Scilab Textbook Companion for Engineering Physics
by G. Aruldhas[1]

Created by
Pankaj Biswas
Modern Physics
Physics
Shri Mata Vaishno Devi University
College Teacher
Dr. Vinay Kumar
Cross-Checked by

August 10, 2013

---

[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: Engineering Physics

Author: G. Aruldhas

Publisher: Phi Learning Pvt. Ltd.

Edition: 1

Year: 2010

ISBN: 9788120339163

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

OSCILLATION AND WAVES

Scilab code Exa 1.1 Time period of SHM

```
// Scilab Code Ex1.1 : Page-23 (2010)
A = 4/2;    // Amplitude of SHM, cm
x = 0;    // Mean position of oscillating particle, cm
v = 12;    // Velocity of the particle at the mean position, cm/s
// As v = omega*sqrt(A^2 - x^2), solving for omega
omega = v/sqrt(A^2 - x^2);
printf("\nThe time period of SHM = %5.2f s", (2*%pi)/omega);

// Result
// The time period of SHM =  1.05 s
```

Scilab code Exa 1.2 Accelertion and maximum velocity in SHM

```
// Scilab Code Ex1.2 : Page-23 (2010)
T = 0.1;    // Time period of oscillation in SHM, s
```

```
x = 0.2;     // Position of the particle from its mean position, cm
A = 4;     // Amplitude of the particle executing SHM, cm
// As T = 2*%pi/omega, solving for omega
omega = 2*%pi/T;     // Angular speed of particle executing SHM, per sec
a = omega^2*x;     // Accelertion of particle executing SHM, cm per sec square
v_max = omega*A;     // Maximum velocity of the particle in SHM, cm per sec
printf("\nThe accelertion of particle executing SHM = %5.1f cm per sec square", a);
printf("\nThe maximum velocity of the particle in SHM = %5.1f cm per sec", v_max);

// Result
// The accelertion of particle executing SHM = 789.6 cm per sec square
// The maximum velocity of the particle in SHM = 251.3 cm per sec
```

Scilab code Exa 1.3 Damped Vibrating System

```
// Scilab Code Ex1.3 : Page-24 (2010)
A1 = 40;     // First amplitude of oscillation, cm
An_plus_1 = 4;     // Amplitude after 100 oscillations, cm
n = 100;     // Number of oscillations
T = 2.5;     // Time period of oscillations, s
t = T/4;     // Time taken to reach the first amplitude from the mean position, s
// Now A1 = x0*exp(-lambda*t) and An_plus_1 = x0*exp(-lambda*(t+nT))
// A1/An_plus_1 = exp(n*lambda*T), solving for
```

```
lambda
lambda = log(A1/An_plus_1)/(n*T);    // Damping constant. per sec
printf("\nDamping constant = %3.2e per sec", lambda);

// Result
// Damping constant = 9.21e-003 per sec
```

Scilab code Exa 1.4 Amplitude and Time Period in SHM

```
// Scilab Code Ex1.4 :  Page-24 (2010)
v1 = 16;    // Velocity of particle executing SHM at position 3 cm
v2 = 12;    // Velocity of particle executing SHM at position 4 cm
x1 = 3;    // First position of the particle, cm
x2 = 4;    // Second position of the particle, cm
// As v = omega*sqrt(A^2 - x^2) so
// (v1/v2)^2 = (A^2 - x1^2)/(A^2 - x2^2), solving for A
A = poly(0, 'A');    // Declare variable A
A = roots((A^2 - x1^2)*v2^2-(A^2 - x2^2)*v1^2);
printf("\nThe amplitude of SHM = %1d cm", A(1));
// v = omega*sqrt(A^2 - x^2), solving for omega
omega = v1/sqrt(A(1)^2 - x1^2);    // Angular speed of the particle, rad per sec
T = 2*%pi/omega;    // Time period of oscillation, sec
printf("\nThe time period of oscillation = %5.3f sec", T);

// Result
// The amplitude of SHM = 5 cm
// The time period of oscillation = 1.571 sec
```

Scilab code Exa 1.5 Oscillation of a spring mass system

```
// Scilab Code Ex1.5 : Page-25 (2010)
m = 0.3;     // Mass attached to the string, kg
g = 9.8;     // Acceleration due to gravity, metre
    per sec square
x = 0.15;     // Stretchness produced in the spring,
   m
F = m*g;     // Restoring force acting on the mass, N
k = F/x;     // Spring constant, newton per metre
A = 0.1;     // Amplitude of the string, m
omega = sqrt(k/m);     // Angular frequency of
    oscillation, rad per sec
v0 = omega*A;     // Maximum velocity during the
    oscillations, m/s
printf("\nThe spring constant = %4.1f newton per
    metre", k);
printf("\nThe amplitude of oscillation = %2.1f m", A
    );
printf("\nThe maximum velocity during oscillations =
     %3.2f m/s", v0);

// Result
// The spring constant = 19.6 newton per metre
// The amplitude of oscillation = 0.1 m
// The maximum velocity during oscillations = 0.81 m
    /s
```

Scilab code Exa 1.6 Frequency of visible region

```
// Scilab Code Ex1.6 :  Page-25 (2010)
```

```
lambda1 = 400e-09;    // Lower limit of wavelength of visible region, m
lambda2 = 700e-09;    // Upper limit of wavelength of visible region, m
c = 3e+08;    // Speed of light in vacuum, m/s
f1 = c/lambda1;    // Upper limit of frequency of visible region, m
f2 = c/lambda2;    // Lower limit of frequency of visible region, m
printf("\nThe frequency equivalent of %3g nm to %3g nm is %3.1e Hz to %3.1e Hz", lambda1/1e-09, lambda2/1e-09, f1, f2);

// Result
// The frequency equivalent of 400 nm to 700 nm is 7.5e+014 Hz to 4.3e+014 Hz
```

Scilab code Exa 1.7 Characteristics of sound wave

```
// Scilab Code Ex1.7 : Page-26 (2010)
// Comparing the standard equation
// u(x,t) = A*sin(2*%pi(x/lambda-t/T))
// with the given equation, we get
A = 1.5e-03;    // Amplitude of the sound wave, m
lambda = 8;    // Wavelength of the sound wave, m
T = 1/40;    // Time period of the sound wave, s
nu = 1/T;    // Frequency of the sound wave, Hz
v = nu*lambda;    // Velocity of the sound wave, m/s
printf("\nThe amplitude of the sound wave = %3.1e m", A);
printf("\nThe wavelength of the sound wave = %1d m", lambda);
printf("\nThe time period of the sound wave = %3.2f s", T);
printf("\nThe frequency of the sound wave = %2d Hz",
```

```
    nu);
printf("\nThe velocity of the sound wave = %3d m/s",
    v);


// Result
// The amplitude of the sound wave = 1.5e-003 m
// The wavelength of the sound wave = 8 m
// The time period of the sound wave = 0.03 s
// The frequency of the sound wave = 40 Hz
// The velocity of the sound wave = 320 m/s
```

Scilab code Exa 1.8 Equation of a wave moving along X axis

```
// Scilab Code Ex1.8 :  Page-26 (2010)
A = 2;    // Amplitude of the wave, cm
T = 0.5;    // Time period of the wave, sec
v = 200;    // Wave velocity , cm/s
f = 1/0.5;    // Frequency of the wave, Hz
lambda = v/f;    // Wavelength of the wave, cm
printf("\nThe Equation of the wave moving along X-
   axis :");
printf("u = %1d*sin*2*pi*(x/%3d-t/%2.1f)", A, lambda
   , T);


// Result
// The Equation of the wave moving along X-axis :u =
    2*sin*2*pi*(x/100-t/0.5)
```

Scilab code Exa 1.9 Wave in the wire

```
// Scilab Code Ex1.9 : Page-27 (2010)
```

```
T = 1000;    // Tension in the wire, N
m = 15/300;    // Mass per unit length of the wire, kg per metre
lambda = 0.30;    // Wavelength of wave along wire, m
v = sqrt(T/m);    // Velocity of wave through wire, m/s
nu = v/lambda;    // Frequency of wave through string, Hz
printf("\nThe velocity and frequency of the wave through wire are %5.1f m/s and %5.1f Hz respectively", v, nu);



// Result
// The velocity and frequency of the wave through wire are 141.4 m/s and 471.4 Hz respectively
```

Chapter 2

ELECTROMAGNETIC THEORY

Scilab code Exa 2.1 Peak value of displacement current

```
// Scilab Code Ex2.1 :  Page-46 (2010)
function V = f(t)
    V = 0.2*sin(120*%pi*t);
endfunction
t = 0;    // Time when peak value of current occurs
C = 10e-012;    // Capacitance of the capacitor, farad
I = C*derivative(f,t);
printf("\nThe peak value of displacement current = %6.4e A", I);

// Result
// The peak value of displacement current = 7.5398e-010 A
```

