 4.0e-07;....// Distance between the two luminous objects, m
// As d = 1.22*lambda/(2*mu*sin(alpha)) = 1.22*lambda/(2*NA), solving for NA
NA = 1.22*lambda/(2*d);      // Numerical aperature of the objective of microscope
printf("\nThe numerical aperature of the objective of microscope = %5.3f ", NA);

// Result
// The numerical aperature of the objective of microscope = 0.833
```

Scilab code Exa 5.30 Aperture of the objective of a telescope

```
// Scilab Code Ex5.30: Page:312 (2011)
clc;clear;
lambda = 6.0e-07;....// Wavelength of light used, m
d_theta = 2.44e-06;....// Angular separation between the two stars, radian
a = 1.22*lambda/d_theta;     // Aperature of the objective of a telescope from Rayleigh criterian, m
printf("\nThe aperature of the objective of the telescope = %3.1f m ", a)

// Result
// The aperature of the objective of the telescope = 0.3 m
```

Scilab code Exa 5.31 Minimum distance from the telescope at which the the pinhole can be resolved

```
// Scilab Code Ex5.31:Page:312 (2011)
clc;clear;
lambda = 5.5e-007;....// Wavelength of light used, m
x = 1.5e-003;....// Distance between the two
    pinholes, m
a = 4.0e-003;....// Diameter of objective, m
D = a*x/(1.22*lambda);     // Minimum distance from
    the telescope at which the the pinhole can be
    resolved from Rayleigh criterian, m
printf("\nThe minimum distance from the telescope at
     which the the pinhole can be resolved = %4.2f m
    ", D);

// Result
// The minimum distance from the telescope at which
    the the pinhole can be resolved = 8.9
```

Scilab code Exa 5.32 Numerical aperature of the objective of microscope for given wavelength of light

```
// Scilab Code Ex5.32: Page:312 (2011)
clc;clear;
lambda = 5.461e-07;....// Wavelength of light used,
    m
d = 5.55e-07;....// Distance between the two
    luminous objects, m
// As d = 1.22*lambda/(2*mu*sin(alpha)) = 1.22*
    lambda/(2*NA), solving for NA
NA = 1.22*lambda/(2*d);      // Numerical aperature
    of the objective of microscope
printf("\nThe numerical aperature of the objective
    of microscope = %4.2f ", NA);
```

```

// Result
// The numerical aperature of the objective of
   microscope = 0.60
```

Scilab code Exa 5.33 Angle of minimum deviation for green light for its passage through a prism

```
// Scilab Code Ex5.33: Page:313 (2011)
clc;clear;
i = 60;    // Angle of incidence, degrees
mu = tand(i);    // Brewester's Law to calculate
   refractive index
A = 60;....// Angle of prism, degrees
// As mu = sind((A+delta_m)/2)/sind(A/2), solving
   for delta_m
delta_m = 2*asind(mu*sind(A/2))-A;    // Angle of
   minimum deviation for green light for its passage
    through a prism, degrees
printf("\nThe angle of minimum deviation for green
   light for its passage through a prism = %2d
   degrees", ceil(delta_m));

// Result
// The angle of minimum deviation for green light
   for its passage through a prism = 60 de
```

Scilab code Exa 5.34 Thicknass of quarter wave plate

```
// Scilab Code Ex5.34: Page:313 (2011)
clc;clear;
lambda = 5.89e-07;....// Wavelength of light used, m
```

```
mu_O = 1.55;     // Refractive index of ordinary light
mu_E = 1.54;     // Refractive index of extraordinary light
tQ = lambda/(4*(mu_O-mu_E));     // The thickness of the quarter wave plate, m
printf("\nThe thickness of the quarter plate is = %6.4e m", tQ);

// Result
// The thickness of the quarter plate is = 1.4725e-005 m
```

Scilab code Exa 5.35 Percentage purity of the sugar sample

```
// Scilab Code Ex5.35: Page:313 (2011)
clc;clear;
theta = 9.9;....// Optical rotation of solution, degrees
l = 20;....// Length of the tube, cm
S = 66;....// Specific rotation of pure sugar solution, degree per dm-(g/cc)
// As the specific rotation, S = 10*theta/l*c, solving for c
c = 10*theta/(l*S);     // Concentration of solution for pure sugar, g/cc
c_prime = 0.080;     // Concentration of solution for impure sugar, g/cc
Percentage_purity = c*100/c_prime;     // Percentage purity of sugar sample
printf("\nThe percentage_purity of the sugar sample = %5.2f percent", Percentage_purity);

// Result
// The percentage_purity of the sugar sample = 93.75
```

```
percent
```

Scilab code Exa 5.36 Specific rotation of sugar solution for given plane of polarization

```
// Scilab Code Ex5.36: Page:314 (2011)
clc;clear;
theta = 26.4;....// Optical rotation of sugar solution, degrees
l = 20;....// Length of the tube, cm
c = 0.20;....// Concentration of the solution, g/cc
S = 10*theta/(l*c);     // The specific rotation of the sugar solution, degree per dm per (g/cc)
printf("\nThe specific rotation of the sugar solution = %2d degrees",S);

// Result
// The specific rotation of the sugar solution = 66 degrees
```

Scilab code Exa 5.37 Angle of rotation produced by quartz plate

```
// Scilab Code Ex5.37: Page:315 (2011)
clc;clear;
// Function to convert degrees to deg-min
function [d,m] = deg2degmin(deg)
    d = int(deg);
    m = (deg-d)*60;
endfunction

lambda = 7.62e-07;....// Wavelength of the polarized light, m
```

```
mu_R = 1.53914;    // Refractive index of quartz for right-handed circularly polarized light
mu_L = 1.53920;    // Refractive index of quartz for left-handed circularly polarized light
t = 5.0e-004;....// Thickness of the plate, m
theta = %pi*t*(mu_L-mu_R)/lambda;    // The angle of optical rotation, radian
[d,m] = deg2degmin(theta*180/%pi);    // Call the conversion function
printf("\nThe angle of rotation produced by its plate = %6.4f radians = %d degrees %d minutes", theta, d, m);

// Result
// The angle of rotation produced by its plate = 0.1237 radians = 7 degrees 5 minutes
```

Scilab code Exa 5.38 Optical rotation produced by new length of sugar solution

```
// Scilab Code Ex5.38: Page:315 (2011)
clc;clear;
theta = 13;....// Optical rotation of the solution, degrees
l = 20;....// Length of the tube, cm
l_prime = 30;....// New length of the tube, cm
c = 1;    // For simplicity assume concentration of sugar solution to be unity, g/cc
c_prime = c/3;    // New concentration of sugar solution, g/cc
// As, S = 10*theta/(l*c) so 10*theta/(l*c) = 10*theta_prime/(l_prime*c_prime)
// Solving for theta_prime
theta_prime = theta/(l*c)*l_prime*c_prime;    // The optical rotation produced by new length of sugar
```

```
    solution, degrees
printf("\nThe optical rotation of %d cm length of
    sugar solution = %3.1f degrees", l_prime,
    theta_prime);

// Result
// The optical rotation of 30 cm length of sugar
    solution = 6.5 degrees
```

Scilab code Exa 5.39 Strength of the solution

```
// Scilab Code Ex5.39: Page:315 (2011)
clc;clear;
theta = 11;....// Optical rotation of sugar solution
    , degrees
l = 20;....// Length of the tube, cm
S = 66;....// Specific rotation of sugar solution,
    degrees
c = theta*10/(l*S);    // The concentration of sugar
     solution, g/cc
printf("\nThe strength of the solution = %6.4f g/cc"
    , c);

// Result
// The strength of the solution = 0.0833 g/cc
```

Scilab code Exa 5.40 Length of sugar solution for given concentration and optical rotation

```
// Scilab Code Ex5.40: Page:316 (2011)
clc;clear;
theta = 20;....// Optical rotation of sugar solution
    , degrees
```

```
theta_prime = 35;....// New optical rotation of
    sugar solution , degrees
c = 5;....// Percentage concentration of the
    solution
c_prime = 10;  // New percentage concentration of
    the solution
l = 1;        // For simplicity assume length of the
    sugar solution to be unity
l_prime = theta_prime*l*c/(c_prime*theta);
printf("\nThe length of sugar solution for %d
    percent concentration and %d degrees optical
    rotation = %5.3f*l ", c_prime, theta_prime,
    l_prime);

// Result
// The length of sugar solution for 10 percent
    concentration and 35 degrees optical rotation =
    0.875*l
```

Chapter 6

X Rays

Scilab code Exa 6.1 Electrons striking the target in X ray coolidge tube

```
// Scilab Code Ex6.1: Page-369 (2011)
clc;clear;
i = 2e-003;....// Current through X-ray tube, A
e = 1.6e-019;....// Charge on an electron, C
V = 12.4e+003;....// Potential difference applied
    across X-ray tube, V
m0 = 9.1e-031;....// Rest mass of the electron, Kg
n = i/e;    // Number of electrons striking the
    target per second
printf("\nThe number of electrons striking the
    target per sec = %4.2e electrons", n);
v = sqrt(2*e*V/m0);....// Velocity of the electrons,
     m/s
printf("\nThe speed with which electrons strike the
    target = %4.2e m/s", v);

// Result
// The number of electrons striking the target per
    sec = 1.25e+016 electrons
// The speed with which electrons strike the target
   = 6.60e+007 m/s
```

Scilab code Exa 6.2 Maximum speed of the electron striking the target

```
// Scilab Code Ex6.2: Page-370 (2011)
clc;clear;
e = 1.6e-019;....// Charge on an electron, C
V = 13.6e+003;....// Potential difference applied
    across X-ray tube, V
m0 = 9.1e-031;....// Rest mass of the electron, Kg
v = sqrt(2*e*V/m0);....// Velocity of the electron,
    m/s
printf("\nThe maximum speed with which the electrons
     strike the target = %4.2e m/s", v);

// Result
// The maximum speed with which the electrons strike
     the target = 6.92e+007 m/s
```

Scilab code Exa 6.3 Longest wavelength that can be analysed by a rock salt crystal

```
// Scilab Code Ex6.3: Page-370 (2011)
clc;clear;
d = 2.82e-010;....// Spacing of the rock-salt, m
n = 2;....// Order of diffraction
theta = %pi/2;     // Angle of diffraction, radian
// Braggs equation for X-rays of wavelength lambda
    is n*lambda = 2*d*sin(theta), solving for lambda
lambda = 2*d*sin(theta)/n;     // Wavelength of X-ray
     using Bragg's law, m
printf("\nThe longest wavelength that can be
    analysed by a rock-salt crystal = %4.2f angstrom"
    , lambda/1e-010);
```

```
// Result
// The longest wavelength that can be analysed by a
   rock-salt crystal = 2.82 angstrom
```

Scilab code Exa 6.4 Angles at which the second and the third Bragg diffraction maxima are observed

```
// Scilab Code Ex6.4: Page-371 (2011)
clc;clear;
lambda = 3e-011;....// Wavelength of the X-ray, m
d = 5e-011;....// Lattice spacing, m
n = [2 3];....// Orders of diffraction
// Bragg's equation for X-rays of wavelength lambda
   is n*lambda = 2*d*sin(theta), solving for thetas
for i = 1:1:2
theta = asind(n(i)*lambda/(2*d));
printf("\nFor n = %d, theta = %4.1f degrees", n(i),
   theta);
end

// Result
// For n = 2, theta = 36.9 degrees
// For n = 3, theta = 64.2 degrees
```

Scilab code Exa 6.5 Interplanar sepration of atomic planes in the crystal

```
// Scilab Code Ex6.5: Page-371 (2011)
clc;clear;
lambda = 3.6e-011;....// Wavelength of X-rays, m
n = 1;     // Order of diffraction
theta = 4.8;     // Angle of diffraction, degrees
```

```
// Braggs equation for X-rays is n*lambda = 2*d*sin(theta), solving for d
d = n*lambda/(2*sind(theta));    // Interplanar spacing, m
printf("\nThe interplanar separation of atomic planes in the crystal = %4.2f angstrom", d/1e-010);

// Result
// The interplanar separation of atomic planes in the crystal = 2.15 angstrom
```

Scilab code Exa 6.6 Wavelength of K alpha radiation in copper for its given value in Mo

```
// Scilab Code Ex6.6: Page-371 (2011)
clc;clear;
lambda1 = 0.71;....// Wavelength of k alpha line in molybdenum, angstrom
Z1 = 42;          // Atomic number of Mo
Z2 = 29;          // Atomic number of Cu
// Wavelength of characteristic X-ray for K-alpha spectral line is given by
// 1/lambda = 3/4*R*(Z-1)^2 then
lambda2 = lambda1*(Z1-1)^2/(Z2-1)^2;    // The wavelength of K alpha radiation in copper, m
printf("\nThe wavelength of K-alpha radiation in copper = %4.2f angstrom", lambda2);

// Result
// The wavelength of K-alpha radiation in copper = 1.52 angstrom
```

Scilab code Exa 6.7 Wavelength of gamma radiation at 90 degree

```
// Scilab Code Ex6.7: Page-372 (2011)
clc;clear;
phi = %pi/2;     // Scattering angle, degrees
m0 = 9.1e-031;....// Rest mass of an electron, kg
h =  6.62e-034;....// Planck's constant, J-s
c = 3e+008;....// Speed of light in vacuum, m/s
E = 8.16e-014;....// Energy of gamma radiation, J
lambda = h*c/(E*1e-010);     // Wavelength of incident photon, angstrom
lambda_prime = lambda+h*(1-cos(phi))/(m0*c*1e-010);     // Wavelength of scattered photon, angstrom
printf("\nThe wavelength of radiation at 90 degrees = %6.4f angstrom", lambda_prime);

// Result
// The wavelength of radiation at 90 degrees = 0.0486 angstrom
```

Scilab code Exa 6.8 Compton shift from a carbon block

```
// Scilab Code Ex6.8: Page-372 (2011)
clc;clear;
phi = %pi/2;     // Scattering angle, radian
m0 = 9.1e-031;....// Rest mass of the electron, kg
h =  6.62e-034;....// Planck's constant, J-s
c = 3e+008;....// Speed of light in vacuum, m/s
lambda = 1.00 ;....// Wavelength of incident photon, in angstrom
del_lambda = h*(1-cos(phi))/(m0*c*1e-010);     // Compton shift, angstrom
printf("\nThe Compton shift = %6.4f angstrom", del_lambda);

```

```
// Result
// The Compton shift = 0.0242 angstrom
```

Scilab code Exa 6.9 Wavelength of incident photon

```
// Scilab Code Ex6.9: Page-373 (2011)
clc;clear;
phi = %pi/2;        // Scattering angle, radian
m0 = 9.1e-031;....// Rest mass of the electron, kg
h =  6.62e-034;....// Planck's constant, J-s
c = 3e+008;....// Speed of light in vacuum, m/s
// As Compton shift = del_lambda = lambda, so
lambda = h*(1-cos(phi))/(m0*c*1e-010);    // Wavelength of incident photon, angstrom
printf("\nThe wavelength of incident radiation = %6.4f angstrom", lambda);

// Result
// The wavelength of incident radiation = 0.0242 angstrom
```

Chapter 7

Lasers and Holography

Scilab code Exa 7.1 Energy of the laser pulse

```
// Scilab Code Ex 7.1: Page-411 (2011)
clc;clear;
e = 1.6e-019;....// Charge on an electron, eV
h = 6.62e-034;....// Planck's constant, J-s
c = 3e+008;....// Speed of light in vacuum, m/s
n = 2.8e+019;....// Number of photons in laser pulse
lambda = 7e-007;....// Wavelength of the radiation
    emited by the laser, m
E = h*c/(lambda*e);....// Energy of the photon in
    the laser light, eV
del_E = E*n;....// The energy of laser pulse having
    n photons, eV
printf("\nThe energy of the laser pulse = %4.2e eV",
    del_E);

// Result
// The energy of the laser pulse = 4.97e+019 eV
```

Scilab code Exa 7.2 Coherence length resultant bandwidth and line width of laser beam

```
// Scilab Code Ex7.2: Page-411 (2011)
clc;clear;
c = 3e+008;....// Speed of light in vacuum, m/s
lambda = 6.5e-007;....// Wavelength of the pulse, m
t = 0.5e-009;....// Time interval between successive
    pulses, s
L_c = c*t;....// Coherence length of laser pulse, m
printf("\nThe coherence length of the pulse = %4.2f
   m", L_c);
del_nu = 1/t;....// Resultant bandwidth of laser
   pulse, Hz
printf("\nThe bandwidth of the laser pulse = %1.0e
   Hz", del_nu);
del_lambda = lambda^2*del_nu/c;....// Linewidth of
   laser beam, m
printf("\nThe linewidth of the pulse = %5.3f
   angstrom", del_lambda/1e-010);