Scilab code Exa 2.2 Displacement current density in a good conductor

```
// Scilab Code Ex2.2 : Page-46 (2010)
function E = fn(t)
    E = sin(120*%pi*t);
endfunction
epsilon_r = 1;    // Relative electrical permittivity of free space
epsilon_0 = 8.854e-012;    // Absolute electrical permittivity of free space, farad per metre
t = 0;    // Time when peak value of current occurs
J2 = epsilon_0*epsilon_r*derivative(fn,t);
printf(”\nThe peak value of displacement current = %4.2e ampere per metre square”, J2);

// Result
// The peak value of displacement current = 3.34e-009 ampere per metre square
```

Scilab code Exa 2.4 Poynting vector

```
// Scilab Code Ex2.4 : Page-47 (2010)
p = 60;    // Power rating of bulb, watt
d = 0.5;    // Distance from the blb, m
P = p/(4*%pi*d^2);    // Value of Poynting vector, watt per metre square
printf(”\nThe value of Poynting vector = %4.1f watt per metre square”, P);

// Result
// The value of Poynting vector = 19.1 watt per metre square
```

Scilab code Exa 2.5 Plane electromagnetic wave in a medium

```
// Scilab Code Ex2.5 : Page−47 (2010)
E_peak = 6;    // Peak value of electric field intensity, V/m
c = 3e+08;    // Speed of electromagnetic wave in free space, m/s
mu_0 = 4*%pi*1e-07;    // Absolute permeability of free space, tesla metre per ampere
epsilon_0 = 8.854e-012;    // Absolute permittivity of free space, farad/m
mu_r = 1;    // Relative permeability of medium
epsilon_r = 3;    // Relative permittivity of the medium
v = c/sqrt(mu_r*epsilon_r);    // Wave velocity, m/s
eta = sqrt((mu_0/epsilon_0)*(mu_r/epsilon_r));    // Intrinsic impedance of the medium, ohm
H_P = E_peak*sqrt((epsilon_0*epsilon_r)/(mu_0*mu_r));    // Peak value of the magnetic intensity, ampere per metre
printf(”\nThe wave velocity = %5.3e m/s”, v);
printf(”\nThe intrinsic impedance of the medium = %6.2f ohm”, eta);
printf(”\nThe peak value of the magnetic intensity = %4.2e A/m”, H_P);

// Result
// The wave velocity = 1.732e+008 m/s
// The intrinsic impedance of the medium = 217.51 ohm
// The peak value of the magnetic intensity = 2.76e−002 A/m
```

Chapter 3

INTERFERENCE

Scilab code Exa 3.1 Wavelength of Light using Young Double Slit experiment

```
// Scilab Code Ex3.1 : Page-71 (2010)
beta = 0.51e-02;    // Fringe width, cm
d = 2.2e-02;        // Distance between the slits, cm
D = 2e+02;     // Distance between the slits and the
   screen, cm
// As beta = D*lambda/d, solving for lambda
lambda = beta*d/D;    // Wavelength of light, m
printf("\nThe wavelength of light = %4d angstrom",
   lambda/1e-010);

// Result
// The wavelength of light = 5610 angstrom
```

Scilab code Exa 3.2 Fringe shift due to change in wavelength

```
// Scilab Code Ex3.2 : Page-71 (2010)
lambda1 = 4250e-010;    // First wavelength emitted
   by source of light, m
```

```
lambda2 = 5050e-010;     // Second wavelength emitted by source of light , m
D = 1.5;     // Distance between the source and the screen , m
d = 0.025e-03;        // Distance between the slits , m
n = 3;     // Number of fringe from the centre
x3 = n*lambda1*D/d;     // Position of third bright fringe due to lambda1 , m
x3_prime = n*lambda2*D/d;     // Position of third bright fringe due to lambda2 , m
printf(”\nThe separation between the third bright fringe due to the two wavelengths = %4.2f cm”, (x3_prime - x3)/1e-02);

// Result
// The separation between the third bright fringe due to the two wavelengths = 1.44 cm
```

Scilab code Exa 3.3 Refractive index from double slit experiment

```
// Scilab Code Ex3.3 : Page-71 (2010)
lambda = 5.5e-05;     // Wavelength emitted by source of light , cm
n = 4;     // Number of fringes shifted
t = 3.9e-04;     // Thickness of the thin glass sheet , cm
mu = n*lambda/t+1;     // Refractive index of the sheet of glass
printf(”\nThe refractive index of the sheet of glass = %6.4f”, mu);

// Result
// The refractive index of the sheet of glass = 1.5641
```

Scilab code Exa 3.4 Interference by thin soap film

```
// Scilab Code Ex3.4 : Page-72 (2010)
lambda = 5893e-010;     // Wavelength of monochromatic lihgt used, m
n = 1;     // Number of fringe for the least thickness of the film
r = 0;     // Value of refraction angle for normal incidence, degrees
mu = 1.42;     // refractive index of the soap film
// As for constructive interference,
// 2*mu*t*cos(r) = (2*n-1)*lambda/2, solving for t
t = (2*n-1)*lambda/(4*mu*cos(r));     // Thickness of the film that appears bright, m
printf("\nThe thickness of the film that appears bright = %6.1f angstrom", t/1e-010);
// As for destructive interference,
// 2*mu*t*cos(r) = n*lambda, solving for t
t = n*lambda/(2*mu*cos(r));     // Thickness of the film that appears bright, m
printf("\nThe thickness of the film that appears dark = %4d angstrom", t/1e-010);

// Result
// The thickness of the film that appears bright = 1037.5 angstrom
// The thickness of the film that appears dark = 2075 angstrom
```

Scilab code Exa 3.5 Interference due to thin air wedge

```
// Scilab Code Ex3.5 : Page-72 (2010)
lambda = 5893e-008;    // Wavelength of monochromatic lihgt used, m
n = 10;    // Number of fringe that are found in the distnace of 1 cm
d = 1;    // Distance of 10 fringes, cm
beta = d/n;    // Fringe width, cm
theta = lambda/(2*beta);    // Angle of the wedge, rad
printf("\nThe angle of the wedge = %5.3e rad", theta);

// Result
// The angle of the wedge = 2.946e-004 rad
```

Scilab code Exa 3.6 Separation between consecutive bright fringes formed by an air wedge

```
// Scilab Code Ex3.6 : Page-72 (2010)
lambda = 5900e-008;    // Wavelength of monochromatic lihgt used, m
t = 0.010e-01;    // Spacer thickness, cm
l = 10;    // Wedge length, cm
theta = t/l;    // Angle of the wedge, rad
beta = lambda/(2*theta);    // Fringe width, cm
printf("\nThe separation between consecutive bright fringes = %5.3e cm", beta);

// Result
// The separation between consecutive bright fringes = 2.950e-001 cm
```

Scilab code Exa 3.7 Newton Rings by reflected light

```
// Scilab Code Ex3.7 : Page-72 (2010)
D4 = 0.4;     // Diameter of 4th dark ring, cm
D12 = 0.7;     // Diameter of 12th dark ring, cm
// We have dn_puls_k^2-Dn^2 = 4*k*R*lambda, so
// D12^2-D4^2 = 32*R*lambda and D20^2-D12^2 = 32*R*lambda for k = 8, solving for D20
D20 = sqrt(2*D12^2-D4^2);     // Diameter of 20th dark ring, cm
printf("\nThe diameter of 20th dark ring = %6.4f cm", D20);

// Result
// The diameter of 20th dark ring = 0.9055 cm
```

Scilab code Exa 3.8 Refractive index from Newton Rings arrangement

```
// Scilab Code Ex3.8 : Page-73 (2010)
Dn = 0.30;     // Diameter of nth dark ring with air film, cm
dn = 0.25;     // Diameter of nth dark ring with liquid film, cm
mu = (Dn/dn)^2;     // Refractive index of the liquid
printf("\nThe refractive index of the liquid = %4.2f", mu);

// Result
// The refractive index of the liquid = 1.44
```

Scilab code Exa 3.9 Wavelength of light using Michelson Interferometer

```
// Scilab Code Ex3.9 : Page-73 (2010)
x = 0.002945;     // Distance through which movable mirror is shifted, cm
```

```
N = 100;     // Number of fringes shifted
lambda = 2*x/N;     // Wavelength of light, m
printf("\nThe wavelength of light = %4d angstrom", lambda/1e-008);

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

Scilab code Exa 3.10 Shift in movable mirror of Michelson Interferometer

```
// Scilab Code Ex3.10 : Page-73 (2010)
lambda1 = 5896e-008;     // Wavelength of D1 line of sodium, m
lambda2 = 5890e-008;     // Wavelength of D2 line of sodium, m
lambda = (lambda1+lambda2)/2;
// As lambda1 - lambda2 = lambda^2/(2*x), solving for x
x = lambda^2/(2*(lambda1 - lambda2));     // Shift in movable mirror of Michelson Interferometer, cm
printf("\nThe shift in movable mirror = %5.3f mm", x/1e-001);

// Result
// The shift in movable mirror = 0.289 mm
```

Chapter 4

DIFFRACTION

Scilab code Exa 4.1 Diffraction at a single slit

```
// Scilab Code Ex4.1 : Page-91 (2010)
D = 50;    // Distance between source and the screen, cm
lambda = 6563e-008;    // Wavelength of light of parallel rays, m
d = 0.385e-01;        // Width of the slit, cm
n = 1;    // Order of diffraction for first minimum
// As sin(theta1) = n*lambda/d = x1/D, solving for x1
x1 = n*lambda*D/d;    // Distance from the centre of the principal maximum to the first minimum, cm
printf("\nThe Distance from the centre of the principal maximum to the first minimum = %4.2f mm", x1/1e-001);
n = 5;    // Order of diffraction for fifth minimum
x2 = n*lambda*D/d;    // Distance from the centre of the principal maximum to the fifth minimum, cm
printf("\nThe Distance from the centre of the principal maximum to the fifth minimum = %4.2f mm", x2/1e-001);

```

```
// Result
// The Distance from the centre of the principal maximum to the first minimum = 0.85 mm
// The Distance from the centre of the principal maximum to the fifth minimum = 4.26 mm
```

Scilab code Exa 4.2 Diffraction at a circular aperture

```
// Scilab Code Ex4.2 : Page-91 (2010)
D = 0.04;     // Diameter of circular aperture, cm
f = 20;     // Focal length of convex lens, cm
lambda = 6000e-008;     // Wavelength of light used, m
// We have sin(theta) = 1.22*lambda/D = theta, for small theta, such that
// For first dark ring
theta = 1.22*lambda/D;     // The half angular width at central maximum, rad
r1 = theta*f;     // The half width of central maximum for first dark ring, cm
// We have sin(theta) = 5.136*lambda/(%pi*D) = theta, for small theta, such that
// For second dark ring
theta = 5.136*lambda/(%pi*D);     // The half angular width at central maximum, rad
r2 = theta*f;     // The half width of central maximum for second dark ring, cm
printf(”\nThe radius of first dark ring = %4.2e cm”, r1);
printf(”\nThe radius of second dark ring = %4.1e cm”, r2);

// Result
// The radius of first dark ring = 3.66e-002 cm
// The radius of second dark ring = 4.90e-002 cm
```

Scilab code Exa 4.3 Second order maximum for diffraction grating

```
// Scilab Code Ex4.3 : Page-91 (2010)
n = 2;    // Order of diffraction
lambda = 650e-009;    // Wavelength of light used, m
d = 1.2e-05;    // Distance between two consecutive slits of grating, m
// We have sin(theta) = n*N*lambda = n*lambda/d, solving for theta
theta = asind(n*lambda/d);    // Angle at which the 650 nm light produces a second order maximum, degrees
printf("\nThe angle at which the 650 nm light produces a second order maximum = %4.2f degrees", theta);

// Result
// The angle at which the 650 nm light produces a second order maximum = 6.22 degrees
```

Scilab code Exa 4.4 The highest spectral order with diffraction grating

```
// Scilab Code Ex4.4 : Page-92 (2010)
lambda = 650e-009;    // Wavelength of light used, m
N = 6000e+02;    // Number of lines per m on grating, per m
theta = 90;    // Angle at which the highest spectral order is obtained, degrees
// We have sin(theta) = n*N*lambda, solving for n
n = sind(theta)/(N*lambda);    // The highest order of spectra with diffraction grating
```

```
printf(”\nThe highest order of spectra obtained with
    diffraction grating = %1d”, n);

// Result
// The highest order of spectra obtained with
   diffraction grating = 2
```

Scilab code Exa 4.5 Overlapping spectra with diffraction grating

```
// Scilab Code Ex4.5 : Page-92 (2010)
N = 4000e+02;    // Number of lines per m on grating
   , per m
// For Blue Line
lambda = 450e-009;    // Wavelength of blue light , m
n = 3;    // Order of diffraction spectrum
// We have sin(theta) = n*N*lambda, solving for sin(
   theta)
sin_theta_3 = n*N*lambda;    // Sine of angle at
   third order diffraction
// For Red Line
lambda = 700e-009;    // Wavelength of blue light , m
n = 2;    // Order of diffraction spectrum
// We have sin(theta) = n*N*lambda, solving for sin(
   theta)
sin_theta_2 = n*N*lambda;    // Sine of angle at
   second order diffraction
// Check for overlapping
if abs(sin_theta_3 - sin_theta_2) < 0.05 then
    printf(”\nThe two orders overlap.”);
else
    printf(”\nThe two orders do not overlap.”);
end

// Result
// The two orders overlap.
```

Scilab code Exa 4.6 Width of first order spectrum

```
// Scilab Code Ex4.6 : Page-93 (2010)
n = 1;    // Order of diffraction spectrum
N = 6000e+02;    // Number of lines per m on
    diffraction grating, per m
D = 2;    // Distance of screen from the source, m
lambda1 = 400e-009;    // Wavelength of blue light,
    m
// We have sin(theta1) = n*N*lambda, solving for
    theta1
theta1 = asind(n*N*lambda1);    // Angle at first
    order diffraction for Blue light, degrees
lambda2 = 750e-009;    // Wavelength of blue light,
    m
// We have sin(theta2) = n*N*lambda, solving for
    theta2
theta2 = asind(n*N*lambda2);    // Angle at first
    order diffraction for Red light, degrees
x1 = D*tand(theta1);    // Half width position at
    central maximum for blue color, m
x2 = D*tand(theta2);    // Half width position at
    central maximum for red color, m

printf("\nThe width of first order spectrum on the
    screen = %4.1f cm", (x2 - x1)/1e-02);

// Result
// The width of first order spectrum on the screen =
    51.3 cm
```

Scilab code Exa 4.7 Resolution of wavelengths for grating

```
// Scilab Code Ex4.7 : Page-93 (2010)
w = 5;    // Width of the grating, cm
N = 320;    // Number of lines per cm on grating,
   per cm
N0 = w*N;    // Total number of lines on the grating
lambda = 640;    // Wavelength of light, nm
n = 2;    // Order of diffraction
d_lambda = lambda/(n*N0);    // Separation between
   wavelengths which the gratign can just resolve,
   nm
printf("\nThe separation between wavelengths which
   the grating can just resolve = %3.1f nm",
   d_lambda);

// Result
// The separation between wavelengths which the
   grating can just resolve = 0.2 nm
```

Scilab code Exa 4.8 Angular separation to satisfy Rayleigh criterion

```
// Scilab Code Ex4.8 : Page-93 (2010)
lambda = 550e-09;    // Wavelength of light, m
D = 3.2e-02;    // Diameter of circular lens, m
f = 24e-02;    // Focal length of the lens, m
theta_min = 1.22*lambda/D;    // Minimum angle of
   resolution provided by the lens, rad
// As delta_x/f = theta_min, solving for delta_x
delta_x = theta_min*f;    // Separation of the
   centres of the images in the focal plane of lens,
   m
printf("\nThe separation of the centres of the
   images in the focal plane of lens = %1d micro-
   metre", delta_x/1e-06);

// Result
```

```
// The separation of the centres of the images in the focal plane of lens = 5 micro-metre
```

Scilab code Exa 4.9 Linear separation between two points

```
// Scilab Code Ex4.9 : Page-94 (2010)
lambda = 550e-09;     // Wavelength of light, m
D = 20e-02;     // Diameter of objective of telescope, m
d = 6e+003;     // Distance of two points from the objective of telescope, m
theta = 1.22*lambda/D;     // Angular separation between two points, rad
x = theta*d;     // Linear separation between two points, m
printf("\nThe linear separation between two points = %5.2f mm", x/1e-03);

// Result
// The linear separation between two points = 20.13 mm
```

Chapter 5

POLARIZATION

Scilab code Exa 5.1 Polarization by reflection

```
// Scilab Code Ex5.1 : Polarization by reflection: Page-113 (2010)
mu_g = 1.72;    // Refractive index of glass
mu_w = 4/3;    // Refractive index of water
// For polarization to occur on flint glass, tan(i) = mu_g/mu_w
// Solving for i
i = atand(mu_g/mu_w);
printf("\nThe angle of incidence for complete polarization to occur on flint glass = %4.1f degrees", i);
// For polarization to occur on water, tan(i) = mu_w/mu_g
// Solving for i
i = atand(mu_w/mu_g);
printf("\nThe angle of incidence for complete polarization to occur on water = %5.2f degrees", i);