// Result
// The coherence length of the pulse = 0.15 m
// The bandwidth of the laser pulse = 2e+009 Hz
// The linewidth of the pulse = 0.028 angstrom
```

Scilab code Exa 7.3 Angular spread and areal spread of laser beam

```
// Scilab Code Ex7.3: Page-411 (2011)
clc;clear;
a = 4e-003;....// Coherence width of laser source, m
lambda = 6e-007;....// Wavelength of the pulse, m
D = 100;....// Distance of the surface from laser
   source, m
A = 2*lambda/a;    // Angular spread of laser beam,
```

```
radian
printf("\nThe angular spread = %1.0e radian", A);
theta = A/2;     // Semi angle, radian
A_s = %pi*(D*theta)^2;....// Areal spread of laser beam, Sq.m
printf("\nThe areal spread = %1.0e Sq.m", A_s);

// Result
// The angular spread = 3e-004 radian
// The areal spread = 7e-004 Sq.m
```

Chapter 8

Ultrasonics

Scilab code Exa 8.1 Frequency of the fundamental mode of ultrasonic wave

```
// Scilab Code Ex8.1: Page-429 (2011)
clc;clear;
d = 8e-004;....// Thickness of the piece of
    piezoelectric crystal, m
v = 5760;....// Velocity of ultrasonic waves in the
    piece of piezoelectric crystal, m/s
n = v/(2*d);    // The frequency of the fundamental
    mode of ultrasonic wave, Hz
printf("\nThe frequency of the fundamental mode of
    ultrasonic wave = %3.1f MHz", n/1e+006);

4// Result
// The frequency of the fundamental mode of
    ultrasonic wave = 3.6 MHz
```

Scilab code Exa 8.2 Fundamental frequency of quarts crystal

```
// Scilab Code Ex8.2: Page-430 (2011)
```

```
clc;clear;
d = 2e-003;....// Thickness of the piece of quarts crystal, m
rho = 2650;....// Density of the crystal, kg/meter-cube
Y = 7.9e+010;....// Value of Youngs Modulus, N/metre-square
n = 1/(2*d)*sqrt(Y/rho);    //The frequency of the fundamental mode of vibration, Hz
printf("\nThe frequency of the fundamental mode of vibration in quatrz crystal = %5.3f Hz", n/1e+006);

// Result
// The frequency of the fundamental mode of vibration in quatrz crystal = 1.365 Hz
```

Scilab code Exa 8.3 Thickness of steel plate using ultrasonic beam

```
// Scilab Code Ex8.3: Page-430 (2011)
clc;clear;
v = 5e+003;....// Velocity of ultrasonic beam in steel plate, m/s
n = 25e+003;....// Difference between two neighbouring harmonic frequencies (Nm - Nm_minus1), Hz
d = v/(2*n);    // The thickness of steel plate, m
printf("\nThe thickness of steel plate = %3.1f m", d);

// Result
// The thickness of steel plate = 0.1 m
```

Scilab code Exa 8.4 Inductance of an inductor to produce ultrasonic waves

```
// Scilab Code Ex8.4: Page-430 (2011)
clc;clear;
n = 1e+006;....// Frequency of Ultrasonic waves, Hz
C = 2.5e-014;....// Capcitance of capacitor, F
// Frequency of elecric oscillations is given by n =
    1/(2*%pi)*sqrt(1/(L*C)), solving for L
L = 1/(4*%pi^2*n^2*C);    // The inductance of an
    inductor to produce ultrasonic waves, henry
printf("\nThe inductance of an inductor to produce
    ultrasonic waves = %d henry", L);

// Result
// The inductance of an inductor to produce
    ultrasonic waves = 1 henry
```

Scilab code Exa 8.5 Position of imperfection and the velocity of pulse inside the rod

```
// Scilab Code Ex8.5: Page-431 (2011)
clc;clear;
d = 50e-002;....// Thickness of the metallic rod, m
t1 = 30e-006;....// Arrival time for first pulse, s
t2 = 80e-006;....    // Arrival time for second
    pulse, s
v = 2*d/t2;.....// Velocity of ultrasonic waves, m/s
printf("\nThe velocity of pulse inside the rod = %4
    .2e m/s", v);
x = t1*v/2;
printf("\nThe position of pulse inside the rod = %6
    .4f m", x);

// Result
// The velocity of pulse inside the rod = 1.25e+004
```

```
m/s
// The position of pulse inside the rod = 0.1875 m
```

Scilab code Exa 8.6 Maximum acceleration and displacement of a quartz ultrasonic transducer

```
// Scilab Code Ex8.6: Page-431 (2011)
clc;clear;
I = 2.5e+004;....// Sound intensity, W/meter-square
v = 1480;....// Sound velocity, m/s
rho_w = 1000;....// Density of water, kg/meter-cube
rho_c = 2650;....// Density of crystal of transducer, kg/meter-cube
d = 0.001;....// Thickness of the quartz, m
f = 20e+003;....// Frequency of sound in water, Hz
// As sound intensity, I = p^2/(2*rho1*v), solving for p
p = sqrt(2*rho_w*v*I);    // Pressure in the medium, N/metre-square
a = p/(d*rho_c);    // Maximum acceleration of the quartz ultrasonic transducer, metre/second-square
printf("\nThe maximum acceleration produced in quartz transducer = %4.2e metre/second-square", a);
y = a/(2*%pi*f)^2;    // Maximum displacement of the quartz transducer, m
printf("\nThe maximum displacement of quartz transducer = %3.1f micron", y/1e-006);

// Result
// The maximum acceleration produced in quartz transducer = 1.03e+005 metre/second-square
// The maximum displacement of quartz transducer = 6.5 micron
```

Scilab code Exa 8.7 Fundamental frequency of a magnetostrictive hydrophone

```
// Scilab Code Ex8.7: Page-432 (2011)
clc;clear;
L = 0.2;....// Length of a magnetostrictive
    hydrophone, m
lambda = 2*L;....// Wavelength of ultrasonic wave, m
v = 4900;....// Velocity of ultrasonic beam in water
    , m/s
f = v/lambda;....// Fundamental frequency of
    ultrasonic, KHz
printf("\nThe fundamental frequency of a
    magnetostrictive hydrophone = %4.2f KHz", f/1e
    +03);

// Result
// The fundamental frequency of a magnetostrictive
    hydrophone = 12.25 KHz
```

Scilab code Exa 8.8 Length of the copper wire used to introduce ultrasonic delay

```
// Scilab Code Ex8.8: Page-432 (2011)
clc;clear;
v = 3700;....// Velocity of ultrasonic beam in
    copper, m/s
t = 1e-006;....// Delay time for ultrasonic beam, s
L = v*t;    // // Length of a copper wire required
    for a delay, m
printf("\nThe length of a copper wire required for a
     delay = %6.4f m", L);

```

```
// Result
// The length of a copper wire required for a delay
    = 0.0037 m
```

Chapter 9

Fibre Optics

Scilab code Exa 9.1 NA and the acceptance angle of optical fibre

```
// Scilab Code Ex9.1: Page-463 (2011)
clc;clear;
mu_1 = 1.55;....// Refractive index of the core
mu_2 = 1.50;....// Refractive indices of cladding
NA = mu_1*sqrt(2*(mu_1-mu_2)/mu_1);
printf("\nThe NA of the optical fibre = %5.3f", NA);
theta_a = asind(NA);    // The acceptance angle of
    optical fibre, degrees
printf("\nThe acceptance angle of the optical fibre
    is = %4.1f degrees", theta_a);

// Result
// The NA of the optical fibre = 0.394
// The acceptance angle of the optical fibre is =
    23.2 degrees
```

Scilab code Exa 9.2 NA acceptance angle and the critical angle of optical fibre

```
// Scilab Code Ex9.2: Page-463 (2011)
clc;clear;
mu_1 = 1.50;....// Refractive index of the core
mu_2 = 1.45;....// Refractive index cladding
NA = mu_1*sqrt(2*(mu_1-mu_2)/mu_1);     // Numerical aperture of optical fibre
printf("\n The NA of the optical fibre = %5.3f", NA);
theta_a = asind(NA);         // The acceptance angle of optical fibre, degrees
printf("\n The acceptance angle of the optical fibre = %5.2f degrees", theta_a);
theta_c = asind(mu_2/mu_1); // The critical angle of the optical fibre, degrees
printf("\n The acceptance angle of the optical fibre = %4.1f degrees", theta_c);

// Result
// The NA of the optical fibre = 0.387
// The acceptance angle of the optical fibre = 22.8 degrees
// The acceptance angle of the optical fibre = 75.2 degrees
```

Scilab code Exa 9.3 Acceptance angle for the optical fibre in water

```
// Scilab Code Ex9.3: Page-464 (2011)
clc;clear;
mu0 = 1;....// Refactive index of fibre in air
mu2 = 1.59;....// Refactive index of the cladding
NA = 0.2;....// Numerial aperture of optical fibre
mu1 = sqrt(NA^2+mu2^2);  // Refractive index of core
mu0 = 1.33;      // Refactive index of fibre in water
NA = sqrt(mu1^2-mu2^2)/mu0;     // Numerial aperture of optical fibre in water
```

```
theta_a = asind(NA);     // Acceptance angle for the fibre in water
printf("\nThe acceptance angle for the optical fibre in water = %3.1f degrees", theta_a);

// Result
// The acceptance angle for the optical fibre in water = 8.6 degrees
```

Scilab code Exa 9.4 The characteristics of glass clad fibre

```
// Scilab Code Ex9.4: Page-464 (2011)
clc;clear;
mu0 = 1;          // Refractive index of air
mu1 = 1.50;....// Refractive index of glass core'
del = 0.005;....// Fractional change in refractive index
mu2 = mu1*(1-del);       // Refractive index of cladding
printf("\nThe refractive index of cladding =%6.4f", mu2);
theta_c = asind(mu2/mu1);    // Critical angle, degrees
printf("\nThe critical angle = %5.2f degrees", theta_c);
theta_a = asind(sqrt(mu1^2-mu2^2)/mu0);      // Acceptance angle, degrees
printf("\nThe value of acceptance angle is = %4.2f degrees", theta_a);
NA = mu1*sqrt(2*del);       // Numerical aperture of optical fibre
printf("\nThe NA of the optical fibre = %4.2f", NA);

// Result
// The refractive index of cladding =1.4925
```

```
// The critical angle = 84.27 degrees
// The value of acceptance angle is = 8.62 degrees
// The NA of the optical fibre = 0.15
```

Scilab code Exa 9.5 Refractive index of core and cladding of an optical fibre

```
// Scilab Code Ex9.5: Page-465 (2011)
clc;clear;
NA = 0.22;        // Numerical aperture of the optical
    fibre
del = 0.012;....// Fractional difference between the
    refractive index of core and cladding
mu1 = NA/sqrt(2*del);    // The refractive index of
   core of optical fibre
printf("\nThe refractive index of core = %4.2f", mu1
   );
mu2 = mu1*(1-del);   // The refractive index of
   cladding of optical fibre
printf("\nThe refractive index of cladding = %4.2f",
    mu2);

// Result
// he refractive index of core = 1.42
// The refractive index of cladding = 1.40
```

Scilab code Exa 9.6 NA and the core radius of an optical fibre

```
// Scilab Code Ex9.6: Page-466 (2011)
clc;clear;
mu1 = 1.466;    // Refractive index of core
mu2 = 1.460;    // Refractive index of cladding
```

```
v = 2.4;....// Cut-off parameter of the optical
    fibre
lambda = 0.8e-006;....// Operating wavelength, m
NA = sqrt(mu1^2-mu2^2);
printf("\nThe NA of optical fibre = %4.2f", NA) ;
// Asthe cut-off parameter v of the optical fibre, v
     = 2*%pi*a*sqrt(mu1^2-mu2^2)/lambda, solving for
    a
a = lambda*v/(2*%pi*sqrt(mu1^2-mu2^2));
printf("\nThe core radius of the optical fibre = %4
    .2e micron", a/1e-006);

// Result
// The NA of optical fibre = 0.13
// The core radius of the optical fibre = 2.31e+00
    micron
```

Scilab code Exa 9.7 v number and the number of modes supported by the optical fibre

```
// Scilab Code Ex9.7: Page-466 (2011)
clc;clear;
mu1 = 1.54;     // The refractive index of core
mu2 = 1.50;     // The refractive index of cladding
lambda = 1.3e-006;....// Operating wavelength of
    optical fibre, m
a = 25e-006;....// Radius of fibre core, m
v = 2*%pi*a*sqrt(mu1^2-mu2^2)/lambda;    // V-number
    of optical fibre
printf("\nThe cut-off parameter of the optical fibre
     = %5.2f", v);
n = v^2/2;        // The number of modes supported by
    the fibre
printf("\nThe number of modes supported by the fibre
     = %3d", ceil(n));
```

```
// Result
// The cut-off parameter of the optical fibre =
   42.14
// The number of modes supported by the fibre = 888
```

Scilab code Exa 9.8 Maximum values of refractive index of cladding and the fractional change in refractive index

```
// Scilab Code Ex9.8: Page-466 (2011)
clc;clear;
mu1 = 1.54;....// Refractive index of core
v = 2.405;....// Cut-off parameter of optical fibre
lambda = 1.3e-006;....// Operating wavelength of
   optical fibre, m
a = 1e-006;....// Radius of the core,
NA = v*lambda/(2*%pi*a);     // Numerical aperture of
    optical fibre
del = 1/2*(NA/mu1)^2;    //  Fractional change in
   refractive index of core and cladding
printf("\nThe fractional difference of refractive
   indices of core and cladding = %7.5f", del);
mu2 = mu1*(1-del);      // Maximum value of
   refractive index of cladding
printf("\nThe maximum refractive index of cladding =
    %5.3f", mu2);

// Result
// The fractional difference of refractive indices
   of core and cladding = 0.05220
// The maximum refractive index of cladding = 1.460
```

Scilab code Exa 9.9 Normalized frequency for the optical fibre

```
// Scilab Code Ex9.9: Page-467 (2011)
clc;clear;
mu1 = 1.45;....// Index of refraction of core
NA = 0.16;....// Numerical aperture of step index
   fibre
a = 3e-006;....// Radius of the core, m
lambda = 0.9e-006;....// Operating wavelength of
   optical fibre, m
v = 2*%pi*a*NA/lambda;      // The normalized
   frequency or v-number of optical fibre
printf("\nThe normalized frequency of the optical
   fibre = %5.2f", v);

// Result
// The normalized frequency of the optical fibre =
   3.35
```

Scilab code Exa 9.10 Cut off parameter or v number of modes supported by the fibre

```
// Scilab Code Ex9.10: Page-467 (2011)
clc;clear;
mu1 = 1.52;....// Refractive index of core
a = 14.5e-006;....// Radius of the fibre core, m
del = 0.0007;....// Fractional index difference
lambda = 1.3e-006;....// Operating wavelength of
   optical fibre, m
mu2 = mu1*(1-del);      // Refractive index of
   cladding
v = 2*%pi*a*sqrt(mu1^2-mu2^2)/lambda;      // Cut-off
    parameter v of the optical fibre
printf("\nThe cut-off parameter of the optical fibre
    = %5.3f", v);
//The is number of modes supported by the fibre
   given by,
```

```
n = v^2/2;
printf("\nThe number of modes supported by the fibre
    = %d", ceil(n));

// Result
// The cut-off parameter of the optical fibre =
   3.985
// The number of modes supported by the fibre = 8
```

Scilab code Exa 9.11 Power output through optical fibre

```
// Scilab Code Ex9.11: Page-468 (2011)
clc;clear;
alpha = 3.5;....// Attenuation of the optical fibre,
    dB/km
Pi = 0.5;....// Input power of optical fibre, mW
L = 4;....     // Distance through the optical wave
   transmits through the fibre, km
// As alpha = 10/L*log10(Pi/Po), solving for Po
Po = Pi/exp(alpha*L*2.3026/10); // Output power of
   optical fibre, mW
printf("\nThe output power of optical fibre = %4.1f
   micro-watt", Po/1e-003);

// Result
// The output power of optical fibre = 19.9 micro-
   watt
```

Scilab code Exa 9.12 Attenuation of power through optical fibre

```
// Scilab Code Ex9.12: Page-468 (2011)
clc;clear;
Pi =1;....// Input power of optical fibre, mW
```

```
Po = 0.85;....// Outptu power of optical fibre , mW
L = 0.5;....//The distance through the optical wave transmits through the fibre , km
alpha = (10/L)*log10(Pi/Po);       // The attenuation of power through the optical fibre
printf("\nThe attenuation of power through the optical fibre = %5.3f dB/km", alpha);

// Result
// The attenuation of power through the optical fibre = 1.412 dB/km
```

Scilab code Exa 9.13 Minimum optical power input to an optical fibre

```
// Scilab Code Ex9.13: Page-469 (2011)
clc;clear;
C = 0.8;     // Connector loss per km, dB
F = 1.5;     // Fibre loss per km, dB
alpha = C + F;....// Attenuation of power the optical fibre , dB/km
Po = 0.3e-006;....// Output power of optical fibre , W
L = 15;....// The distance through the optical wave transmits through the fibre , km
//As the attenuation , alpha = 10/L*log(Pi/Po), solving for Pi
Pi = Po*exp(2.3026*alpha*L/10);      // Input power of optical fibre , mW
printf("\nThe minimum input power to optical fibre = %5.3f mW", Pi/1e-003);

// Result
// The minimum input power to optical fibre = 0.846 mW
```

Chapter 10

Electrostatics

Scilab code Exa 10.1 Potential difference between the two charged horizontal plates

```
// Scilab Code Ex10.1: Page-507 (2011)
clc;clear;
m = 4e-013;....// Mass of the particle , kg
q = 2.4e-019;....// Charge on particle , C
d = 2e-002;....// Distance between the two
    horizontally charged plates , m
g = 9.8;....//‘Acceleration due to gravity , m/sec-
    square
E = m*g/q ;....// Electric field strength , N/C
V = E*d;.....// Potential difference between the two
     charged horizontal plates , V
printf("\nThe potential difference between the two
    horizontally charged plates = %3.1e V", V);

// Result
// The potential difference between the two
    horizontally charged plates = 3.3e+005 V
```

Scilab code Exa 10.2 Electric potential at a point equidistant from the three corners of a triangle

```
// Scilab Code Ex10.2: Page-507 (2011)
clc;clear;
q1 = 1e-009;     // Charge at first corner, C
q2 = 2e-009;     // Charge at second corner, C
q3 = 3e-009;     // Charge at third corner, C
d = 1;....// Side of the equilateral triangle, m
theta = 30;....// Angle at which line joining the
    observation point to the source charge makes with
     the side, degrees
r = (d/2)/cosd(theta);....// Distance of observation
     point from the charges, m
//since,1/4*%pi*%eps = 9e+009;
V = (q1+q2+q3)*(9e+009)/r;.....// Elecric potential,
    V
printf("\nThe electric potential at the point
    equidistant from the three corners of the
    triangle = %4.1f V", V);

// Result
// The electric potential at the point equidistant
    from the three corners of the triangle = 93.5 V
```

Scilab code Exa 10.3 Electric potential at the centre of a square

```
// Scilab Code Ex10.3: Page-507 (2011)
clc;clear;
q = 2e-008;
q1 = q;         // Charge at first corner, C
q2 = -2*q;      // Charge at second corner, C
q3 = 3*q;       // Charge at third corner, C
q4 = 2*q;       // Charge at fourth corner, C
d = 1;....     // Side of the square, m
```

```
r = d*sin(2*%pi/8);....// Distance of centre of the square from each corner, m
V = (q1+q2+q3+q4)*(9e+009)/r;.....// Elecric potential at the centre of the square, V
printf("\nThe electric potential at the centre of the square = %4d V", V);

// Result
// The electric potential at the centre of square = 1018 V
```

Scilab code Exa 10.6 New potential when the two charged drops coalesce to form a bigger drop

```
// Scilab Code Ex10.6: Page-512 (2011)
clc;clear;
V = 60;....// Electric potential of smaller drop, volt
r = 1;....// For simplcity assume radius of each small drop to be unity, unit
q = 1;....// For simplicity assume charge on smaller drop to be unity, C
k = 1;....// For simplicity assume Coulomb's constant to be unity, unit
R = 2^(1/3)*r;.....// Radius of bigger drop, unit
Q = 2*q;....// Charge on bigger drop, C
V_prime = k*Q/R*V;....// Electric potential of bigger drop, volt
printf("\nThe electric potential of new drop = %4.1f V", V_prime);

// Result
// The electric potential of new drop = 95.2 V
```

Scilab code Exa 10.7 Magnitude and the direction of electric field which would balance the weight of an electron placed in it

```
// Scilab Code Ex10.7: Page-512 (2011)
clc;clear;
m = 9.1e-031;....// Mass of the electron , kg
e = 1.6e-019;....// Charge on an electron , C
g = 9.8;....// Acceleration due to gravity , m/sec-
   square
// Electric force , F = e*E, where F = m*g or e*E = m
   *g
E = m*g/e;     // Electric field which would balance
   the weight of an electron placed in it , N/C
printf("\nThe required electric field strength = %3
   .1e N/C", E);
printf("\nThis field acts opposite to the direction
   of weight");

// Result
// The required electric field strength = 5.6e-011 N
   /C
// This field acts opposite to the direction of
   weight
```

Scilab code Exa 10.8 Magnitude and the direction of electric field at a point midway between two charges

```
// Scilab Code Ex10.8: Page-512 (2011)
clc;clear;
q1 = 8e-007;....// First Charge , C
q2 = -8e-007;....// Second Charge , C
```

```
r = 15e-002;....// Distance between the two charges, m
k = 9e+009; // Coulomb's constant, N-metre-square/coulomb-square
E1 = k*q1/r^2;....// Electric field strength due to charge 8e-007 C
printf("\nThe electric field strength at midpoint = %3.1e N/C", E1);
E2 = abs(k*q2/r^2);....// Electric field strength -8e-007 C
printf("\nThe electric field strength at midpoint = %3.1e N/C", E2);
// Total electric field strength at the mid-point is
E = E1+E2;       // Net electric field at mid point, N/C
printf("\nThe net electric field strength at midpoint = %3.1e N/C", E);

// Result
// The electric field strength at midpoint = 3.2e+05 N/C
// The electric field strength at midpoint = 3.2e+05 N/C
// The net electric field strength at midpoint = 6.4e+05 N/C
```

Scilab code Exa 10.9 Electric field strength at a point

```
// Scilab Code Ex10.9: Page:513 (2011)
clc;clear;
x = poly(0, 'x');
V = 1000/x+1500/x^2+500/x^3;     // Given electric potential at a point (x,0,0), V
E = -1*derivat(V);  // Electric field at a point as gradient of scalar potential, N/C
```

```
E_x = horner(E, 1);       // Electric field at the point x = 1, N/C
printf("\nThe electric field strength at point x = 1 is %4di V/m", E_x);

// Result
// The electric field strength at point x = 1 is 5500i V/m
```

Scilab code Exa 10.11 Electric field strength due to spherical charge distribution

```
// Scilab Code Ex10.11: Page:514 (2011)
clc;clear;
function e = E(r)
    a = 1;    // For convenience assume radius of sphere to be unity
    r = poly(0, 'r');
    e = r/3-r^3/(5*a^2);
endfunction

rho_0 = 1;    // For convenience assume charge density to be unity
epsilon_0 = 1;    // For convenience assume permittivity to be unity
r = poly(0, 'r');
E_int = rho_0/epsilon_0*E('r');
delta_E = derivat(E_int);
r = roots(delta_E);
printf("\nThe electric field strength is maximum at an internal point at a distance r = sqrt(%g)a/3 from the centre", (3*r(1))^2);

// Result
// The electric field strength is maximum at an
```

```
internal point at a distance r = sqrt(5)a/3 from the centre
```

Scilab code Exa 10.12 Maximum electric field strength at an internal point

```
// Scilab Code Ex10.12: Page:517 (2011)
clc;clear;
function e = E(r)
    a = 1;    // For convenience assume radius of sphere to be unity
    r = poly(0, 'r');
    e = r/3-r^2/(4*a);
endfunction

rho_0 = 1;    // For convenience assume charge density to be unity
epsilon_0 = 1;    // For convenience assume permittivity to be unity
r = poly(0, 'r');
E_int = rho_0/epsilon_0*E('r');
delta_E = derivat(E_int);
r = roots(delta_E);
printf("\nThe electric field strength is maximum at an internal point at a distance r = %da/3 from the centre", 3*r);

// Result
// The electric field strength is maximum at an internal point at a distance r = 2a/3 from the centre
```

Chapter 11

Electromagnetic Theory

Scilab code Exa 11.1 Amplitude of field vector E in free space

```
// Scilab Code Ex11.1: Page-559 (2011)
clc;clear;
H_0 = 1;....// Amplitude off field vector,in A/m
mu_0 = 12.56e-7;....// Permeability,in weber/A-m
eps = 8.85e-12;....// Permittivity in free space,in
    C/N-meter-square
// From the relation between the amplitude of the
    field vector E and vector H of an EM wave in free
    space
E_0 = H_0*(sqrt(mu_0/eps));
printf("\nThe amplitude of field vector E in free
    space = %5.1f V/m",E_0);


// Result
// The amplitude of field vector E in free space =
    376.7 V/m
```

Scilab code Exa 11.2 Maximum value of magnetic induction vector

```
// Scilab Code Ex11.2: Page-560 (2011)
clc;clear;
E_o = 1e+3;....// Amplitude field vector in free space,N/C
c = 3e+8;....// Speed of light,in m/s
// From the relation between the amplitude of the field vector E and vector H of an EM wave in free space E_o = H_o*(sqrt(mu_o/eps))and B_o = mu_o*H_o, we have
B_o = E_o/c;
printf("\nThe maximum value of magnetic induction vector = %4.2e weber/A-m",B_o);

// Result
// The maximum value of magnetic induction vector = 3.33e-006 weber/A-m
```

Scilab code Exa 11.3 Conduction and displacement current densities in the conducting medium

```
// Scilab Code Ex11.3: Page-560 (2011)
clc;clear;
sigma = 5;....// Conductivity of the conducting medium, mho/m
eps_r = 8.85e-12;....// Relative electrical permittivity of medium, F/m
eps_0 = 1;    // Electrical permittivity of free space, F/m
E0 = 250;    // Amplitude of applied electric field, V/m
J = sigma*E0;    // Amplitude of conduction current density, A/metre-square
J_D = eps_r*eps_0*E0*1e+010;    // Amplitude of displacement current density, A/metre-square
omega = sigma/(eps_0*eps_r);    // Frequency at
```

```
which J = J_D
printf("\nThe conduction current density = %3dsin
    (10^10t) A/metre-quare", J);
printf("\nThe displacement current density = %5.3
    fcos(10^10t) A/metre-quare", J_D);
printf("\nThe frequency at which J = J_D is %3.1e Hz
    ", omega);

// Result
// The conduction current density = 1250sin(10^10t)
    A/metre-quare
// The displacement current density = 22.125cos
    (10^10t) A/metre-quare
// The frequency at which J = J_D is 5.6e+11 Hz
```

Scilab code Exa 11.8 Average value of the intensity of electric field of radiation

```
// Scilab Code Ex11.8 :Page-565 (2011)
clc;clear;
P = 1000;....// Energy radiated by the lamp, watt
r = 2;....// Distance from the source at which the
    electric field intensity is given, m
S = P/(4*%pi*r^2);       // Magnitude of Poynting
    vector, W/metre-square
// As wave imepdence, Z0 = E/H = 377 and H = E/377,
    so that with E*H = S we have
E = poly(0, 'E');
E = roots(E*E/377-S);
printf("\nThe average value of the intensity of
    electric field of radiation = %4.1f V/m", E(2));

// Result
// The average value of the intensity of electric
    field of radiation = 86.6 V/m
```

Scilab code Exa 11.9 Amplitude of electric and magnetic fields of radiation

```
// Scilab Code Ex11.9: Page-566 (2011)
clc;clear;
S = 2*4.186/60*1e+04;....// Solar constant, J/s/
    metre-square
// From the poynting vector S = E*H
C = 377;....// Wave Impedence, ohm
E = sqrt(S*C);        // Electric field of radiation,
    V/m
H = E/C;              // Magnetic field of radiation,
    A/m
E0 = E*sqrt(2);       // Amplitude of electric field
    of radiation, V/m
H0 = H*sqrt(2);       // Amplitude of magnetic field
    of radiation, A/m
printf("\nThe amplitude of electric field of
    radiation = %6.1f V/m", E0);
printf("\nThe amplitude of magnetic field of
    radiation = %5.3f V/m", H0);


// Result
// The amplitude of electric field of radiation =
    1025.7 V/m
// The amplitude of magnetic field of radiation =
    2.721 V/m
```

Scilab code Exa 11.10 Phase difference between electric and magentic field vectors of an EM wave

```
// Scilab Code Ex 11.10: Page-568 (2011)
clc;clear;
function s = sine(x)
    s = x - x^3/factorial(3) + x^5/factorial(5) - x^7/factorial(7) + x^9/factorial(9);
endfunction

function s = cosine(x)
    s = 1 - x^2/factorial(2) + x^4/factorial(4) - x^6/factorial(6) + x^8/factorial(8);
endfunction

k = 1;  // For simplicity assume constant of proportionality to be unity, units
for theta = 1:1:45
alpha = k*cosd(theta);
b = k*sind(theta);
    if alpha == b then
        phi = atand(b/alpha);
        break;
    end
end
//printf("\nThe phase difference between electric and magentic field vectors = %4.2f rad", phi);


// Result
// The skin depth of an EM-wave in Al = 0.000010 m
```

Scilab code Exa 11.12 Skin depth of an EM wave in Al

```
// Scilab Code Ex11.12: Page-569 (2011)
clc;clear;
sigma = 3.54e+007;....// Electrical conductivity of Al, mho per metre
```

```
mu = 12.56e-007;....// Permeability of the medium, weber/A-m
f = 71.6e+06;     // Frequency of the wave, Hz
omega = 2*%pi*f;....// Angular frequency of the wave, rad per sec
delta = sqrt(2/(sigma*mu*omega));     // Skin depth of the EM wave in Al, m
printf("\nThe skin depth of an EM-wave in Al = %2.0f micron", delta/1e-06);

// Result
// The skin depth of an EM-wave in Al = 10 micron
```

Scilab code Exa 11.14 Skin depth and attenuation constant of sea water

```
// Scilab Code Ex11.14: Page-571 (2011)
clc;clear;
sigma = 5;....// Electrical conductivity, mho per metre
mu = 12.56e-007;....// Permeability of the medium, weber/A-m
eps_0 = 8.85e-012;....// Electric permittivity of free space, C-square/N-m-square
eps = 70*eps_0;     // Electric permittivity of the medium, C-square/N-m-square
delta = 2/sigma*sqrt(eps/mu); // The skin depth and attenuation constant of sea water
printf("\nThe skin depth of an EM-wave in sea water = %6.4f m", delta);
Beta = 1/delta;     // The attenuation constant of sea water, per metre
printf("\nThe attenuation constant of the sea water = %6.2f m", Beta);

// Result
```

```
// The skin depth of an EM-wave in sea water =
   0.0089 m
// The attenuation constant of the sea water =
   112.57 m
```

Chapter 12

Magnetic Properties of Materials

Scilab code Exa 12.1 Current through the solenoid

```
// Scilab Code Ex12.1 Page-603 (2011)
clc; clear;
H = 5e+3;....// Coercivity of a bar magnet, A/m
L = 0.1;....// Length of the solenoid, m
N = 50;....// Turns in solenoid
n = 500;....// Turns/m
// Using the relation
I = H/n;....// where I is the current through the
   solenoid
printf("\nThe current through the solenoid is = %2d
   A", I);

// Result
// The current through the solenoid is = 10 A
```

Scilab code Exa 12.2 Magnetic moment of the iron rod

```
// Scilab Code Ex12.2 : Page-603 (2011)
clc; clear;
n = 500;....// Number of turns wound per metre on
    the solenoid
i = 0.5;....// Current through the solenoid, A
V = 1e-03;....// Volume of iron rod, per metre cube
mu_r = 1200;     // Relative permeability of the iron
H = n*i;     // Magnetic intensity inside solenoid,
    ampere-turn per metre
// As B = mu_o * (H + I) => I = B/mu_o -  H
// But B = mu_o * mu_r * H and solving for I
I = (mu_r - 1) * H;
printf("\nThe Intensity of magnetisation inside the
    solenoid, I = %5.3e A/m", I);
M = I * V;     // Magnetic moment of the rod, ampere
    metre square
printf("\nThe magnetic moment of the rod, M = %3d
    ampere metre square", M)