// Result
// The angle of incidence for complete polarization
```

```
to occur on flint glass = 52.2 degrees
// The angle of incidence for complete polarization to occur on water = 37.78 degrees
```

Scilab code Exa 5.2 Percentage transmission of polarized light

```
// Scilab Code Ex5.2 : Percentage transmission of polarized light: Page-113 (2010)
I0 = 1;     // For simplicity, we assume the intensity of light falling on the second Nicol prism to be unity, watt per metre square
theta = 30;     // Angle through which the crossed Nicol is rotated, degrees
I = I0*cosd(90-theta)^2;     // Intensity of the emerging light from second Nicol, watt per metre square
T = I/(2*I0)*100;     // Percentage transmission of incident light
printf("\nThe percentage transmission of incident light after emerging through the Nicol prism = %4.1f percent", T);

// Result
// The percentage transmission of incident light after emerging through the Nicol prism = 12.5 percent
```

Scilab code Exa 5.3 Thickness of Quarter Wave Plate

```
// Scilab Code Ex5.3 : Thickness of Quarter Wave Plate : Page-113 (2010)
lambda = 6000e-008;     // Wavelength of incident light, cm
```

```
mu_e = 1.55;     // Refractive index of extraordinary ray
mu_o = 1.54;     // Refractive index of ordinary ray
t = lambda/(4*(mu_e - mu_o));     // Thickness of Quarter Wave plate of positive crystal, cm
printf("\nThe thickness of Quarter Wave plate = %6.4f cm", t);

// Result
// The thickness of Quarter Wave plate = 0.0015 cm
```

Scilab code Exa 5.4 Behaviour of half wave plate for increased wavelength

```
// Scilab Code Ex5.4 : Behaviour of half wave plate for increased wavelength : Page-114 (2010)
lambda = 1;     // For simplicity, wavelength of incident light is assumed to be , cm
mu_e = 1.55;     // Refractive index of extraordinary ray
mu_o = 1.54;     // Refractive index of ordinary ray
t = lambda/(2*(mu_e - mu_o));     // Thickness of Half Wave plate for given lambda, cm
t_prime = 2*lambda/(2*(mu_e - mu_o));   // Thickness of Half Wave plate for twice lambda, cm
printf("\nThe thickness of half wave plate is %2.1f times that of the quarter wave plate.", t/t_prime);
printf("\nThe half wave plate behaves as a quarter wave plate for twice the wavelength of incident light.");

// Result
// The thickness of half wave plate is 0.5 times that of the quarter wave plate.
// The half wave plate behaves as a quarter wave
```

```
plate for twice the wavelength of incident light.
```

Scilab code Exa 5.5 Phase retardation for quartz

```
// Scilab Code Ex5.5 : Phase retardation for quartz : Page-114 (2010)
lambda = 500e-09;    // Wavelength of incident light, m
mu_e = 1.5508;    // Refractive index of extraordinary ray
mu_o = 1.5418;    // Refractive index of ordinary ray
t = 0.032e-03;    // Thickness of quartz plate, m
dx = (mu_e - mu_o)*t;    // Path difference between E-ray and O-ray, m
dphi = (2*%pi)/lambda*dx;    // Phase retardation for quartz for given wavelength, rad
printf("\nThe phase retardation for quartz for given wavelength = %5.3f pi rad", dphi/%pi);

// Result
// The phase retardation for quartz for given wavelength = 1.152 pi rad
```

Scilab code Exa 5.6 Brewster angle at the boundary between two materials

```
// Scilab Code Ex5.6 : Brewster angle at the boundary between two materials : Page-114 (2010)
C = 52;    // Critical angle for total internal reflection at a boundary between two materials, degrees
// From Brewster's law, tand(i_B) = 1_mu_2
```

```
// Also sind(C) = 1_mu_2, so that
// tand(i_B) = sind(C), solving for i_B
i_B = atand(sind(C));     // Brewster angle at the boundary, degrees
printf("\nThe Brewster angle at the boundary between two materials = %2d degrees", i_B);

// Result
// The Brewster angle at the boundary between two materials = 38 degrees
```

Chapter 6

CRYSTALLOGRAPHY

Scilab code Exa 6.1 Lattice parameter of NaCl crystal

```
// Scilab Code Ex6.1 : Lattice parameter of NaCl crystal : Page-134 (2010)
M = 23+35.5;           // Molecular weight of NaCl, kg per k-mole
d = 2.18e+03;     // Density of rock salt, kg per metre cube
n = 4;     // No. of atoms per unit cell for an fcc lattice of NaCl crystal
N = 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/(d*N))^(1/3);     // Lattice constant of unit cell of NaCl
printf("\nLattice parameter for the NaCl crystal = %4.2f angstrom", a/1e-010);

// Result
// Lattice parameter for the NaCl crystal = 5.63 angstrom
```

Scilab code Exa 7.1 Variation of critical magnetic field with temperature

```
// Scilab Code Ex7.1 : Variation of critical magnetic field with temperature : Page-152 (2010)
T_c = 3.722;    // Critical temperature of superconducting transition, kelvin
B_c0 = 0.0306;    // Critical magnetic field to destroy superconductivity, tesla
T = 2;    // Temperature at which critical magnetic field is to be found out, kelvin
B_cT = B_c0*(1-(T/T_c)^2);
printf("\nThe critical magnetic field at %d K = %6.4f T", T, B_cT);

// Result
// The critical magnetic field at 2 K = 0.0218 T s
```

Scilab code Exa 6.2 Miller indices of the crystal plane

```
// Scilab Code Ex6.2 : Miller indices of the crystal plane : Page-134 (2010)
m = 3; n = 2; p = 1; // Coefficients of intercepts along three axes
m_inv = 1/m;        // Reciprocate the first coefficient
n_inv = 1/n;        // Reciprocate the second coefficient
p_inv = 1/p;        // Reciprocate the third coefficient
mul_fact = double(lcm(int32([m,n,p]))); // Find l.c.m. of m,n and p
m1 = m_inv*mul_fact;    // Clear the first fraction
```

```
m2 = n_inv*mul_fact;    // Clear the second fraction
m3 = p_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 : (2 3 6)
```

Scilab code Exa 6.3 Indices of lattice plane

```
// Scilab Code Ex6.3 : Indices of lattice plane :
   Page-135 (2010)
m = 2; // Coefficient of intercept along x-axis
n = %inf;    // Coefficient of intercept along y-
   axis
p = 3/2;    // Coefficient of intercept along z-axis
m_inv = 1/m;    // Reciprocate m
n_inv = 1/n;    // Reciprocate n
p_inv = 1/p;    // Reciprocate p
mul_fact = 6;    // multiplicative factor, L.C.M. of
    2 and 3 i.e. 6
m1 = m_inv*mul_fact;    // Clear the first fraction
m2 = n_inv*mul_fact;    // Clear the second fraction
m3 = p_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 : 3, 0, 4
```

Scilab code Exa 6.5 Interplanar spacing in cubic crystal

```
// Scilab Code Ex6.5 : Interplanar spacing in cubic
   crystal: Page-136 (2010)

// For (110) planes
h = 1; k = 1; l = 0; // Miller Indices for planes in
    a cubic crystal
a = 0.43e-009;     // Interatomic spacing, m
d = a/(h^2+k^2+l^2)^(1/2);  // The interplanar
   spacing for cubic crystals, m
printf("\nThe interplanar spacing between
   consecutive (110) planes = %4.2f angstrom", d/1e
   -010);

// For (212) planes
h = 2; k = 1; l = 2; // Miller Indices for planes in
    a cubic crystal
a = 4.21D-10;     // Interatomic spacing, m
d = a/(h^2+k^2+l^2)^(1/2);  // The interplanar
   spacing for cubic crystals, m
printf("\nThe interplanar spacing between
   consecutive (212) planes = %4.3f angstrom", d/1e
   -010);

// Result
// The interplanar spacing between consecutive (110)
    planes = 3.04 angstrom
// The interplanar spacing between consecutive (212)
    planes = 1.403 angstrom
```

Scilab code Exa 6.6 Interplanar spacing in cubic crystal

```
// Scilab Code Ex6.6 : Interplanar spacing in cubic
   crystal: Page-136 (2010)
h = 2; k = 3; l = 1; // Miller Indices for planes in
    a cubic crystal
```