//Result
// The Intensity of magnetisation inside the
    solenoid, I = 2.998e+005 A/m
// The magnetic moment of the rod, M = 299 ampere
    metre square
```

Scilab code Exa 12.3 Magnetic moment of the rod placed inside the solenoid

```
// Scilab Code Ex12.3 : Page-604 (2011)
clc; clear;
n = 300;....// Number of turns wound per metre on
    the solenoid
i = 0.5;....// Current through the solenoid, A
V = 1e-03;....// Volume of iron rod, per metre cube
mu_r = 100;     // Relative permeability of the iron
H = n*i;     // Magnetic intensity inside solenoid,
```

```
    ampere−turn per metre
// As, I = (B−mu_o* H)/mu_o
//But, B= mu * H = mu_r * mu_o * H and I = (mu_r−1)*
    H
I = (mu_r-1)*n*i;
printf("\nThe Intensity of magnetisation inside the
    solenoid , I = %5.3e A/m", I);
l = 0.2;....//length of the rod ,m
r = 5e-3;....//radius of the rod ,m
V = 1.57e-5;....//V=%pi*r^2*l where the volume of
    the rod having radius r and length ,m
 M = I * V ;    // Magnetic moment of the rod ,
    ampere metre square
printf("\nThe magnetic moment of the rod , M = %5.3f
    ampere metre square",M)

//Result
// The Intensity of magnetisation inside the
    solenoid , I = 1.485e+004 A/m
// The magnetic moment of the rod , M = 0.233 ampere
    metre square
```

Scilab code Exa 12.4 Magnetizing force and relative permeability of the material

```
// Scilab Code Ex12.4 : Page−605 (2011)
clc; clear;
B = 0.0044;....// Magnetic flux density , weber/meter
    square
mu_o = 4*%pi*1e-07;....// Relative permeability of
    the material , henery/m
I = 3300;....// Magnetization of a magnetic material
    , A/m
//B = mu_o*(I+H), solving for H
H = (B/mu_o)- I;....// Magnetizing force ,A/m
```

```
printf("\nThe magnetic intensity ,H = %3d A/m",H);
// Relation between intensity of magnetization and
   relative permeability
mu_r = (I/H)+1;....//substitute the value of I and H
printf("\nThe relative permeability , mu_r = %5.2f",
   mu_r);

//Result
// The magnetic intensity ,H = 201 A/m
// The relative permeability , mu_r = 17.38
```

Scilab code Exa 12.5 Magnetic flux density and magnetic intensity

```
// Scilab Code Ex12.5 : Page-605 (2011)
clc; clear;
mu_o = 4*%pi*1e-07;....// Magnetic permeability of
   the free space, henery/m
mu_r = 600;
mu = mu_o*mu_r;      // Magnetic permeability of the
   medium, henery/m
n = 500;...// Turns in a wire
i = 0.3;....// Current flows through a ring ,amp
r = 12e-02/2;....// Mean radius of a ring , m
B = mu_o*mu_r*n*i/(2*%pi*r);
printf("\nThe magnetic flux density = %2.1f weber/
   meter-square", B);
H = B/mu;   // Magnetic intensity , ampere-turns/m
printf("\nThe magnetic intensity = %5.1f ampere-
   turns/m", H);
// As B = mu_o*(I + H) => mu_o*I = B - mu_o*H
printf("\nThe percentage magnetic flux density due
   to electronic loop currents = %5.2f percent", (B
   - mu_o*H)/B*100);

// Result
```

```
// The magnetic flux density = 0.3 weber/meter-
   square
// The magnetic intensity = 397.9 ampere-turns/m
// The percentage magnetic flux density due to
   electronic loop currents = 99.83 percent
```

Scilab code Exa 12.6 Total dipole moment of the sample

```
// Scilab Code Ex12.6 : Page-606 (2011)
clc; clear;
M_i = 4.5;....// Intial value of total dipole moment
    of the sample
H_i = 0.84;....// External magnetic field, tesla
T_i = 4.2;....// Cooling temerature of the sample, K
H_f = 0.98;....// External magnetic field, tesla
T_f = 2.8;....// Cooling temerature of the sample, K
// According to the curie's law, Mf/Mi = (Hf/Hi)*(Ti
   /Tf)
M_f = M_i*H_f/H_i*T_i/T_f;
printf("\nThe total dipole moment of the sample = %5
   .3f joule/tesla",M_f);

// Result
// The total dipole moment of the sample = 7.875
   joule/tesla
```

Scilab code Exa 12.7 Magnetization and magnetic moment in the bar

```
// Scilab Code Ex12.7 : Page-606 (2011)
clc; clear;
mu_o = 4*%pi*1e-07;....// Magnetic permeability of
   free space, henry/m
```

```
n = 1e+29;....// Number density of atoms of iron, per metre cube
p_m = 1.8e-23;....// Magnetic moment of each atom, ampere-metre square
k_B = 1.38e-23;....// Boltzmann constant, J/K
B = 0.1;     // Magnetic flux density, weber per metre square
T = 300;....// Absolute room temperature, K
l = 10e-02; // Length of the iron bar, m
a = 1e-04;  // Area of cross-section of the iron bar, metre square
V = l*a;    // Voluem of the iron bar, metre cube
chi = n*p_m^2*mu_o/(3*k_B*T);
printf("\nThe paramagnetic susceptibility of a material = %5.3e", chi);
pm_mean = p_m^2*B/(3*k_B*T); // Mean dipole moment of an iron atom, ampere metre-square
P_m = n*pm_mean;  // Dipole moment of the bar, ampere metre-square
I = n*p_m;  // Magnetization of the bar in one domain, ampere/metre
M = I*V;    // Magnetic moment of the bar, ampere metre-square
printf("\nThe dipole moment of the bar = %5.3e ampere metre-square", P_m);
printf("\nThe magnetization of the bar in one domain = %3.1e ampere/metre", I);
printf("\nThe magnetic moment of the bar = %2d ampere metre-square", M);

// Result
// The paramagnetic susceptibility of a material = 3.278e-03
// The dipole moment of the bar = 2.609e+02 ampere metre-square
// The magnetization of the bar in one domain = 1.8e+06 ampere/metre
// The magnetic moment of the bar = 18 ampere metre-
```

```
square
```

Scilab code Exa 12.8 Hysteresis loss of energy per hour in the iron core of the transformer

```
// Scilab Code Ex12.8 :Page-607 (2011)
clc; clear;
A = 500;....// Area of the B-H loop, joule per metre
    cube
n = 50;....// Total number of cycles, Hz
m = 9;....// Mass of the core, kg
d = 7.5e+3;....// Density of the core, kg/metre cube
t = 3600;....// Time during which the energy loss
    takes place, s
V = m/d;....// Volume of the core, metre cube
E = n*V*A*t;....// Hystersis loss of energy per hour
    , joule
printf("\nThe hystersis loss per hour = %5.2eJ", E);

// Result
// The hystersis loss per hour = 1.08e+005J
```

Scilab code Exa 12.9 Hystersis loss from BH loop

```
// Scilab Code Ex12.9 : Page-607 (2011)
clc; clear;
n = 50;....// Total number of cycles per sec, Hz
V = 1e-03;....// Volume of the specimen, metre cube
t = 1;....// Time during which the loss occurs, s
A = 0.25e+03;....// Area of B-H loop, joule per
    metre cube
E = n*V*A*t;     // Energy loss due to hysteresis, J/
    s
```

```
printf("\nThe hystersis loss = %4.1f J/s", E);

// Result
// The hystersis loss = 12.5 J/s
```

Scilab code Exa 12.10 Change in the magnetic dipole moment of the electron

```
// Scilab Code Ex12.10 : Page-608 (2011)
clc; clear;
e = 1.6e-19;....// Charge on anlectron , C
m = 9.1e-31;....// Mass of the electron , kg
r = 5.1e-11;....// Radius of the electronic orbit , m
B = 2.0;....// Applied magnetic field , weber per
    metre-square
delta_pm = e^2*r^2*B/(4*m);
printf("\nThe change in the magnetic dipole moment
    of the electron = %3.1e A-metre square", delta_pm
    );

// Result
// The change in the magnetic dipole moment of the
    electron = 3.7e-29 A-metre square
```

Chapter 13

Dielectric Properties of Materials

Scilab code Exa 13.1 Electric dipole placed in a uniform electric field

```
// Scilab Code Ex13.1: Page-648 (2011)
clc;clear;
q = 1e-006;     // Electric charge on either side of
    the dipole , C
l = 2e-02;     // Dipole length , m
p = q*l;....// Dipole moment for the pair of
    opposite charges , C-m
E = 1e+005;....// External electric field , N/C
theta = 90;....// Angle which the dipole makes with
    the external field , degrees
tau = p*E*sind(theta);....// The maximum torque on
    dipole placed in external electric field , Nm
printf("\nThe maximum torque = %1.0e N-m", tau);
W = integrate('p*E*sin(thet)', 'thet', 0, %pi);
    // The work done in rotating the dipole direction
    = %1.0e J", W
printf("\nThe work done in rotating the dipole
    direction = %1.0e J", W);
```

```
// Result
// The maximum torque = 2e-003 N-m
// The work done in rotating the dipole direction = 4e-003 J
```

Scilab code Exa 13.2 Force acting on an electric dipole in different orientations relative to the electric field

```
// Scilab Code Ex13.2:Page-648 (2011)
clc;clear;
Q = 8e-019;....// Charge of the nucleus, C
p = 3.2e-029;....// Electric dipole moment, C-m
r = 1e-10;    // Distance of dipole relative to the nucleus, m
k = 9e+9;....// Coulomb constant, N-meter-square/C-square
theta = 0;....// Angle for radial direction, radian
F = k*p*Q*sqrt(3*cos(theta^2)+1)/r^3;    // The force acting on the dipole in the radial direction, N
printf("\nThe force acting on the dipole in the radial direction = %3.1e N", F);
theta = %pi/2;....// Angle for perpendicular direction, radian
F = k*p*Q*sqrt(3*cos(theta)^2+1)/r^3;
printf("\nThe force acting on the dipole in the direction perpendicular to radial direction = %3.1e N", F);

// Result
// The force acting on the dipole in the radial direction = 4.6e-007 N
// The force acting on the dipole in the direction perpendicular to radial direction = 2.3e-007 N
```

Scilab code Exa 13.3 Dielectric constant and the electric permittivity of the material

```
// Scilab Code Ex13.3: Page-649 (2011)
clc;clear;
chi_e = 35.4e-12;....// Susceptability of the
    material, C-square/N-meter-square
eps_0 = 8.85e-12;....// Electric permittivity in
    free space, C-squre/N-meter-square
K = 1 + (chi_e/eps_0);
printf("\nThe dielectric constant = %d ",K);
eps = (eps_0*K);
printf("\nThe electric permittivity  = %5.3e C-
    square/N-meter square ",eps);

// Result
// The dielectric constant = 5
// The electric permittivity  = 4.425e-011 C-square/
    N-meter square
```

Scilab code Exa 13.4 Dielectric constant and the electric susceptability of diamond

```
// Scilab Code Ex13.4:Page-649 (2011)
clc;clear;
eps = 1.46e-10;....// Electric permittivity, C-
    square/n-meter-square
eps_0 = 8.85e-12;....// Permittivity in free space,
    C-squre/N-meter-square
K = (eps/eps_0);
printf("\nThe dielectric constant = %4.1f ", K);
```

```
chi_e = eps_0*(K-1);....// Susceptability ,in C-square/N-meter-square
printf("\nThe electric susceptability = %4.2e C-square/N-meter square ", chi_e);

// Result
// The dielectric constant = 16.5
// The electric susceptability = 1.37e-010 C-square/N-meter square
```

Scilab code Exa 13.5 Calculate the values of E D and P

```
// Scilab Code Ex13.5 Page-650 (2011)
clc;clear;
K = 7.0;....// Dielectric constant of the slab
d = 0.01;....// Distance between the two parallel plates , m
V_0 = 100;....// Potential difference across the plates , V
eps_0 = 8.85e-12;....// Electric permability of the free space , C-square/N-meter-square
E_0 = V_0/d;....// Electric intensity in the absence of dielectric slab , V/m
E = E_0/K;     // Electric intensity with dielectric slab introduced between the plates , V/m
printf("\nThe electric field intensity in the presence of the dielectric slab = %4.2e V/m ", E);
D = (eps_0*K*E);     // Electric displacement , C-square/m-square
printf("\nThe electric displacement in the dielectric slab = %4.2e C-square/meter-square ",D);
P = eps_0*(K-1)*E;     // Electric polarization in the dielectric slab , C-square/m-square
```

```
printf("\nThe electric polarization in the
    dielectric slab = %3.1e C-square/meter-square ",P
    );

// Result
// The electric field intensity in the presence of
    the dielectric slab = 1.43e+003 V/m
// The electric displacement in the dielectric slab
    = 8.85e-008 C-square/meter-square
// The electric polarization in the dielectric slab
    = 7.6e-008 C-square/meter-square
```

Scilab code Exa 13.6 Dipole moment induced in He atom

```
// Scilab Code Ex13.6: Page-650 (2011)
clc;clear;
K = 1.000074;....// Dielectric constant of the He
n = 2.69e+025;....// Atomic density of He, atoms/
    meter-cube
eps_0 = 8.85e-012;....// Electric permability of the
     free space, C-square/N-meter-square
E = 1;....// Electric field strength, V/m
p = (eps_0*(K-1)*E)/n;    // Dipole moment induced
    in He, C-m
printf("\nThe dipole moment induced in each He atom
    = %4.2e C-m ", p);

// Result
// The dipole moment induced in each He atom = 2.43e
    -041 C-m
```

Scilab code Exa 13.7 Induced dipole moment and atomic polarizability of neon gas

```
// Scilab Code Ex13.7: Page-650 (2011)
clc;clear;
K = 1.000134;....// Dielectric constant of the neon
n = 2.69e+25;....// Atomic density of argon,atoms/
    meter-cube
eps_0 = 8.85e-12;....// Electric Permability in the
    free space, C-square/N-meter-square
E = 90e+03;    // External electric field, V/m
p = eps_0*(K-1)*E/n;    // Dipole moment induced in
    each neon atom, C-m
alpha = p/E;    // Atomic polarizability of neon gas
    , C-metre-square/V
printf("\nThe induced dipole moment of noen atom =
    %4.2e C-m", p) ;
printf("\nThe electronic polarizability of neon gas
    = %3.1e C-m-square/V ", alpha);

// Result
// The induced dipole moment of noen atom = 3.97e
    -036 C-m
// The electronic polarizability of neon gas = 4.4e
    -041 C-m-square/V
```

Scilab code Exa 13.8 Electronic polarizability of argon atom

```
// Scilab Code Ex13.8: Page-651 (2011)
clc;clear;
K = 1.0024;....// Dielectric constant of the argon
n = 2.7e+25;....// Atomic density of argon,atoms/
    meter-cube
eps_0 = 8.85e-12;....// Electric Permability in the
    free space, C-square/N-meter-square
alpha = eps_0*(K-1)/n;
printf("\nThe electronic polarizability of argon
    atom = %4.1e C-m-square/V ", alpha);
```

```
// Result
// The electronic polarizability of argon atom = 7.9
   e-040 C-m-square/V
```

Scilab code Exa 13.9 Individual dipole moment of carbon tetrachloride

```
// Scilab Code Ex13.9: Page-651 (2011)
clc;clear;
K = 2.24;....// Dielectric constant
eps_0 = 8.85e-12;....// Electric permability in the
   free space, C-square/N-meter-square
rho = 1.6e+003;....// Density of CCl4, kg/meter-cube
M = 156;....// Molecular weight of CCl4
E = 1e+007;....// External electric field strength,
   V/m
N_A = 6.02e+26;    // Avogadro's number, per kmol
rho_M = rho*N_A/M;     // Molecular density of CCl4
p = eps_0*(K-1)*E/rho_M;    // Individual dipole
   moment of CCL4 molecule, C-m
printf("\nIndividual dipole moment of CCL4 molecule
   = %4.2e C-m ", p);

// Result
// Individual dipole moment of CCL4 molecule = 1.78e
   -032 C-m
```

Scilab code Exa 13.10 Atomic radius of He

```
// Scilab Code Ex13.10: Page-652 (2011)
clc;clear;
K = 1.0000684;....// Dielectric constant of He at 1
   atm
```

```
n = 2.7e+25;....// Density of He at 1 atm and 273 K,
    atoms/meter-cube
// The atomic polarizibility , alpha = eps_0*(K-1)/n
// In terms of atomic radius , alpha = 4*%pi*eps_0*R
   ^3 so , we have
R = ((K-1)/(4*%pi*n))^(1/3);     // Radius of He atom
   , m
printf("\nThe atomic radius of He = %4.2e m ", R);

// Result
// The atomic radius of He = 5.86e-011 m
```

Scilab code Exa 13.11 Percentage of ionic polarizability in NaCl crystal

```
// Scilab Code Ex13.11: Page-652 (2011)
clc;clear;
mu = 1.5;....// Optical index of refraction of NaCl
   crystal
K = 5.6;....// Static dielectric constant of NaCl
   crystal
P_IP = (1-((mu^2-1)*(K+2))/((mu^2+2)*(K-1)))*100;
printf("\nThe percentage of ionic polarizibility in
   NaCl crystal = %4.1f percent ", P_IP);

// Result
// The percentage of ionic polarizibility in NaCl
   crystal = 51.4 percent
```

Scilab code Exa 13.12 Determine the dipole moment

```
// Scilab Code Ex13.12: Page-653 (2011)
clc;clear;
K_B = 1.38e-23;....// Boltzmann constant , J/mol/K
```

```
T = 300;....// Room temperature, K
eps_0 = 8.85e-12;....// Electric permittivity of free space, F/m
N_A = 6.0e+23;     // Avogadro's number
n2 = N_A*1000;      // Number of molecules of non-polar substance in 1000 cc volume
p_0 = sqrt((9*K_B*T*eps_0*0.023)/n2);     // Dipole moment of polar molecules, C-m
printf("\nThe dipole moment of polar molecules = %5.3e C-m", p_0);

// Result
// The dipole moment of polar molecules = 3.555e-030 C-m
```

Chapter 14

Solid State Electronics

Scilab code Exa 14.1 Density of impurity atoms to N type and P type silicon

```
// Scilab code Ex14.1 : Pg:718(2011)
clc;clear;
e = 1.6e-019;    // Charge on an electron, C
mu_h = 0.048;   // Mobility of holes, metre square/volt-s
mu_e = 0.135;   // Mobility of electrons, metre square/volt-s
// For P-type semiconductor
rho_p = 1e-01;   // Resistivity of P type silicon, omh-m
// As rho_p = 1/(e*N_a*mu_h), solving for N_a
N_a = 1/(e*rho_p*mu_h); // Density of acceptor atoms, per metre cube
// For N-type semiconductor
rho_n = 1e-01;   // Resistivity of N type silicon, omh-m
// As rho_n = 1/(e*N_d*mu_h), solving for N_d
N_d = 1/(e*rho_n*mu_e); // Density of donor atoms, per metre cube
printf("\nDensity of acceptor atoms = %4.2e per
```

```
metre cube", N_a);
printf("\nDensity of donor atoms = %4.2e per metre cube", N_d);

// Result
// Density of acceptor atoms = 1.30e+21 per metre cube
// Density of donor atoms = 4.63e+20 per metre cube
```

Scilab code Exa 14.2 Electrical conductivity and resistivity of intrinsic germanium sample

```
// Scilab code Ex14.2 : Pg:718(2011)
clc;clear;
e = 1.6e-019;    // Charge on an electron, C
mu_e = 0.36;     // Mobility of an electron, metre square/V-s
mu_h = 0.17;     // Mobility of a hole, metre square/V-s
n_i = 2.5e+018; // Intrinsic concentration of Ge sample, per metre cube
sigma = e*n_i*(mu_h+mu_e);   // Electrical conductivity of Ge sample, mho per metre
rho = 1/sigma;   // Electrical resistivity of Ge, ohm-m
printf("\nThe electrical conductivity of intrinsic germanium sample = %5.3f mho/m", sigma);
printf("\nThe electrical resistivity of intrinsic germanium sample = %3.1f ohm-m", rho);

// Result
// The electrical conductivity of intrinsic germanium sample = 0.212 mho/m
// The electrical resistivity of intrinsic germanium sample = 4.7 ohm-m
```

Scilab code Exa 14.3 Electrical conductivity of undoped and doped silicon

```
// Scilab code Ex14.3 : Pg:719(2011)
clc;clear;
e = 1.6e-019;    // Charge on an electron , C
mu_e = 0.13;     // Mobility of an electron , metre
    square/V-s
mu_h = 0.05;     // Mobility of a hole , metre square/
    V-s
n_i = 1.5e+016; // Intrinsic concentration of Si ,
    per metre cube
// Pure Si
sigma = e*n_i*(mu_h+mu_e);   // Electrical
    conductivity of Si , mho per metre
// Pure Si doped with donor impurity
n_e = 5e+028/1e+09;       // Concentration of
    electrons , per metre cube
sigma_n = e*n_e*mu_e;    // Electrical conductivity
    of Si doped with donor impurity , mho per metre
// Pure Si doped with acceptor impurity
n_h = 5e+028/1e+09;       // Concentration of holes ,
    per metre cube
sigma_p = e*n_h*mu_h;    // Electrical conductivity
    of Si doped with acceptor impurity , mho per metre
printf("\nThe electrical conductivity of pure Si =
    %4.2e mho/m", sigma);
printf("\nThe electrical conductivity of Si doped
    with donor impurity = %4.2f mho/m", sigma_n);
printf("\nThe electrical conductivity of Si doped
    with acceptor impurity= %4.2f mho/m", sigma_p);

// Result
// The electrical conductivity of pure Si = 4.32e-04
```

```
      mho/m
// The electrical conductivity of Si doped with donor impurity = 1.04 mho/m
// The electrical conductivity of Si doped with acceptor impurity= 0.40 mho/m
```

Scilab code Exa 14.4 Shift in Fermi level due to change in density of donor atoms

```
// Scilab code Ex14.4 : Pg:720(2011)
clc;clear;
Nd = 1;      // For simplicity assume donor concentration to be unity, per metre cube
Nd_prime = 3*Nd;    // Thrice the donor concentration, per metre cube
dE_CF1 = 0.5;   // Energy difference between normal Fermi level and conduction level, eV
k_BT = 0.03;     // Thermal energy at room temperature, eV
// As Nd_prime/Nd = exp((dE_CF1 - dE_CF2))/k_BT, solving for dE_CF2
dE_CF2 = dE_CF1-k_BT*log(Nd_prime/Nd);   // Energy difference between new postion of Fermi level and conduction level, eV
printf("\nThe new postion of Fermi level when donor concentration is trebled = %5.3f eV", dE_CF2);

// Result
// The new postion of Fermi level when donor concentration is trebled = 0.467 eV
```

Scilab code Exa 14.5 Voltage required to cause a forward current density in pn junction diode

```
// Scilab code Ex14.5 :Pg:721(2011)
clc;clear;
e = 1.6e-019;    // Charge on an electron , C
T = 300;    // Room temperature , K
J0 = 300e-03;    // Saturation current density of the
    pn junction diode , A/metre square
J = 1e+05;  // Forward current density of pn
   junction diode , A/metre square
k_B = 1.38e-023;     // Boltzmann constant , J/K
eta = 1;    // Ideality factor for Ge diode
// As J = J0*exp(e*V/(eta*k_B*T)) , solving for V
V = eta*k_B*T/e*log(J/J0);    // Voltage required to
   cause a forward current density in pn junction
   diode , volt
printf("\nThe voltage required to cause a forward
   current density in pn junction diode = %5.3f V",
   V);

// Result
// The voltage required to cause a forward current
   density in pn junction diode = 0.329 V
```

Scilab code Exa 14.6 Applied voltage for forward current density

```
// Scilab code Ex14.6 : Pg:721(2011)
clc;clear;
e = 1.6e-019;    // Charge on an electron , C
T = 300;    // Room temperature , K
J0 = 200e-03;    // Saturation current density of the
    pn junction diode , A/metre square
J = 5e+04;  // Forward current density of pn
   junction diode , A/metre square
k_B = 1.38e-023;     // Boltzmann constant , J/K
eta = 1;    // Ideality factor for Ge diode
// As J = J0*exp(e*V/(eta*k_B*T)) , solving for V
```

```
V = eta*k_B*T/e*log(J/J0);    // Voltage required to
    cause a forward current density in pn junction
    diode, volt
printf("\nThe voltage required to cause a forward
    current density in pn junction diode = %5.3f V",
    V);

// Result
// The voltage required to cause a forward current
    density in pn junction diode = 0.322 V
```

Scilab code Exa 14.7 Forward voltage to increase the current density of Si diode

```
// Scilab code Ex14.7 : Pg:722(2011)
clc;clear;
e = 1.6e-019;    // Charge on an electron, C
T = 300;     // Room temperature, K
J0 = 300e-03;    // Saturation current density of the
      pn junction diode, A/metre square
J = 1e+05;   // Forward current density of pn
    junction diode, A/metre square
k_B = 1.38e-023;     // Boltzmann constant, J/K
eta = 2;     // Ideality factor for Ge diode
// As J = J0*exp(e*V/(eta*k_B*T)), solving for V
V = eta*k_B*T/e*log(J/J0);    // Voltage required to
    cause a forward current density in pn junction
    diode, volt
printf("\nThe voltage required to cause a forward
    current density in Si iode = %5.3f V", V);

// Result
// The voltage required to cause a forward current
    density in Si diode = 0.658 V
```

Scilab code Exa 14.8 Static and dynamic values of diode resistance

```
// Scilab code Ex14.8 : Pg:723(2011)
clc;clear;
I = 55e-03;      // Forward current through Si diode, A
V = 3;           // Forward bias across Si diode, V
eta = 2;    // Ideality factor for Si diode
R_dc = V/I;      // Static diode resistance, ohm
R_ac = 0.026*eta/I;     // Dynamic diode resistance, ohm
printf("\nThe static diode resistance = %4.1f ohm", R_dc);
printf("\nThe dynamic diode resistance = %5.3f ohm", R_ac);

// Result
// The static diode resistance = 54.5 ohm
// The dynamic diode resistance = 0.945 ohm
```

Scilab code Exa 14.9 Half wave rectifier parameters

```
// Scilab code Ex14.9 : Pg:723(2011)
clc;clear;
R_L = 1000; // Load resistance across HWR, ohm
V_rms = 200;     // Rms value of voltage supply, V
V0 = sqrt(2)*V_rms; // Peak value of voltage, V
I0 = V0/(R_L*1e-03);     // Peak value of current, mA
I_dc = I0/%pi;   // Average value of current, mA
I_rms = I0/2;    // Rms value of current, mA
V_dc = I_dc*R_L/1e+03;     // Dc output voltage, V
PIV = V0;    // Peak inverse voltage, V
```

```
printf("\nThe average value of current = %2d mA", I_dc);
printf("\nThe rms value of current = %5.1f mA", I_rms);
printf("\nThe dc output voltage = %2d V", V_dc/1);
printf("\nPIV = %5.1f V", PIV);


// Result
// The average value of current = 90 mA
// The rms value of current = 141.4 mA
// The dc output voltage = 90 V
// PIV = 282.8 V
```

Scilab code Exa 14.10 Full wave rectifier parameters

```
// Scilab code Ex14.10 : Pg:724(2011)
clc;clear;
R_L = 980; // Load resistance across FWR, ohm
R_F = 20;   // Internal resistance of two crystal diodes in FWR, ohm
V_rms = 50;    // Rms value of voltage supply, V
V0 = sqrt(2)*V_rms; // Peak value of voltage, V
I0 = V0/((R_L+R_F)*1e-03);    // Peak value of current, mA
I_dc = 2*I0/%pi;  // Average value of current, mA
I_rms = I0/sqrt(2);   // Rms value of current, mA
V_dc = I_dc*R_L/1e+03;    // Dc output voltage, V
eta = 81.2/(1+R_F/R_L);     // Rectification efficiency
PIV = 2*V0;   // Peak inverse voltage, V
printf("\nThe average value of current = %2d mA", I_dc);
printf("\nThe rms value of current = %2d mA", I_rms);
```

```
printf("\nThe dc output voltage = %4.1f V", V_dc/1);
printf("\nThe rectification efficiency = %4.1f
    percent", eta);
printf("\nPIV = %5.1f V", PIV);


// Result
// The average value of current = 45 mA
// The rms value of current = 50 mA
// The dc output voltage = 44.1 V
// The rectification efficiency = 79.6 percent
// PIV = 141.4 V
```

Scilab code Exa 14.11 Current gains in BJT

```
// Scilab code Ex14.11 : Pg:725(2011)
clc;clear;
delta_IC = 1e-03;    // Change in collector current ,
    A
delta_IB = 50e-06;    // Change in base current , A
bta = delta_IC/delta_IB;   // Base current
    amplification factor
alpha = bta/(1+bta);     // Emitter current
    amplification factor
printf("\nAlpha of BJT = %4.2f", alpha);
printf("\nBeta of BJT = %2d", bta);


// Result
// Alpha of BJT = 0.95
// Beta of BJT = 20
```

Scilab code Exa 14.12 Base current of BJT in CB mode

```
// Scilab code Ex14.12 : Pg:725(2011)
clc;clear;
I_E = 2;     // Emitter current, mA
alpha = 0.88;    // Emitter current amplification
    factor
I_C = alpha*I_E;     // Collector current, mA
I_B = I_E - I_C;         // Base current of BJT in CB
    mode, mA
printf("\nThe base current of BJT in CB mode = %4.2f
    mA", I_B);


// Result
// The base current of BJT in CB mode = 0.24 mA
```

Scilab code Exa 14.13 Current gain and base current of BJT in CB mode

```
// Scilab code Ex14.13 : Pg:725(2011)
clc;clear;
I_CBO = 12.5e-03;    // Reverse saturation current,
    mA
I_E = 2;     // Emitter current, mA
I_C = 1.97; // Collector current, mA
// As I_C = alpha*I_E+I_CBO, solving for alpha
alpha = (I_C - I_CBO)/I_E;   // Emitter current gain
I_B = I_E - I_C;     // Base current, mA
printf("\nThe emitter current gain = %5.3f", alpha);
printf("\nThe base current = %4.2f mA", I_B);


// Result
// The emitter current gain = 0.979
// The base current = 0.03 mA
```

Scilab code Exa 14.14 Current gain and leakage current of BJT in CE mode

```
// Scilab code Ex14.14 : Pg:726(2011)
clc;clear;
alpha = 0.98;    // Emitter current amplification
    factor
bta = alpha/(1-alpha);   // Emitter current
    amplification factor
I_CBO = 5e-06;   // Reverse saturation current , A
I_CEO = 1/(1-alpha)*I_CBO;   // Leakage current of
    BJT in CE mode, mA
printf("\nThe base current gain = %2g", bta);
printf("\nThe leakage current of BJT in CE mode = %4
    .2f mA", I_CEO/1e-03);


// Result
// The base current gain = 49
// The leakage current of BJT in CE mode = 0.25 mA
```

Scilab code Exa 14.15 Voltage and power gain of PNP transistor in CB mode

```
// Scilab code Ex14.15 : Pg:726(2011)
clc;clear;
R_i = 50;    // Dynamic input resistance of PNP
    transistor , ohm
R_L = 5e+03;     // Load resistance in collector
    circuit , ohm
alpha = 0.96;    // Emitter current amplification
    factor
```

```
A_v = alpha*R_L/R_i;    // Voltage gain
A_p = alpha*A_v;    // Power gain
printf("\nThe voltage gain = %2g", A_v);
printf("\nThe power gain = %2d", A_p);


// Result
// The voltage gain = 96
// The power gain = 92
```

Chapter 15

Digital Electronics

Scilab code Exa 15.1 Binary equivalent of decimal number

```
// Scilab code Ex15.1 : Pg:771(2011)
clc;clear;
function [bin]= decimal_binary(n) // Function to convert decimal to binary
    bin = 0;
    i = 1;
    while (n <> 0)
      rem = n-fix(n./2).*2;
      n = int(n/2);
      bin = bin + rem*i;
      i = i * 10;
    end
endfunction

n = 25;     // Initialize the decimal number
printf("Binary equivalent of %d = %d", n, decimal_binary(n));

// Result
// Binary equivalent of 25 = 11001
```

Scilab code Exa 15.2 Decimal equivalent of 6 bit binary number