```
r = 0.175e-009;    // Atomic radius of fcc lattice, m
a = 2*sqrt(2)*r;    // Interatomic spacing of fcc lattice, m
d = a/(h^2+k^2+l^2)^(1/2);  // The interplanar spacing for cubic crystals, m
printf("\nThe interplanar spacing between consecutive (231) planes = %4.2f ansgtrom", d/1e-010);

// Result
// The interplanar spacing between consecutive (231) planes = 1.32 ansgtrom
```

Scilab code Exa 6.7 ngle of reflection by using wavelength of X ray

```
// Scilab Code Ex6.7 : Angle of reflection by using wavelength of X-ray: Page-136 (2010)
lambda = 1.440e-010;    // Wavelength of X-rays, m
d = 2.8e-010;    // Interplanar spacing of rocksalt crystal, m
// 2*d*sin(theta) = n*lambda    **Bragg's law, n is the order of diffraction
// Solving for theta, we have

// For Ist Order diffraction
n = 1;
theta = asind(n*lambda/(2*d));    // Angle of diffraction, degrees
printf("\nThe angle of reflection for first order diffraction = %4.1f degrees", theta);

// For IInd Order diffraction
n = 2;
theta = asind(n*lambda/(2*d));    // Angle of
```

```
    diffraction, degrees
printf("\nThe angle of reflection for first order
    diffraction = %4.1f degrees", theta);

// Result
// The angle of reflection for first order
    diffraction = 14.9 degrees
// The angle of reflection for first order
    diffraction = 30.9 degrees
```

Scilab code Exa 6.8 Actual volume occupied by the spheres in fcc structure

```
// Scilab Code Ex6.8 : Actual volume occupied by the
     spheres in fcc structure  Page-136 (2010)
N = 8*1/8 + 6*1/2;    // total number of spheres in
    a unit cell
a = 1;    // For convenience, assume interatomic
    spacing to be unity, m
r = a/(2*sqrt(2));    // The atomic radius, m
V_atom = N*4/3*%pi*r^3;    // Volume of atoms, metre
     cube
V_uc = a^3;    // Volume of unit cell, metre cube
printf("\nThe percentage of actual volume occupied
    by the spheres in fcc structure = %4.2f percent",
     V_atom/V_uc*100);

// Result
// The percentage of actual volume occupied by the
    spheres in fcc structure = 74.05 percent
```

Scilab code Exa 6.9 X ray Diffraction by crystal planes

```
// Scilab Code Ex6.9 : X-ray Diffraction by crystal planes: Page-137 (2010)
// For (221) planes
h = 2; k = 2; l = 1; // Miller Indices for planes in a cubic crystal
a = 2.68e-010;     // Interatomic spacing, m
n = 1;     // First Order of diffraction
theta = 8.5;     // Glancing angle at which Bragg's reflection occurs, degrees
d = a/(h^2+k^2+l^2)^(1/2);   // The interplanar spacing for cubic crystal, m
lambda = 2*d*sind(theta);     // Bragg's Law for wavelength of X-rays, m
n = 2;     // Second order of diffraction
theta = asind(n*lambda/(2*d));     // Angle at which second order Bragg reflection occurs, degrees
printf("\nThe interplanar spacing between consecutive (221) planes = %5.3e", d);
printf("\nThe wavelength of X-rays = %5.3f angstrom", lambda/1e-010);
printf("\nThe angle at which second order Bragg reflection occurs = %4.1f degrees", theta);

// Result
// The interplanar spacing between consecutive (221) planes = 8.933e-011
// The wavelength of X-rays = 0.264 angstrom
// The angle at which second order Bragg reflection occurs = 17.2 degrees
```

Scilab code Exa 6.10 X ray Diffraction by crystal planes

```
// Scilab Code Ex6.10 : Lattice parameter for (110) planes of cubic crystal: Page-137 (2010)
h = 1; k = 1; l = 0; // Miller Indices for planes in
```

```
     a cubic crystal
n = 1;    // First Order of diffraction
theta = 25;    // Glancing angle at which Bragg's
    reflection occurs, degrees
lambda = 0.7e-010;    // Wavelength of X–rays, m
// From Bragg's Law, n*lambda = 2*d*sind(theta),
    solving for d
d = n*lambda/(2*sind(theta));    // Interplanar
    spacing of cubic crystal, m
a = d*(h^2+k^2+l^2)^(1/2);   // The lattice parameter
     for cubic crystal, m
printf("\nThe lattice parameter for cubic crystal =
    %4.2f angstrom", a/1e-010);

// Result
// The lattice parameter for cubic crystal = 1.17
    angstrom
```

Scilab code Exa 6.11 Maximum order of diffraction

```
// Scilab Code Ex6.11 : Maximum order of diffraction
    : Page−138 (2010)
d = 0.31e-009;    // Interplanar spacing, m
n = 1;    // First Order of diffraction
theta = 9.25;    // Glancing angle at which Bragg's
    reflection occurs, degrees
// From Bragg's Law, n*lambda = 2*d*sind(theta),
    solving for lambda
lambda = 2*d*sind(theta)/n;    // Wavelength of X–
    rays, m (Bragg's Law)
theta_max = 90;    // Maximum possible angle at
    which reflection can occur, degrees
n = 2*d*sind(theta_max)/lambda;    // Maximum
    possible order of diffraction
printf("\nThe Maximum possible order of diffraction
```

```
   = %1d",n);

// Result
// The Maximum possible order of diffraction = 6
```

Scilab code Exa 6.12 Bragg reflection angle for the second order diffraction

```
// Scilab Code Ex6.12 : Bragg reflection angle for
   the second order diffraction: Page-138 (2010)
// For (110) planes
h = 1, k = 1, l = 0;    // Miller indices for (110)
   planes
d_110 = 0.195e-009;    // Interplanar spacing
   between (110) planes, m
// As d_110 = a/(h^2 + k^2 + l^2)^(1/2), solving for
    a
a = d_110*(h^2 + k^2 + l^2)^(1/2);    // Lattice
   parameter for bcc crystal, m
// For (210) planes
h = 2, k = 1, l = 0;    // Miller indices for (110)
   planes
d_210 = a/(h^2 + k^2 + l^2)^(1/2);    // Interplanar
    spacing between (210) planes, m
n = 2;    // Seconds Order of diffraction
lambda = 0.072e-009;    // Wavelength of X-rays, m
// From Bragg's Law, n*lambda = 2*d_210*sind(theta),
    solving for theta
theta = asind(n*lambda/(2*d_210));    // Bragg
   reflection angle for the second order diffraction
   , degrees
printf("\nBragg reflection angle for the second
   order diffraction = %5.2f degrees", theta);

// Result
```

```
// Bragg reflection angle for the second order
    diffraction = 35.72 degrees
```

Scilab code Exa 6.13 Distance between nearest neighbours of NaCl

```
// Scilab Code Ex6.13 : Distance between nearest
    neighbours of NaCl: Page-138 (2010)
M = 23+35.5;          // Molecular weight of NaCl, kg
    per k-mole
d = 2.18e+03;     // Density of rock salt , kg per
    metre cube
n = 4;     // No. of atoms per unit cell for an fcc
    lattice of NaCl crystal
N = 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/(d*N))^(1/3);     // Lattice constant of
    unit cell of NaCl
printf("\nThe distance between nearest neighbours of
     NaCl structure = %5.3e", a/2);

// Result
// The distance between nearest neighbours of NaCl
    structure = 2.814e-010
```

Scilab code Exa 6.14 Effect of structural change on volume

```
// Scilab Code Ex6.14 : Effect of structural change
    on volume : Page-139 (2010)
// For bcc structure
r = 1.258e-010;     // Atomic radius of bcc structure
     of iron , m
```

```
a = 4*r/sqrt(3);    // Lattice parameter of bcc structure of iron, m
V = a^3;    // Volume of bcc unit cell, metre cube
N = 2;    // Number of atoms per unit cell in bcc structure
V_atom_bcc = V/N;    // Volume occupied by one atom, metre cube
// For fcc structure
r = 1.292e-010;    // Atomic radius of fcc structure of iron, m
a = 2*sqrt(2)*r;    // Lattice parameter of fcc structure of iron, m
V = a^3;    // Volume of fcc unit cell, metre cube
N = 4;    // Number of atoms per unit cell in fcc structure
V_atom_fcc = V/N;    // Volume occupied by one atom, metre cube
delta_V = (V_atom_bcc-V_atom_fcc)/V_atom_bcc*100;    // Percentage change in volume due to structural change of iron
printf("\nThe percentage change in volume of iron = %4.2f percent", delta_V);

// Result
// The percentage change in volume of iron = 0.49 percent
```

Chapter 7

SUPERCONDUCTIVITY

Scilab code Exa 7.2 Frequency of Josephson current