```
// Scilab code Ex15.2 : Pg:771(2011)
clc;clear;
function [deci]= binary_decimal(ni) // Function to
   convert binary to decimal
    deci = 0;
    i = 0;
    while (ni <> 0)
      rem = ni-fix(ni./10).*10;
      ni = int(ni/10);
      deci = deci + rem*2.^i;
      i = i + 1;
    end
endfunction

function [decf]= binfrac_decifrac(nf) // Function to
    convert binary fraction to decimal fraction
    decf = 0;
    i = -1;
    while (i >= -3)
      nf = nf*10;
      rem = round(nf);
      nf = nf-rem;
      decf = decf + rem*2.^i;
      i = i - 1;
    end
endfunction

n = 101.101;     // Initialize the binary number
n_int = int(n);      // Extract the integral part
n_frac = n-n_int;    // Extract the fractional part
printf("Decimal equivalent of %7.3f = %5.3f", n,
   binary_decimal(n_int)+binfrac_decifrac(n_frac));
```

```
// Result
// Decimal equivalent of 101.101 = 5.625
```

Scilab code Exa 15.3 Decimal equivalent of octal number

```
// Scilab code Ex15.3 : Pg:772(2011)
clc;clear;
function [dec]= octal_decimal(n) // Function to
   convert binary to decimal
    dec = 0;
    i = 0;
    while (n <> 0)
      rem = n-fix(n./10).*10;
      n = int(n/10);
      dec = dec + rem*8.^i;
      i = i + 1;
    end
endfunction

n = 173;     // Initialize the octal number
printf("Decimal equivalent of %d = %d", n,
   octal_decimal(n));

// Result
// Decimal equivalent of 173 = 123
```

Scilab code Exa 15.4 Octal equivalent of decimal number

```
// Scilab code Ex15.4 : Pg:772(2011)
clc;clear;
function octal = decimal_octal(n)  // Function to
   convert decimal to octal
```

```
    i=1; octal = 0;
    while (n<>0)
        rem = n-fix(n./8).*8;
        octal = octal + rem*i;
        n = int(n/8);
        i = i*10;
    end
endfunction

n = 278;    // Initialize the octal number
printf("The octal equivalent of %d = %d", n,
   decimal_octal(n));

// Result
// The octal equivalent of 278 = 426
```

Scilab code Exa 15.5 Hexadecimal equivalent of binary numbers

```
// Scilab code Ex15.5 :Pg:772(2011)
clc;clear;
function hex = binary_hex(n) // Function to convert
   decimal to hexadecimal
    hex = emptystr();
    while (n <>0)
        rem = n-fix(n./10000).*10000;   // Division
           Algorithm
        if  rem == 0 then
            hex = hex+'0';
        elseif rem == 1 then
            hex = hex+'1';
        elseif rem == 10 then
            hex = hex+'2';
        elseif rem == 11 then
            hex = hex+'3';
        elseif rem == 100 then
```

```
            hex = hex+'4';
        elseif rem == 101 then
            hex = hex+'5';
        elseif rem == 110 then
            hex = hex+'6';
        elseif rem == 111 then
            hex = hex+'7';
        elseif rem == 1000 then
            hex = hex+'8';
        elseif rem == 1001 then
            hex = hex+'9';
        elseif rem == 1010 then
            hex=hex+'A';
        elseif rem == 1011 then
            hex=hex+'B';
        elseif rem == 1100 then
            hex=hex+'C';
        elseif rem == 1101 then
            hex=hex+'D';
        elseif rem == 1110 then
            hex=hex+'E';
        elseif rem == 1111 then
            hex=hex+'F';
        end // If statement ends
        n = int(n/10000);
    end // While loop ends
    hex = strrev(hex);   // Reverse string
endfunction

n = [10001100, 1011010111];    // Initialize the binary numbers
printf("\nThe hex equivalent of %d = %s", n(1), binary_hex(n(1)));
printf("\nThe hex equivalent of %d = %s", n(2), binary_hex(n(2)));

// Result
// The hex equivalent of 10001100 = 8C
```

```
// The hex equivalent of 1011010111 = 2D7
```

Scilab code Exa 15.6 Hexadecimal equivalent of decimal numbers

```
// Scilab code Ex15.6 : Pg:772(2011)
clc;clear;
function hex = decimal_hex(n) // Function to convert
    decimal to hexadecimal
    hex = emptystr();
    while (n <>0)
        rem = n-fix(n./16).*16;
        if rem == 10 then
            hex(i)=hex+'A';
        elseif rem == 11 then
            hex=hex+'B';
        elseif rem == 12 then
            hex=hex+'C';
        elseif rem == 13 then
            hex=hex+'D';
        elseif rem == 14 then
            hex=hex+'E';
        elseif rem == 15 then
            hex=hex+'F';
        else
            hex=hex+string(rem);
        end
        n = int(n/16);
    end
    hex = strrev(hex);    // Reverse string
endfunction

n = 72905;     // Initialize the binary numbers
printf("\nThe hex equivalent of %d = %s", n,
   decimal_hex(n));

```

```
// Result
// The hex equivalent of 72905 = 11CC9
```

Scilab code Exa 15.7 Addition of two binary numbers

```
// Scilab code Ex15.7 : Pg:773(2011)
clc;clear;
function [bini]= decimal_binary(ni) // Function to convert decimal to binary
    bini = 0;
    i = 1;
    while (ni <> 0)
      rem = ni-fix(ni./2).*2;
      ni = int(ni/2);
      bini = bini + rem*i;
      i = i * 10;
    end
endfunction

function [deci]= binary_decimal(ni) // Function to convert binary to decimal
    deci = 0;
    i = 0;
    while (ni <> 0)
      rem = ni-fix(ni./10).*10;
      ni = int(ni/10);
      deci = deci + rem*2.^i;
      i = i + 1;
    end
endfunction

num1 = 11101;    // Initialize the first binary number
num2 = 10111;    // Initialize the second binary
```

```
    number

printf("%6d + %6d = %7d", num1, num2, decimal_binary
    (binary_decimal(num1)+binary_decimal(num2)));

// Result
// 11101 +  10111 =  110100
```

Scilab code Exa 15.8 Addition of two binary numbers with fractions

```
// Scilab code Ex15.8 : Pg:773(2011)
clc;clear;
function [bini]= decimal_binary(ni) // Function to
    convert decimal to binary
    bini = 0;
    i = 1;
    while (ni <> 0)
      rem = ni-fix(ni./2).*2;
      ni = int(ni/2);
      bini = bini + rem*i;
      i = i * 10;
    end
endfunction

function [binf]= decifrac_binfrac(nf) // Function to
     convert binary fraction to decimal fraction
    binf = 0; i = 0.1;
    while (nf <> 0)
      nf = nf*2;
      rem = int(nf);
      nf = nf-rem;
      binf = binf + rem*i;
      i = i/10;
    end
endfunction
```

```

function [deci]= binary_decimal(ni) // Function to convert binary to decimal
    deci = 0;
    i = 0;
    while (ni <> 0)
      rem = ni-fix(ni./10).*10;
      ni = int(ni/10);
      deci = deci + rem*2.^i;
      i = i + 1;
    end
endfunction

function [decf]= binfrac_decifrac(nf) // Function to convert binary fraction to decimal fraction
    decf = 0;
    i = -1;
    while (i >= -3)
      nf = nf*10;
      rem = round(nf);
      nf = nf-rem;
      decf = decf + rem*2.^i;
      i = i - 1;
    end
endfunction

bin1 = 1011.11;     // Initialize the first binary binber
bin2 = 1011.01;     // Initialize the second binary binber
bin1_int = int(bin1);      // Extract the integral part for first
bin1_frac = bin1-bin1_int;    // Extract the fractional part for second
bin2_int = int(bin2);      // Extract the integral part for first
bin2_frac = bin2-bin2_int;    // Extract the fractional part for second
```

```
dec1 = binary_decimal(bin1_int)+binfrac_decifrac(
    bin1_frac);
dec2 = binary_decimal(bin2_int)+binfrac_decifrac(
    bin2_frac);
dec = dec1+dec2;
dec_int = int(dec);
dec_frac = dec-dec_int;
printf("%7.2f + %7.2f = %8.2f", bin1, bin2,
    decimal_binary(dec_int)+decifrac_binfrac(dec_frac
    ));

// Result
// 1011.11 + 1011.01 = 10111.00
```

Scilab code Exa 15.9 Subtraction of two binary numbers

```
// Scilab code Ex15.9 : Pg:773(2011)
clc;clear;
function [bini]= decimal_binary(ni) // Function to
    convert decimal to binary
  bini = 0;
  i = 1;
  while (ni <> 0)
    rem = ni-fix(ni./2).*2;
    ni = int(ni/2);
    bini = bini + rem*i;
    i = i * 10;
  end
endfunction

function [deci]= binary_decimal(ni) // Function to
    convert binary to decimal
  deci = 0;
  i = 0;
  while (ni <> 0)
```

```
      rem = ni-fix(ni./10).*10;
      ni = int(ni/10);
      deci = deci + rem*2.^i;
      i = i + 1;
    end
endfunction

num1 = 1001;     // Initialize the first binary number
num2 = 0111;     // Initialize the second binary number

printf("%4d - 0%3d = 00%2d", num1, num2, decimal_binary(binary_decimal(num1)-binary_decimal(num2)));

// Result
// 1001 - 0111 = 0010
```

Scilab code Exa 15.10 Multiplication of two binary numbers

```
// Scilab code Ex15.10 :Pg:773(2008)
clc;clear;
function [bini]= decimal_binary(ni) // Function to convert decimal to binary
    bini = 0;
    i = 1;
    while (ni <> 0)
      rem = ni-fix(ni./2).*2;
      ni = int(ni/2);
      bini = bini + rem*i;
      i = i * 10;
    end
endfunction

```

```
function [deci]= binary_decimal(ni) // Function to convert binary to decimal
    deci = 0;
    i = 0;
    while (ni <> 0)
      rem = ni-fix(ni./10).*10;
      ni = int(ni/10);
      deci = deci + rem*2.^i;
      i = i + 1;
    end
endfunction

function binp = bin_product(op1, op2)
    binp = decimal_binary(binary_decimal(op1)*binary_decimal(op2));
endfunction

mul1 = 1101;     // Initialize the first binary multiplicand
mul2 = 1100;     // Initialize the second binary multiplicand
product = bin_product(mul1, mul2);

printf("%4d X %4d = %8d", mul1, mul2, product);

// Result
//  1101 X 1100 = 10011100
```

Scilab code Exa 15.11 Binary division of two numbers

```
// Scilab code Ex15.11 : Pg:774(2008)
clc;clear;
function [bini]= decimal_binary(ni) // Function to convert decimal to binary
    bini = 0;
```

```
    i = 1;
    while (ni <> 0)
        rem = ni-fix(ni./2).*2;
        ni = int(ni/2);
        bini = bini + rem*i;
        i = i * 10;
    end
endfunction

function [deci]= binary_decimal(ni) // Function to convert binary to decimal
    deci = 0;
    i = 0;
    while (ni <> 0)
        rem = ni-fix(ni./10).*10;
        ni = int(ni/10);
        deci = deci + rem*2.^i;
        i = i + 1;
    end
endfunction

function binp = bin_division(op1, op2)
    binp = decimal_binary(binary_decimal(op1)/binary_decimal(op2));
endfunction

dividend = 11001 ;    // Initialize the first binary multiplicand
divisor = 101;    // Initialize the second binary multiplicand
product = bin_division(dividend, divisor);

printf("%5d divided by %3d gives %3d", dividend, divisor, product);

// Result
// 11001 divided by 101 gives 101
```

Chapter 16

Crystal Physics

Scilab code Exa 16.1 Miller indices of the crystal plane

```
// Scilab Code Ex16.1 : Page-820 (2011)
clc; clear;
p = 1; q = 2; r = 3; // Coefficients of intercepts along three axes
p_inv = 1/p;         // Reciprocate the first coefficient
q_inv = 1/q;         // Reciprocate the second coefficient
r_inv = 1/r;         // Reciprocate the third coefficient
mul_fact = double(lcm(int32([p,q,r]))); // Find l.c.m. of m,n and p
m1 = p_inv*mul_fact;    // Clear the first fraction
m2 = q_inv*mul_fact;    // Clear the second fraction
m3 = r_inv*mul_fact;    // Clear the third fraction
printf("\nThe required miller indices are : (%d %d %d) ", m1,m2,m3);

// Result
// The required miller indices are : (6 3 2)
```

Scilab code Exa 16.2 Miller indices of the lattice plane

```
// Scilab Code Ex16.2 : Page-820 (2011)
clc; clear;
p = 2; q = 3; r = -4; // Coefficients of intercepts
    along three axes
p_inv = 1/p;          // Reciprocate the first
    coefficient
q_inv = 1/q;          // Reciprocate the second
    coefficient
r_inv = 1/r;          // Reciprocate the third
    coefficient
mul_fact = double(lcm(int32([p,q,abs(r)]))); // Find
     l.c.m. of m,n and p
m1 = p_inv*mul_fact;    // Clear the first fraction
m2 = q_inv*mul_fact;    // Clear the second fraction
m3 = r_inv*mul_fact;    // Clear the third fraction
printf("\nThe miller indices of laticce plane are :
    (%d %d %d) ", m1,m2,m3);

// Result
// The miller indices of laticce plane are : (6 4
    -3)
```

Scilab code Exa 16.3 Miller indices of the set of parallel planes

```
// Scilab Code Ex16.3 : Page-821 (2011)
clc; clear;
p = 3; q = 4; r = %inf; // Coefficients of
    intercepts along three axes
p_inv = 1/p;          // Reciprocate the first
    coefficient
```

```
q_inv = 1/q;          // Reciprocate the second coefficient
r_inv = 1/r;          // Reciprocate the third coefficient
mul_fact = double(lcm(int32([p,q]))); // Find l.c.m. of m,n and p
m1 = p_inv*mul_fact;     // Clear the first fraction
m2 = q_inv*mul_fact;     // Clear the second fraction
m3 = r_inv*mul_fact;     // Clear the third fraction
printf("\nThe miller indices of the given planes are : (%d %d %d) ", m1,m2,m3);

// Result
// The miller indices of the given planes are : (4 3 0)
```

Scilab code Exa 16.4 Length of the intercepts on Y and Z axes

```
// Scilab Code Ex16.4 : Page-822 (2011)
clc; clear;
p = 1.2;   // First coefficient of intercept along X-axis, angstrom
a = 1.2, b = 1.8, c = 2.0;     // Lattice parameters along three axes, angstrom
h = 2, k = 3, l = 1;     // Miller indices of lattice plane
// As p:q:r = a/h:b/k:c/l, solving for q and r
q = p*(b/k)/(a/h); // Second coefficient of intercept along X-axis, angstrom
r = p*(c/l)/(a/h); // Third coefficient of intercept along X-axis, angstrom
printf("\nThe lengths of the intercepts on Y and Z axes are %3.1f angstrom and %3.1f angstrom respectively", q, r);

```

```
// Result
// The lengths of the intercepts on Y and Z axes are
   1.2 angstrom and 4.0 angstrom respectively
```

Scilab code Exa 16.5 Lattice constant for NaCl crystal

```
// Scilab Code Ex16.5 : Page-822 (2011)
clc; clear;
M = 58.5;          // Molecular weight of NaCl, g-mole
rho = 2.198e+03;      // Density of Nacl, kg per metre
   cube
n = 4;     // No. of atoms per unit cell for an fcc
   lattice of NaCl crystal
NA = 6.023D+26;     // Avogadro's No., atoms/k-mol
// Volume of the unit cell is given by
// a^3 = M*n/(N*d)
// Solving for a
a = (n*M/(rho*NA))^(1/3);     // Lattice constant of
   unit cell of NaCl
printf("\nLattice constant for the NaCl crystal = %4
   .2f angstrom", a/1e-010);

// Result
// Lattice constant for the NaCl crystal = 5.61
   angstrom
```

Scilab code Exa 16.6 Lattice constant for KBr crystal

```
// Scilab Code Ex16.6 :Page-823 (2011)
clc; clear;
M = 119;          // Molecular weight of KBr, g-mole
rho = 2.7;     // Density of KBr, g per cm-cube
```

```
n = 4;     // No. of atoms per unit cell for an fcc lattice of KBr crystal
NA = 6.023D+23;     // Avogadro's No., atoms/mol
// Volume of the unit cell is given by
// a^3 = M*n/(N*d)
// Solving for a
a = (n*M/(rho*NA))^(1/3);     // Lattice constant of unit cell of KBr
printf("\nLattice constant for the KBr crystal = %4.2f angstrom", a/1e-008);

// Result
// Lattice constant for the KBr crystal = 6.64 angstrom
```

Scilab code Exa 16.7 Lattice constant for Cu and distance between the two nearest Cu atoms

```
// Scilab Code Ex16.7 : Page-823 (2011)
clc; clear;
M = 63.5;          // Molecular weight of Cu, g-mole
rho = 8.96;     // Density of Cu, g per cm-cube
n = 4;     // No. of atoms per unit cell for an fcc lattice of Cu
NA = 6.023D+23;     // Avogadro's No., atoms/mol
// Volume of the unit cell is given by
// a^3 = M*n/(N*d)
// Solving for a
a = (n*M/(rho*NA))^(1/3);     // Lattice constant of unit cell of Cu
d = a/sqrt(2);     // Distance between the two nearest Cu atoms, angstrom
printf("\nLattice constant for the Cu crystal = %4.2f angstrom", a/1e-008);
printf("\nThe distance between the two nearest Cu
```

```
atoms = %4.2f angstrom", d/1e-008);

// Result
// Lattice constant for the Cu crystal = 3.61 angstrom
// The distance between the two nearest Cu atoms = 2.55 angstrom
```

Scilab code Exa 16.8 Inter planar spacing for lattice planes

```
// Scilab Code Ex16.8 : Page-824 (2011)
clc; clear;
a = 1;      // For simplicity assume lattice parameter of cubic crystal to be unity, unit
// For (011) planes
h = 0; k = 1; l = 1; // Miller Indices for planes in a cubic crystal
d_011 = a/(h^2+k^2+l^2)^(1/2);   // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (011) planes = a/sqrt(%d)", 1/d_011^2);

// For (101) planes
h = 1; k = 0; l = 1; // Miller Indices for planes in a cubic crystal
d_101 = a/(h^2+k^2+l^2)^(1/2);   // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (101) planes = a/sqrt(%d)", 1/d_101^2);

// For (112) planes
h = 1; k = 1; l = 2; // Miller Indices for planes in a cubic crystal
```

```
d_112 = a/(h^2+k^2+l^2)^(1/2);   // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (112) planes = a/sqrt(%d)", 1/d_112^2);

// Result
// The interplanar spacing between consecutive (011) planes = a/sqrt(2)
// The interplanar spacing between consecutive (101) planes = a/sqrt(2)
// The interplanar spacing between consecutive (112) planes = a/sqrt(5)
```

Scilab code Exa 16.9 Interplanar spacing in cubic crystal

```
// Scilab Code Ex16.9 : Page-824 (2011)
clc; clear;
a = 4.2e-010;     // Lattice parameter of cubic crystal, m
h = 3; k = 2; l = 1; // Miller Indices for planes in a cubic crystal
d_321 = a/(h^2+k^2+l^2)^(1/2);   // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (321) planes = %4.2f angstrom", d_321/1e-010);

// Result
// The interplanar spacing between consecutive (321) planes = 1.12 angstrom
```

Scilab code Exa 16.10 Interplanar spacing in tetragonal crystal lattice

```
// Scilab Code Ex16.10 : Page-824 (2011)
clc; clear;
a = 2.5, b = 2.5, c = 1.8;     // Lattice parameter
    of tetragonal crystal, angstrom
h = 1; k = 1; l = 1; // Miller Indices for planes in
     a tetragonal crystal
d_hkl = 1/sqrt((h/a)^2+(k/b)^2+(l/c)^2);  // The
    interplanar spacing for tetragonal crystals, m
printf("\nThe interplanar spacing between
    consecutive (111) planes = %4.2f angstrom", d_hkl
    );

// Result
// The interplanar spacing between consecutive (111)
     planes = 1.26 angstrom
```

Chapter 17

Nuclear Physics

Scilab code Exa 17.1 Binding energy per nucleon for the deutron

```
// Scilab code Ex17.1 : Pg:888 (2011)
clc;clear;
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
m_n = 1.675e-027;    // Mass of the neutron, kg
m_p = 1.672e-027;    // Mass of the proton, kg
M_D = 3.343e-027;    // Mass of the deutron, kg
c = 3e+08;  // Speed of light, m/s
delta_m = m_n + m_p - M_D;   // Mass defect in the
    formation of deuterium, kg
BE = delta_m*c^2;    // Binding energy of the
    deuterium, J
BE_bar = BE/2;        // Binding energy per nucleon of
     deuterium, J
printf("\nBinding energy per nucleon for the deutron
     = %5.3f MeV", BE_bar/(e*1e+06));

// Result
// Binding energy per nucleon for the deutron =
    1.125 MeV
```

Scilab code Exa 17.2 Binding energy for an alpha particle

```
// Scilab code Ex17.2 : Pg:889 (2011)
clc;clear;
amu = 931.5;    // Energy equivalent of 1 amu, MeV
m_n = 1.008665;    // Mass of the neutron, amu
m_p = 1.007825;    // Mass of the proton, amu
M_He = 4.002870;    // Mass of the heluim nucleus,
    amu
c = 3e+08;  // Speed of light, m/s
BE = (2*m_n+2*m_p - M_He)*amu;  // Binding energy
    for the alpha particle, MeV
printf("\nThe binding energy for the alpha particle
    = %2d MeV", BE);

// Result
// The binding energy for the alpha particle = 28
    MeV
```

Scilab code Exa 17.3 Weizsacker formula for stability of nuclei

```
// Scilab code Ex17.3 :Pg:889 (2011)
clc;clear;
A = 1;  // For simplicity assume mass number to be
    unity
nucleus = cell(4,3);
nucleus(1,1).entries = 'He';
nucleus(1,2).entries = 2;
nucleus(1,3).entries = 6;
nucleus(2,1).entries = 'Be';
nucleus(2,2).entries = 4;
nucleus(2,3).entries = 6;
```

```
nucleus(3,1).entries = 'Li';
nucleus(3,2).entries = 3;
nucleus(3,3).entries = 6;
a_c = 0.7053;    // Asymmetry energy constant, MeV
a_a = 23.702;    // Coulomb energy constant, MeV
Z = A/(2+a_c/(2*a_a)*A^(2/3));
for i = 1:1:3
    if abs(nucleus(i,2).entries/nucleus(i,3).entries
        - Z) < 0.005 then
        printf("\n%s(%d,%d) is more stable than
            other two nuclei", nucleus(i,1).entries,
            nucleus(i,2).entries, nucleus(i,3).
            entries);
    end
end

// Result
// Li(3,6) is more stable than other two nuclei
```

Scilab code Exa 17.4 Nuclei stability

```
// Scilab code Ex17.4 : Pg:890 (2011)
clc;clear;
nucleus = cell(4,3);
// For Li nuclides
nucleus(1,1).entries = 'Li';
nucleus(1,2).entries = 3;
nucleus(1,3).entries = 7;
nucleus(2,1).entries = 'Li';
nucleus(2,2).entries = 3;
nucleus(2,3).entries = 8;
// For Be nuclides
nucleus(3,1).entries = 'Be';
nucleus(3,2).entries = 4;
nucleus(3,3).entries = 9;
```

```
nucleus(4,1).entries = 'Be';
nucleus(4,2).entries = 4;
nucleus(4,3).entries = 10;
a_c = 0.7053;   // Asymmetry energy constant, MeV
a_a = 23.702;   // Coulomb energy constant, MeV
for i = 1:1:4
    Z = nucleus(i,3).entries/(2+a_c/(2*a_a)*nucleus(
       i,3).entries^(2/3));
    if abs(Z-int(Z)) < 0.5 then
        printf("\n%s(%d,%d) is more stable", nucleus
           (i,1).entries, nucleus(i,2).entries,
           nucleus(i,3).entries);
    end
end

// Result
// Li(3,7) is more stable
// Be(4,9) is more stable
```

Scilab code Exa 17.5 Exothermicity and endothermicity of nuclear reactions

```
// Scilab code Ex17.5 : Pg:891 (2011)
clc;clear;
c = 1;  // For simplicity assume speed of light to
   be unity, unit
nucleus = cell(4,4);
// For first reaction
nucleus(1,1).entries = 'N';
nucleus(1,2).entries = 7;
nucleus(1,3).entries = 14;
nucleus(1,4).entries = 14.00753;
nucleus(2,1).entries = 'He';
nucleus(2,2).entries = 2;
nucleus(2,3).entries = 4;
```

```
nucleus(2,4).entries = 4.00206;
nucleus(3,1).entries = 'O';
nucleus(3,2).entries = 8;
nucleus(3,3).entries = 17;
nucleus(3,4).entries = 17.00450;
nucleus(4,1).entries = 'H';
nucleus(4,2).entries = 1;
nucleus(4,3).entries = 1;
nucleus(4,4).entries = 1.00814;
Q = (nucleus(1,4).entries + nucleus(2,4).entries)*c^2 - (nucleus(3,4).entries + nucleus(4,4).entries)*c^2;
if Q < 0 then
    T_state = "endothermic";
   elseif Q > 0
    T_state = "exothermic";
end
printf("\nThe reaction");
printf("\n%s(%d,%d) + %s(%d,%d) --> %s(%d,%d) + %s(%d,%d) is %s", nucleus(1,1).entries, nucleus(1,2).entries, nucleus(1,3).entries, nucleus(2,1).entries, nucleus(2,2).entries, nucleus(2,3).entries, nucleus(3,1).entries, nucleus(3,2).entries, nucleus(3,3).entries, nucleus(4,1).entries, nucleus(4,2).entries, nucleus(4,3).entries, T_state);
// For second reaction
nucleus(1,1).entries = 'Li';
nucleus(1,2).entries = 3;
nucleus(1,3).entries = 7;
nucleus(1,4).entries = 7.01822;
nucleus(2,1).entries = 'H';
nucleus(2,2).entries = 1;
nucleus(2,3).entries = 1;
nucleus(2,4).entries = 1.00814;
nucleus(3,1).entries = 'He';
nucleus(3,2).entries = 2;
nucleus(3,3).entries = 4;
```

```
nucleus(3,4).entries = 4.00206;
Q = (nucleus(1,4).entries + nucleus(2,4).entries)*c^2 - (nucleus(3,4).entries + nucleus(3,4).entries)*c^2;
if Q < 0 then
    T_state = "endothermic";
   elseif Q > 0
    T_state = "exothermic";
end
printf("\nThe reaction");
printf("\n%s(%d,%d) + %s(%d,%d) --> %s(%d,%d) + %s(%d,%d) is %s", nucleus(1,1).entries, nucleus(1,2).entries, nucleus(1,3).entries, nucleus(2,1).entries, nucleus(2,2).entries, nucleus(2,3).entries, nucleus(3,1).entries, nucleus(3,2).entries, nucleus(3,3).entries, nucleus(4,1).entries, nucleus(4,2).entries, nucleus(4,3).entries, T_state);

// Result
//
// The reaction
// N(7,14) + He(2,4) --> O(8,17) + H(1,1) is endothermic
// The reaction
// Li(3,7) + H(1,1) --> He(2,4) + H(1,1) is exothermic
```

Scilab code Exa 17.6 Q value of the formation of P30 in ground state

```
// Scilab code Ex17.6 : Pg:891 (2011)
clc;clear;
nucleus = cell(4,3);
nucleus(1,1).entries = 'Si';
nucleus(1,2).entries = 14;
```

```
nucleus(1,3).entries = 29;
nucleus(2,1).entries = 'H';
nucleus(2,2).entries = 1;
nucleus(2,3).entries = 2;
nucleus(3,1).entries = 'P';
nucleus(3,2).entries = 15;
nucleus(3,3).entries = 30;
nucleus(4,1).entries = 'n';
nucleus(4,2).entries = 0;
nucleus(4,3).entries = 1;
Q = 2*23.834-44.359;    // Q-value of the reaction, MeV
printf("\nThe reaction");
printf("\n%s(%d,%d) + %s(%d,%d) --> %s(%d,%d) + %s(%d,%d)", nucleus(1,1).entries, nucleus(1,2).entries, nucleus(1,3).entries, nucleus(2,1).entries, nucleus(2,2).entries, nucleus(2,3).entries, nucleus(3,1).entries, nucleus(3,2).entries, nucleus(3,3).entries, nucleus(4,1).entries, nucleus(4,2).entries, nucleus(4,3).entries);
printf("\nhas the Q-value : %5.3f MeV", Q);

// Result
// The reaction
// Si(14,29) + H(1,2) --> P(15,30) + n(0,1)
// has the Q-value : 3.309 MeV
```

Scilab code Exa 17.7 Threshold energy required to initiate the reaction

```
// Scilab code Ex17.7 : Pg:892 (2011)
clc;clear;
amu = 931.5;    // Energy equivalent of 1 amu, MeV
nucleus = cell(4,3);
nucleus(1,1).entries = 'P';
```

```
nucleus(1,2).entries = 15;
nucleus(1,3).entries = 31;
nucleus(1,4).entries = 30.98356;
nucleus(2,1).entries = 'n';
nucleus(2,2).entries = 0;
nucleus(2,3).entries = 1;
nucleus(2,4).entries = 1.00898;
nucleus(3,1).entries = 'Si';
nucleus(3,2).entries = 14;
nucleus(3,3).entries = 31;
nucleus(3,4).entries = 30.98515;
nucleus(4,1).entries = 'p';
nucleus(4,2).entries = 1;
nucleus(4,3).entries = 1;
nucleus(4,4).entries = 1.00814;
Q = ((nucleus(1,4).entries + nucleus(2,4).entries)-(
    nucleus(3,4).entries + nucleus(4,4).entries))*amu
    ;    // Q-value of the reaction, MeV
E_th = -1*Q*(nucleus(1,4).entries+nucleus(2,4).
    entries)/nucleus(1,4).entries;
printf("\nThe threshold energy required to initiate
    the reaction");
printf("\n\t%s(%d,%d) + %s(%d,%d) --> %s(%d,%d) + %s
    (%d,%d)", nucleus(1,1).entries, nucleus(1,2).
    entries, nucleus(1,3).entries, nucleus(2,1).
    entries, nucleus(2,2).entries, nucleus(2,3).
    entries, nucleus(3,1).entries, nucleus(3,2).
    entries, nucleus(3,3).entries, nucleus(4,1).
    entries, nucleus(4,2).entries, nucleus(4,3).
    entries);
printf("\nis %5.3f MeV", E_th);

// Result
// The threshold energy required to initiate the
    reaction
//     P(15,31) + n(0,1) --> Si(14,31) + p(1,1)
// is 0.721 MeV
```

Scilab code Exa 17.8 Kinetic energy of emitted protons and threshold energy

```
// Scilab code Ex17.8 : Pg:892 (2011)
clc;clear;
amu = 931.5;     // Energy equivalent of 1 amu, MeV
nucleus = cell(4,3);
nucleus(1,1).entries = 'F';
nucleus(1,2).entries = 9;
nucleus(1,3).entries = 19;
M_P = 19.0457;     // Mass of product nucleus, amu
nucleus(2,1).entries = 'n';
nucleus(2,2).entries = 0;
nucleus(2,3).entries = 1;
m_i = 1.0087;     // Mass of incident particle, amu
nucleus(3,1).entries = 'O';
nucleus(3,2).entries = 8;
nucleus(3,3).entries = 19;
nucleus(4,1).entries = 'H';
nucleus(4,2).entries = 1;
nucleus(4,3).entries = 1;
m_e = 1.00728;     // Mass of emitted particle, amu
K_i = 15;     // Kinetic energy of incident neutrons,
    MeV
Q = -7.6342;     // Q-value of the reaction, MeV
K_e = (Q*M_P-(m_i-M_P)*K_i)/(m_e+M_P);     // Kinetic
    energy of emitted photon, MeV
E_th = -1*Q*(M_P+m_i)/M_P;     // Threshold energy
    required to initiate the reaction, MeV
printf("\nThe kinetic energy of emitted photon = %5
    .3f MeV", K_e);
printf("\nThe threshold energy required to initiate
    the reaction");
printf("\n\t%s(%d,%d) + %s(%d,%d) --> %s(%d,%d) + %s
```

```
(%d,%d)", nucleus(1,1).entries, nucleus(1,2).entries, nucleus(1,3).entries, nucleus(2,1).entries, nucleus(2,2).entries, nucleus(2,3).entries, nucleus(3,1).entries, nucleus(3,2).entries, nucleus(3,3).entries, nucleus(4,1).entries, nucleus(4,2).entries, nucleus(4,3).entries);
printf("\nis %5.3f MeV", E_th);