```
// Scilab Code Ex7.2 : Frequency of Josephson current : Page-152 (2010)
V = 1e-06;     // DC voltage applied across the Josephson junction, volt
e = 1.6e-019;     // Charge on an electron, C
h = 6.626e-034;     // Planck's constant, Js
f = 2*e*V/h;     // Frequency of Josephson current, Hz
printf("\nThe frequency of Josephson current = %5.1f MHz", f/1e+06);

// Result
// The frequency of Josephson current = 482.9 MHz
```

Scilab code Exa 7.3 Superconducting energy gap at 0K

```
// Scilab Code Ex7.3 : Superconducting energy gap at 0K : Page-152 (2010)
```

```
T_c = 0.517;     // Critical temperature for cadmium,
    K
k = 1.38e-023;     // Boltzmann constant, J/K
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
E_g = 3.5*k*T_c/e;     // Superconducting energy gap
    at absolute zero, eV
printf("\nThe superconducting energy gap for Cd at
    absolute zero = %4.2e eV",E_g);

// Result
// The superconducting energy gap for Cd at absolute
    zero = 1.56e-004 eV
```

Scilab code Exa 7.4 Wavelength of photon to break up a Cooper pair

```
// Scilab Code Ex7.4 : Wavelength of photon to break
    up a Cooper-pair: Page-152 (2010)
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
c = 3e+08;     // Speed of light in free space, m/s
h = 6.626e-034;     // Planck's constant, Js
E_g = 1.5e-004;     // Superconducting energy gap for
    a material, eV
// As E_g = h*f = h*c/lambda, solving for lambda
lambda = h*c/(E_g*e);     // Wavelength of photon to
    break up a Cooper-pair, m
printf("\nThe wavelength of photon to break up a
    Cooper-pair = %4.2e m", lambda);

// Result
// The wavelength of photon to break up a Cooper-
    pair = 8.28e-003 m
```

Scilab code Exa 7.5 Variation of London penetration depth with temperature

```
// Scilab Code Ex7.5: Variation of London penetration depth with temperature: Page-153 (2010)
lambda_0 = 37e-009;    // Penetration depth of lead at 0 kelvin, m
T_c = 7.193;    // Critical temperature of superconducting transition for lead, kelvin
T = 5.2;    // Temperature at which penetration depth for lead becomes lambda_T, kelvin
lambda_T = lambda_0*(1-(T/T_c)^4)^(-1/2);    // Penetration depth of lead at 5.2 kelvin, m
printf("\nThe penetration depth of lead at %3.1f K = %4.1f nm",T, lambda_T/1e-009);

// Result
// The penetration depth of lead at 5.2 K = 43.4 nm
```

Scilab code Exa 7.6 Isotope Effect in mercury

```
// Scilab Code Ex7.6: Isotope Effect in mercury: Page-153 (2010)
M1 = 199;    // Mass of an isotope of mercury, amu
T_C1 = 4.185;    // Transition temperature of the isoptope of Hg, K
T_C2 = 4.153;    // Transition temperature of another isoptope of Hg, K
alpha = 0.5;    // Isotope coefficient
M2 = M1*(T_C1/T_C2)^(1/alpha);    // Mass of another isotope of mercury, amu
printf("\nThe mass of another isotope of mercury at %5.3f K = %6.2f amu",T_C2, M2);

```

```
// Result
// The mass of another isotope of mercury at 4.153 K
    = 202.08 amu
```

Chapter 8

SPECIAL THEORY OF RELATIVITY

Scilab code Exa 8.1 Relativistic length contraction

```
// Scilab Code Ex8.1: Page-171 (2010)
L_0 = 1;    // For simplicity, we assume classical length to be unity, m
c = 1;    // For simplicity assume speed of light to be unity, m/s
L = (1-1/100)*L_0;    // Relativistic length, m
// Relativistic length contraction gives
// L = L_0*sqrt(1-v^2/c^2), solving for v
v = sqrt(1-(L/L_0)^2)*c;    // Speed at which relativistic length is 1 percent of the classical length, m/s
printf("\nThe speed at which relativistic length is 1 percent of the classical length = %5.3fc", v);

// Result
// The speed at which relativistic length is 1 percent of the classical length = 0.141c
```

Scilab code Exa 8.2 Time Dilation

```
// Scilab Code Ex8.2: Page-171 (2010)
c = 1;    // For simplicity assume speed of light to
     be unity, m/s
v = 0.9*c;    // Speed at which beam of particles
    travel, m/s
delta_t = 5e-006;    // Mean lifetime of particles
    as observed in the Lab. frame, s
delta_tau = delta_t*sqrt(1-(v/c)^2);    // Proper
    lifetime of particle as per Time Dilation rule, s
printf("\nThe proper lifetime of particle = %4.2e s"
    , delta_tau);

// Result
// The proper lifetime of particle = 2.18e-006 s
```

Scilab code Exa 8.4 Relativistic velocity addition

```
// Scilab Code Ex8.4: Page-172 (2010)
c = 1;    // For simplicity assume speed of light to
     be unity, m/s
v = 0.6*c;    // Speed with which the rocket leaves
    the earth, m/s
u_prime = 0.9*c;    // Relative speed of second
    rocket w.r.t. the first rocket, m/s
u = (u_prime+v)/(1+(u_prime*v)/c^2);    // Speed of
    second rocket for same direction of firing as per
     Velocity Addition Rule, m/s
printf("\nThe speed of second rocket for same
    direction of firing = %5.3fc", u);
```

```
u = (-u_prime+v)/(1-(u_prime*v)/c^2);    // Speed of second rocket for opposite direction of firing as per Velocity Addition Rule, m/s
printf("\nThe speed of second rocket for opposite direction of firing = %5.3fc", u);

// Result
// The speed of second rocket for same direction of firing = 0.974c
// The speed of second rocket for opposite direction of firing = -0.652c
```

Scilab code Exa 8.5 Relativistic effects as observed for spaceship

```
// Scilab Code Ex8.5: Page-172 (2010)
c = 1;    // For simplicity assume speed of light to be unity, m/s
L0 = 1;    // For simplicity assume length in spaceship's frame to be unity, m
L = 1/2*L0;    // Length as observed on earth, m
// Relativistic length contraction gives
// L = L_0*sqrt(1-v^2/c^2), solving for v
v = sqrt(1-(L/L0)^2)*c;    // Speed at which length of spaceship is observed as half from the earth frame, m/s
tau = 1;    // Unit time in the spaceship's frame, s
t = tau/sqrt(1-(v/c)^2);    // Time dilation of the spaceship's unit time, s
printf("\nThe speed at which length of spaceship is observed as half from the earth frame = %5.3fc", v);
printf("\nThe time dilation of the spaceship unit time = %1g*tau", t);

// Result
```

```
// The speed at which length of spaceship is
   observed as half from the earth frame = 0.866c
// The time dilation of the spaceship unit time = 2*
   tau
```

Scilab code Exa 8.6 Time difference and distance between the events

```
// Scilab Code Ex8.6: Page-172 (2010)
c = 3e+008;    // Speed of light in vacuum, m/s
v = 0.6*c;    // Velocity with which S2 frame moves
   relative to S1 frame, m/s
L_factor = 1/sqrt(1-(v/c)^2);    // Lorentz factor
t1 = 2e-007;    // Time for which first event occurs
   , s
t2 = 3e-007;    // Time for which second event
   occurs, s
x1 = 10;    // Position at which first event occurs,
    m
x2 = 40;    // Position at which second event occurs
   , m
delta_t = L_factor*(t2 - t1)+L_factor*v/c^2*(x1 - x2
   );    // Time difference between the events, s
delta_x = L_factor*(x2 - x1)-L_factor*v*(t2 - t1);
       // Distance between the events, m
printf("\nThe time difference between the events =
   %3.1e s", delta_t);
printf("\nThe distance between the events = %2d m",
   delta_x);

// Result
// The time difference between the events = 5.0e-008
    s
// The distance between the events = 15 m
```

Scilab code Exa 8.7 Speed of unstable particle in the Laboratory frame

```
// Scilab Code Ex8.7: Page-173 (2010)
c = 3e+008;    // Speed of light in vacuum, m/s
tau = 2.6e-008;    // Mean lifetime the particle in
    its own frame, s
d = 20;    // Distance which the unstable particle
    travels before decaying, m
// As t = d/v and also t = tau/sqrt(1-(v/c)^2), so
    that
// d/v = tau/sqrt(1-(v/c)^2), solving for v
v = sqrt(d^2/(tau^2+(d/c)^2));    // Speed of the
    unstable particle in Lab. frame, m/s
printf("\nThe speed of the unstable particle in Lab.
     frame = %3.1e m/s", v)

// Result
// The speed of the unstable particle in Lab. frame
    = 2.8e+008 m/s
```

Scilab code Exa 8.8 Relativistic effects applied to mu meson

```
// Scilab Code Ex8.8: Page-174 (2010)
c = 1;    // For simplicity assume speed of light to
     be unity, m/s
me = 1;    // For simplicity assume mass of electron
     to be unity, kg
tau = 2.3e-006;    // Average lifetime of mu-meson
    in rest frame, s
t = 6.9e-006;    // Average lifetime of mu-meson in
    laboratory frame, s
```

```
// Fromm Time Dilation Rule, tau = t*sqrt(1-(v/c)^2), solving for v
v = sqrt(1-(tau/t)^2)*c;    // Speed of mu-meson in the laboratory frame, m/s
c
m0 = 207*me;    // Rest mass of mu-meson, kg
m = m0/sqrt(1-(v/c)^2);    // Relativistic variation of mass with velocity, kg
me = 9.1e-031;    // Mass of an electron, kg
c = 3e+008;    // Speed of light in vacuum, m/s
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
T = (m*me*c^2 - m0*me*c^2)/e;    // Kinetic energy of mu-meson, J
printf("\nThe speed of mu-meson in the laboratory frame = %6.4fc", v);
printf("\nThe effective mass of mu-meson = %3d me", m);
printf("\nThe kinetic energy of mu-meson = %5.1f MeV", T/1e+006);

// Result
// The speed of mu-meson in the laboratory frame = 0.9428c
// The effective mass of mu-meson = 620 me
// The kinetic energy of mu-meson = 211.9 MeV
```

Scilab code Exa 8.9 Speed of moving mass

```
// Scilab Code Ex8.9: Page-174 (2010)
c = 1;    // For simplicity assume speed of light to be unity, m/s
m0 = 1;    // For simplicity assume rest mass to be unity, kg
m = (20/100+1)*m0;    // Mass in motion, kg
// As m = m0/sqrt(1-(u/c)^2), solving for u
```

```
u = sqrt(1-(m0/m)^2)*c;     // Speed of moving mass, m/s
printf("\nThe speed of moving body, u = %5.3fc", u);

// Result
// The speed of moving body, u = 0.553c
```

Scilab code Exa 8.10 Rate of decreasing mass of sun

```
// Scilab Code Ex8.10: Page-175 (2010)
c = 3e+008;     // Speed of light in vacuum, m/s
dE = 4e+026;     // Energy radiated per second my the sun, J/s
dm = dE/c^2;     // Rate of decrease of mass of sun, kg/s
printf("\nThe rate of decrease of mass of sun = %4.2e kg/s", dm);

// Result
// The rate of decrease of mass of sun = 4.44e+009 kg/s
```

Scilab code Exa 8.11 Relativistic mass energy relation

```
// Scilab Code Ex8.11: Page-175 (2010)
c = 1;     // For simplicity assume speed of light to be unity, m/s
m0 = 9.1e-031;  // Mass of the electron, kg
E0 = 0.512;     // Rest energy of electron, MeV
T = 10;     // Kinetic energy of electron, MeV
E = T + E0;     // Total energy of electron, MeV
// From Relativistic mass-energy relation
// E^2 = c^2*p^2 + m0^2*c^4, solving for p
```

```
p = sqrt(E^2-m0^2*c^4)/c;     // Momentum of the electron, MeV
// As E = E0/sqrt(1-(u/c)^2), solving for u
u = sqrt(1-(E0/E)^2)*c;     // Velocity of the electron, m/s
printf("\nThe momentum of the electron = %4.1f/c MeV", p);
printf("\nThe velocity of the electron = %6.4fc", u);

// Result
// The momentum of the electron = 10.5/c MeV
// The velocity of the electron = 0.9988c
```

Scilab code Exa 8.13 Mass from relativistic energy

```
// Scilab Code Ex8.13: Page-176 (2010)
c = 3e+008;     // Speed of light in vacuum, m/s
E = 4.5e+017;     // Total energy of object, J
px = 3.8e+008;     // X-component of momentum, kg-m/s
py = 3e+008;     // Y-component of momentum, kg-m/s
pz = 3e+008;     // Z-component of momentum, kg-m/s
p = sqrt(px^2+py^2+px^2);     // Total momentum of the object, kg-m/s
// From Relativistic mass-energy relation
// E^2 = c^2*p^2 + m0^2*c^4, solving for m0
m0 = sqrt(E^2/c^4 - p^2/c^2);     // Rest mass of the body, kg
printf("\nThe rest mass of the body = %4.2f kg", m0);

// Result
// The rest mass of the body = 4.56 kg
```

Scilab code Exa 8.14 Relativistic momentum of high speed probe

```
// Scilab Code Ex8.14: Page-176 (2010)
c = 3e+008;     // Speed of light in vacuum, m/s
m = 50000;     // Mass of high speed probe, kg
u = 0.8*c;     // Speed of the probe, m/s
p = m*u/sqrt(1-(u/c)^2);     // Momentum of the probe, kg-m/s
printf("\nThe momentum of the high speed probe = %1g kg-m/s", p);

// Result
// The momentum of the high speed probe = 2e+013 kg-m/s
```

Scilab code Exa 8.15 Moving electron subjected to the electric field

```
// Scilab Code Ex8.15: Page-177 (2010)
e = 1.6e-019;     // Electronic charge, C = Energy equivalent of 1 eV, J/eV
m0 = 9.11e-031;     // Rest mass of electron, kg
c = 3e+008;     // Speed of light in vacuum, m/s
u1 = 0.98*c;     // Inital speed of electron, m/s
u2 = 0.99*c;     // Final speed of electron, m/s
m1 = m0/sqrt(1-(u1/c)^2);     // Initial relativistic mass of electron, kg
m2 = m0/sqrt(1-(u2/c)^2);     // Final relativistic mass of electron, kg
dm = m2 - m1;     // Change in relativistic mass of the electron, kg
W = dm*c^2;     // Work done on the electron to change its velocity, J
```

```
// As W = eV, V = accelerating potential, solving for V
V = W/e;    // Accelerating potential, volt
printf("\nThe change in relativistic mass of the electron = %4.1e kg", dm);
printf("\nThe work done on the electron to change its velocity = %4.2f MeV", W/(e*1e+006));
printf("\nThe accelerating potential = %4.2e volt", V);

// Result
// The change in relativistic mass of the electron = 1.9e-030 kg
// The work done on the electron to change its velocity = 1.06 MeV
// The accelerating potential = 1.06e+006 volt
```

Chapter 9

QUANTUM MECHANICS

Scilab code Exa 9.1 De broglie wavelength of an electron from accelerating potential

```
// Scilab Code Ex9.1: De-broglie wavelength of an electron from accelerating potential : Page-202 (2010)
V = 100;    // Accelerating potential for electron, volt
lambda = sqrt(150/V)*1e-010;    // de-Broglie wavelength of electron, m
printf("\nThe De-Broglie wavelength of electron = %4.2e m", lambda);

// Result
// The De-Broglie wavelength of electron = 1.22e-010 m
```

Scilab code Exa 9.2 De broglie wavelength of an electron from kinetic energy

```
// Scilab Code Ex9.2: De-broglie wavelength of an electron from kinetic energy : Page-203 (2010)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
h = 6.626e-034;    // Planck's constant, Js
m = 9.1e-031;    // Mass of the electron, kg
Ek = 10;    // Kinetic energy of electron, eV
// Ek = p^2/(2*m), solving for p
p = sqrt(2*m*Ek*e);    // Momentum of the electron, kg-m/s
lambda = h/p ;    // de-Broglie wavelength of electron from De-Broglie relation, m
printf("\nThe de-Broglie wavelength of electron = %4.2e nm", lambda/1e-009);

// Result
// The de-Broglie wavelength of electron = 3.88e-001 nm
```

Scilab code Exa 9.4 Uncertainty principle for position and momentum

```
// Scilab Code Ex9.4: Uncertainty principle for position and momentum: Page-203 (2010)
h = 6.626e-034;    // Planck's constant, Js
m = 9.1e-031;    // Mass of the electron, kg
v = 1.1e+006;    // Speed of the electron, m/s
p = m*v;    // Momentum of the electron, kg-m/s
dp = 0.1/100*p;    // Uncertainty in momentum, kg-m/s
h_bar = h/(2*%pi);    // Reduced Planck's constant, Js
// From Heisenberg uncertainty principle,
// dx*dp = h_bar/2, solving for dx
dx = h_bar/(2*dp);    // Uncertainty in position, m
printf("\nThe uncertainty in position of electron = %4.2e m", dx);
```

```

// Result
// The uncertainty in position of electron = 5.27e-008 m
```

Scilab code Exa 9.5 Uncertainty principle for energy and time