// Result
// The kinetic energy of emitted photon = 6.241 MeV
// The threshold energy required to initiate the reaction
//  F(9,19) + n(0,1) --> O(8,19) + H(1,1)
// is 8.039 MeV
```

Scilab code Exa 17.9 Energy released by the fission of 1 kg of U235

```
// Scilab code Ex17.9 : Pg:893 (2011)
clc;clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
N_A = 6.023e+023;    // Avogadro's number
E_f = 200*1e+06*e;    // Energy released per fission, J
E_mol = E_f*N_A;    // Energy released by one mole of U235, J
E = E_mol*1000/235;    // Energy released by the fission of 1 kg of U235, J
printf("\nThe Energy released by the fission of 1 kg of U235 = %4.2e kWh", E/(1000*3600));

// Result
// The Energy released by the fission of 1 kg of U235 = 2.28e+007 kWh
```

Scilab code Exa 17.10 Rate of fission of U235

```
// Scilab code Ex17.10 : Pg:894 (2011)
clc;clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
E = 3.2e+07;    // Energy released per second by the
    reactor , J
E_f = 200*1e+06*e;    // Energy released per fission
   , J
N = E/E_f;    // Number of fissions per second of
   U235, per second
printf("\nThe number of U235 atoms undergoing
   fissions per second = %1.0e", N);

// Result
// The number of U235 atoms undergoing fissions per
   second = 1e+018
```

Scilab code Exa 17.11 Rate of fission and energy released in the complete fissioning of U235

```
// Scilab code Ex17.11 : Pg:894 (2011)
clc;clear;
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
N_A = 6.023e+026;    // Avogadro's number, per kmol
P = 2;    // Power produced by the fission of U235,
   watt
E_f = 200*1e+06*e;    // Energy released per fission
   , J
FR = P/E_f;    // Fission rate of U235, fission/sec
N = 0.5/235*N_A;    // Number of U235 nuclei in 0.5
   kg of U235
```

```
E = 200*N;     // Energy released in the complete fissioning of 0.5 kg of U235, MeV
printf("\nThe fission rate of U235 = %4.2e fissions/sec", FR);
printf("\nThe energy released in the complete fissioning of 0.5 kg of U235 = %1.0e kcal", E*1e+06*e/(1000*4.186));

// Result
// The fission rate of U235 = 6.25e+010 fissions/sec
// The energy released in the complete fissioning of 0.5 kg of U235 = 1e+010 kcal
```

Scilab code Exa 17.12 Energy and oscillator frequency of some positively charged particles accelerating in a cyclotron

```
// Scilab code Ex17.12 : Pg:894 (2011)
clc;clear;
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
R_max = 0.6;     // Radius of two dees of the cyclotron, m
B = 1.6;     // Strength of pole pieces of the cyclotron, tesla
// For proton
m = 1.67e-027;     // Mass of the proton, kg
q = 1.6e-019;     // Charge on a proton, C
E = 1/2*q^2*R_max^2*B^2/(m*e*1e+06);     // Energy of the proton, MeV
f_proton = q*B/(2*%pi*m*1e+06);     // Cyclotron oscillator frequency for the proton, MHz
printf("\nEnergy of the proton = %5.2f MeV", E);
printf("\nCyclotron frequency for proton = %5.2f MHz", f_proton);
// For deuteron
m = 2*1.67e-027;     // Mass of the deuteron, kg
```

```
q = 1.6e-019;     // Charge on a deuteron, C
E = 1/2*q^2*R_max^2*B^2/(m*e*1e+06);     // Energy of
     the deuteron, MeV
f_deuteron = q*B/(2*%pi*m*1e+06);     // Cyclotron
    oscillator frequency for the deuteron, MHz
printf("\nEnergy of the deuteron = %5.2f MeV", E);
printf("\nCyclotron frequency for deuteron = %5.2f
    MHz", f_deuteron);
// For alpha-particle
m = 4*1.67e-027;     // Mass of the alpha-particle,
    kg
q = 2*1.6e-019;     // Charge on a alpha-particle, C
E = 1/2*q^2*R_max^2*B^2/(m*e*1e+06);     // Energy of
     the deuteron, MeV
f_alpha = q*B/(2*%pi*m*1e+06);     // Cyclotron
    oscillator frequency for the alpha-particle, MHz
printf("\nEnergy of the alpha-particle = %5.2f MeV",
     E);
printf("\nCyclotron frequency for alpha-particle =
    %5.2f MHz", f_alpha);

// Result
// Energy of the proton = 44.15 MeV
// Cyclotron frequency for proton = 24.40 MHz
// Energy of the deuteron = 22.07 MeV
// Cyclotron frequency for deuteron = 12.20 MHz
// Energy of the alpha-particle = 44.15 MeV
// Cyclotron frequency for alpha-particle = 12.20
    MHz
```

Scilab code Exa 17.13 Energy of the protons issuing out of the cyclotron

```
// Scilab code Ex17.13 : Pg:895 (2011)
clc;clear;
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
```

```
R_max = 0.75;     // Radius of two dees of the
    cyclotron , m
f = 15e+06;     // Frequency of alternating potential
    , Hz
m = 1.67e-027;     // Mass of the proton , kg
// As E = 1/2*q^2*R_max^2*B^2/(m*e) and f = q*B/(2*
    %pi*m) , solving for E
E = 2*%pi^2*m*f^2*R_max^2/(e*1e+06);
disp(E)
printf("\nEnergy of the protons issuing out of the
    cyclotron = %6.4f MeV", E);

// Result
// Energy of the protons issuing out of the
    cyclotron = 26.0754 MeV
```

Scilab code Exa 17.14 Energy gained per turn and maximum energy of the electron in a betatron

```
// Scilab code Ex17.14 : Pg:896 (2011)
clc;clear;
e = 1.6e-019;     // Charge on an electron , C
c = 3e+08;          // Speed of light , m/s
B_orbit = 0.5;     // Magnetic field at the orbit of
    the betatron , T
f = 60;     // Operating frequency of the betatron ,
    Hz
omega = 2*%pi*f;     // Angular frequency of
    operation , rad/s
r = 1.6/2;     // Radius of stable orbit , m
K_av = 4*omega*e*r^2*B_orbit/1.6e-019;     // Average
     energy gained by the electron per turn , eV
K_max = c*e*r*B_orbit/1.6e-019;     // Maximum energy
     gained by the eectron , eV
printf("\nThe average energy gained by the electron
```

```
    per turn = %5.1f eV", K_av);
printf("\nThe maximum energy gained by the electron
    = %5.1e eV", K_max);

// Result
// The average energy gained by the electron per
    turn = 482.5 eV
// The maximum energy gained by the electron = 1.2e
    +008 eV
```

Scilab code Exa 17.15 Maximum frequency of Dee voltage and gain in energy of the deuteron in a synchrocyclotron

```
// Scilab code Ex17.15 : Pg:896 (2011)
clc;clear;
q = 1.6e-019;     // Charge on a deuteron , C
amu = 931.5;     // Energy equivalent of 1 amu, MeV
m0 = 2.0141;     // Rest mass of a deuteron , kg
B0 = 1.5;     // Magnetic field at the centre of the
    synchrocyclotron , T
B = 1.431;     // Magnetic field at the periphery of
    the synchrocyclotron , T
f0 = q*B0/(2*3.14*m0*1.67e-027*1e+06);     // Maximum
     frequency of Dee voltage of synhrocyclotron , MHz
f = 1e+07;     // Minimum frequency of Dee voltage ,
    Hz
m = q*B/(2*3.14*f*1.67e-027);     // Mass of deuteron
     at the periphery of the Dee, amu
K = (m-m0)*amu;     // Gain in energy of the deuteron
    , MeV
printf("\nThe maximum frequency of Dee voltage = %5
    .2f MHz", f0);
printf("\nThe gain in energy of the deuteron = %6.2f
     MeV", K);

```

```
// Result
// The maximum frequency of Dee voltage = 11.36 MHz
// The gain in energy of the deuteron = 157.47 MeV
```

Scilab code Exa 17.16 Maximum radial field and the life of a GM counter

```
// Scilab code Ex17.16 : Pg:897 (2011)
clc;clear;
V = 1000;    // Operating voltage of the GM counter, volt
a = 1e-04'    // Radius of GM counter wire, m
b = 2e-02;    // Radius of cathode, m
E = V/(2.3026*a*log10(b/a));    // Maximum radial field at the surface of central wire of GM tube, V/m
tau = 1e+09;    // Life time of GM tube, counts
N = tau/(50*60*60*2000);    // Life of the GM counter, years
printf("\nThe maximum radial field at the surface of central wire of GM tube = %4.2e V/m", E);
printf("\nThe life of the GM counter = %4.2f years", N);

// Result
// The maximum radial field at the surface of central wire of GM tube = 1.89e+006 V/m
// The life of the GM counter = 2.78 years
```

Scilab code Exa 17.17 Avalanche voltage in a GM tube

```
// Scilab code Ex17.17 : Pg:898 (2011)
clc;clear;
```

```
I = 15.7;      // Ionization potential of argon in GM counter, volt
a = 0.012/2*1e-02;     // Radius of GM counter wire, m
b = 5/2*1e-02;     // Radius of cathode, m
lambda = 7.8e-006;     // Mean free path of argon in GM counter, m
// As E*lambda = I = V*lambda/(2.3026*a*log10(b/a)), solving for V
V = 2.3026*a*I*log10(b/a)/lambda;     // Voltage that must be applied to produce an avalanche in GM tube, volt
printf("\nThe voltage that must be applied to produce an avalanche in GM tube = %6.2f volt", V);

// Result
// The voltage that must be applied to produce an avalanche in GM tube = 728.52 volt
```

Scilab code Exa 17.18 Maximum permissible voltage fluctuation in a GM counter

```
// Scilab code Ex17.18 : Pg:898 (2011)
clc;clear;
count_err = 1e-03;     // Fractional error in counting
m = 3;     // Plateau slope
delta_V = count_err*100/m*100;     // Maximum permissible voltage fluctuation in a GM counter, volt
printf("\nThe maximum permissible voltage fluctuation in a GM counter = %3.1f volts", delta_V);

```

```
// Result
// The maximum permissible voltage fluctuation in a
   GM counter = 3.3 volts
```

Chapter 19

Superconductivity

Scilab code Exa 19.1 Critical field for lead at 4 K

```
// Scilab Code Ex19.1: Page-959 (2011)
clc; clear;
T_c = 6.2;     // Critical temperature of lead in
    superconducting state, K
T = 4;         // Temperature at which critical field
     of lead is to be found out, K
H_c0 = 0.064;    // Critical field for lead at 0 K,
    MA/m
H_cT = H_c0*(1-(T/T_c)^2);    // Critical field for
    lead at 4 K, MA/m
printf("\nThe critical field for lead at 4 K = %5.3f
     MA/m", H_cT);
// Result
// The critical field for lead at 4 K = 0.037 MA/m
```

Scilab code Exa 19.2 Isotopic effect in mercury

```
// Scilab Code Ex19.2: Page-959 (2011)
```

```
clc; clear;
T_c1 = 4.153;     // Critical temperature of mercury for its one isotope, K
M1 = 200.59;     // Mass of first isotope of mercury, amu
M2 = 204;        // Mass of second isotope of mercury, amu
// From isotopic effect of superconductivity,
// T_c2/T_c1 = sqrt(M1/M2), solving for T_c2
T_c2 = T_c1*sqrt(M1/M2);     // Critical temperature of mercury for second isotope, K
printf("\nThe critical temperature of mercury for its isotope of mass 204 amu = %5.3f K", T_c2);

// Result
// The critical temperature of mercury for its isotope of mass 204 amu = 4.118 K
```

Scilab code Exa 19.3 Critical current through superconducting aluminium wire

```
// Scilab Code Ex19.3: Page-960 (2011)
clc; clear;
d = 1e-003;     // Diameter of aluminium wire, m
r = d/2;        // Radius of aluminium wire, m
H_c = 7.9e+003;     // Critical magnetic field for Al, A/m
I_c = 2*3.14*r*H_c;     // Critical current through superconducting aluminium wire, A
printf("\nThe critical current through superconducting aluminium wire = %6.3f A", I_c);

// Result
// The critical current through superconducting aluminium wire = 24.806 A
```

Scilab code Exa 19.4 Critical current density for superconducting wire of lead at different temperatures

```
// Scilab Code Ex19.4: Page-960 (2011)
clc; clear;
T_c = 7.18;      // Critical temperature of lead in superconducting state, K
H_c0 = 6.5e+004;     // Critical field for lead at 0 K, A/m
// At T = 4.2 K
T = 4.2;         // Temperature at which critical field of lead is to be found out, K
H_cT = H_c0*(1-(T/T_c)^2);     // Critical field for lead at 4 K, A/m
d = 1e-003;      // Diameter of lead wire, m
r = d/2;         // Radius of lead wire, m
I_c = 2*3.14*r*H_cT;     // Critical current through superconducting lead wire, A
J_c = I_c/(3.14*r^2);     // Critical current density for superconducting lead wire, A/Sq. meter
printf("\nThe critical current density at %3.1f K = %5.3e A/Sq.m", T, J_c);
// At T = 7 K
T = 7;         // Temperature at which critical field of lead is to be found out, K
H_cT = H_c0*(1-(T/T_c)^2);     // Critical field for lead at 4 K, A/m
d = 1e-003;      // Diameter of lead wire, m
r = d/2;         // Radius of lead wire, m
I_c = 2*3.14*r*H_cT;     // Critical current through superconducting lead wire, A
J_c = I_c/(3.14*r^2);     // Critical current density for superconducting lead wire, A/Sq. meter
printf("\nThe critical current density at %3.1f K =
```

```
    %4.2e A/Sq.m", T, J_c);

// Result
// The critical current density at 4.2 K = 1.710e
    +008 A/Sq.m
// The critical current density at 7.0 K = 1.29e+007
     A/Sq.m
```

Scilab code Exa 19.5 Critical temperature of lead from London penetration depth

```
// Scilab Code Ex19.5:Page-961 (2011)
clc; clear;
T1 = 3;     // Initial temperature of lead wire, K
T2 = 7.1;     // Final temperature of lead wire, K
lambda1 = 39.6;     // Initial London penetration
    depth for lead, mm
lambda2 = 173;     // Final London penetration depth
    for lead, mm
// As lambda_T = lambda_0*(1-(T/T_c)^4)^(-1/2) so
// (lambda1/lambda2)^2 = (T_c^4 - T2^4)/(T_c^4 - T1
    ^4)
// Solving for T_c
T_c = ((T2^4-T1^4*(lambda1/lambda2)^2)/(1-(lambda1/
    lambda2)^2))^(1/4);
printf("\nThe critical temperature of lead = %5.3f K
    ", T_c);

// Result
// The critical temperature of lead = 7.193 K
```

Scilab code Exa 19.6 Field strength required by lead to lose its superconducting state at 0 K

```
// Scilab Code Ex19.6: Page-962 (2011)
clc; clear;
T_c = 7.2;    // Critical temperature of lead in
    superconducting state, K
T = 5;        // Temperature at which lead loses its
     superconducting state, K
H_cT = 3.3e+004;    // Critical magnetic field for
    superconducting lead at 5 K, A/m
// As H_cT = H_c0*(1-(T/T_c)^2), solving for H_c0
H_c0 = H_cT/(1-(T/T_c)^2);    // Critical field for
    lead at 0 K, A/m
printf("\nThe critical magnetic field for lead at 0
   K = %4.2e A/m", H_c0);

// Result
// The critical magnetic field for lead at 0 K =
    6.37e+004 A/m
```

Scilab code Exa 19.7 Critical temperature for niobium

```
// Scilab Code Ex19.7: Page-962 (2011)
clc; clear;
H_c0 = 2e+005;    // Critical field for niobium at 0
     K, A/m
H_cT = 1e+005;    // Critical magnetic field for
    superconducting niobium at 5 K, A/m
T = 8;        // Temperature at which lead loses its
     superconducting state, K
// As H_cT = H_c0*(1-(T/T_c)^2), solving for T_c
T_c = T/(1-H_cT/H_c0)^(1/2);
printf("\nThe critical temperature for niobium = %6
    .3f K", T_c);

// Result
// The critical temperature for niobium = 11.314 K
```