```
// Scilab Code Ex9.5: Uncertainty principle for energy and time: Page-203 (2010)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
h = 6.626e-034;    // Planck's constant, Js
dt = 1e-008;    // Uncertainty in time, s
h_bar = h/(2*%pi);    // Reduced Planck's constant, Js
// From Heisenberg uncertainty principle,
// dE*dt = h_bar/2, solving for dE
dE = h_bar/(2*dt*e);    // Uncertainty in energy of the excited state, m
printf("\nThe uncertainty in energy of the excited state = %4.2e eV", dE);

// Result
// The uncertainty in energy of the excited state = 3.30e-008 eV
```

Scilab code Exa 9.6 Width of spectral line from Uncertainty principle

```
// Scilab Code Ex9.6: Width of spectral line from Uncertainty principle: Page-204 (2010)
c = 3e+008;    // Speed of light, m/s
dt = 1e-008;    // Average lifetime, s
lambda = 400e-009;    // Wavelength of spectral line, m
```

```
// From Heisenberg uncertainty principle,
// dE = h_bar/(2*dt) and also dE = h*c/lambda^2*d_lambda, which give
// h_bar/(2*dt) = h*c/lambda^2*d_lambda, solving for d_lambda
d_lambda = lambda^2/(4*%pi*c*dt);    // Width of spectral line, m
printf("\nThe width of spectral line = %4.2e m", d_lambda);

// Result
// The width of spectral line = 4.24e-015 m
```

Scilab code Exa 9.14 Probability of electron moving in 1D box

```
// Scilab Code Ex9.14: Probability of electron moving in 1D box : Page-207 (2010)
a = 2e-010;    // Width of 1D box, m
x1 = 0;    // Position of first extreme of the box, m
x2 = 1e-010;    // Position of second extreme of the box, m
P = integrate('2/a*(sin(2*%pi*x/a))^2', 'x', x1, x2);    // The probability of finding the electron between x = 0 and x = 1e-010
printf("\nThe probability of finding the electron between x = 0 and x = 1e-010 = %3.1f", P);

// Result
// The probability of finding the electron between x = 0 and x = 1e-010 = 0.5
```

Chapter 10

STATISTICAL MECHANICS

Scilab code Exa 10.1 Ratio of occupancy of two states

```
// Scilab Code Ex10.1: Page-222 (2010)
k = 1.38e-023;     // Boltzmann constant, J/K
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
g1 = 2;     // The degeneracy of ground state
g2 = 8;     // The degeneracy of excited state
delta_E = 10.2;     // Energy of excited state above
    the ground state, eV
T = 6000;     // Temperature of the state, K
D_ratio = g2/g1;     // Ratio of degeneracy of states
N_ratio = D_ratio*exp(-delta_E/(k*T/e));     // Ratio
     of occupancy of the excited to the ground state
printf("\nThe ratio of occupancy of the excited to
    the ground state at %d K = %4.2e", T, N_ratio);

// Result
// The ratio of occupancy of the excited to the
    ground state at 6000 K = 1.10e-008
```

Scilab code Exa 10.4 Number density and fermi energy of silver

```
// Scilab Code Ex10.4: Page-223 (2010)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
N_A = 6.023e+023;    // Avogadro's number
h = 6.626e-034;    // Planck's constant, Js
me = 9.1e-031;    // Mass of electron, kg
rho = 10.5;    // Density of silver, g per cm
m = 108;    // Molecular mass of silver, g/mol
N_D = rho*N_A/(m*1e-006);    // Number density of conduction electrons, per metre cube
E_F = h^2/(8*me)*(3/%pi*N_D)^(2/3);
printf("\nThe number density of conduction electrons = %4.2e per metre cube", N_D);
printf("\nThe Fermi energy of silver = %4.2f eV", E_F/e);

// Result
// The number density of conduction electrons = 5.86e+028 per metre cube
// The Fermi energy of silver = 5.51 eV
```

Scilab code Exa 10.5 Electronic contribution to the molar heat capacity of silver

```
// Scilab Code Ex10.5: Page-224 (2010)
N_A = 6.023e+023;    // Avogadro's number
k = 1.38e-023;    // Boltzmann constant, J/K
T = 293;    // Temperature of sodium, K
E_F = 3.24;    // Fermi energy of sodium, eV
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
C_v = %pi^2*N_A*k^2*T/(2*E_F*e);    // Molar specific heat of sodium, J/mole/K
printf("\nThe molar specific heat of sodium = %4.2f J/mole/K", C_v);

// Result
```

```
// The molar specific heat of sodium = 0.32 J/mole/K
```

Scilab code Exa 10.6 Fermi energy and mean energy of aluminium

```
// Scilab Code Ex10.6: Page-224 (2010)
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
h = 6.626e-034;    // Planck's constant, Js
m = 9.1e-031;    // Mass of the electron, kg
N_D = 18.1e+028;    // Number density of conduction
   electrons in Al, per metre cube
E_F = h^2/(8*m)*(3/%pi*N_D)^(2/3);    // Fermi
   energy of aluminium, J
Em_0 = 3/5*E_F;    // Mean energy of  the electron
   at 0K, J
printf("\nThe Fermi energy of aluminium = %5.2f eV",
    E_F/e);
printf("\nThe mean energy of  the electron at 0K =
   %4.2f eV", Em_0/e);

// Result
// The Fermi energy of aluminium = 11.70 eV
// The mean energy of  the electron at 0K = 7.02 eV
```

Chapter 11

LASERS

Scilab code Exa 11.1 Ratio of spontaneous and stimulated emission

```
// Scilab Code Ex11.1: Page-249 (2010)
h = 6.626e-034;    // Planck's constant, Js
c = 3e+08;    // Speed of light in free space, m/s
k = 1.38e-023;    // Boltzmann constant, J/K
T = 300;    // Temperature at absolute scale, K
lambda = 5500e-010;    // Wavelength of visible light, m
rate_ratio = exp(h*c/(lambda*k*T))-1;    // Ratio of spontaneous emission to stimulated emission
printf("\nThe ratio of spontaneous emission to stimulated emission for visible region = %1.0e", rate_ratio);
lambda = 1e-02;    // Wavelength of microwave, m
rate_ratio = exp(h*c/(lambda*k*T))-1;    // Ratio of spontaneous emission to stimulated emission
printf("\nThe ratio of spontaneous emission to stimulated emission for microwave region = %6.4f", rate_ratio);

// Result
// The ratio of spontaneous emission to stimulated
```

```
emission for visible region = 8e+037
// The ratio of spontaneous emission to stimulated emission for microwave region = 0.0048
```

Scilab code Exa 11.2 Energy of excited state of laser system

```
// Scilab Code Ex11.2: Page-250 (2010)
e = 1.6e-019;   // Energy equivalent of 1 eV, J/eV
h = 6.626e-034;    // Planck's constant, Js
c = 3e+08;    // Speed of light in free space, m/s
lambda = 690e-009;    // Wavelength of laser light, m
E_lower = 30.5;    // Energy of lower state, eV
E = h*c/(lambda*e);    // Energy of the laser light, eV
E_ex = E_lower + E;    // Energy of excited state of laser system, eV
printf("\nThe energy of excited state of laser system = %4.1f eV", E_ex);

// Result
// The energy of excited state of laser system = 32.3 eV
```

Scilab code Exa 11.3 Condition of equivalence of stimulated and spontaneous emission

```
// Scilab Code Ex11.3: Page-250 (2010)
h = 6.626e-034;    // Planck's constant, Js
k = 1.38e-023;    // Boltzmann constant, J/K
// Stimulated Emission = Spontaneous Emission <=> exp(h*f/(k*T))-1 = 1 i.e.
// f/T = log(2)*k/h = A
```

```
A =  log(2)*k/h;     // Frequency per unit temperature, Hz/K
printf("\nThe stimulated emission equals spontaneous emission iff f/T = %4.2e Hz/K", A);

// Result
// The stimulated emission equals spontaneous emission iff f/T = 1.44e+010 Hz/K
```

Scilab code Exa 11.4 Area and intensity of image formed by laser

```
// Scilab Code Ex11.4: Page-250 (2010)
lambda = 500e-009;     // Wavelength of laser light, m
f = 15e-02;     // Focal length of the lens, m
d = 2e-02;     // Diameter of the aperture of source, m
a = d/2;     // Radius of the aperture of source, m
P = 5e-003;     // Power of the laser, W
A = %pi*lambda^2*f^2/a^2;     // Area of the spot at the focal plane, metre square
I = P/A;     // Intensity at the focus, W per metre square
printf("\nThe area of the spot at the focal plane = %4.2e metre square", A);
printf("\nThe intensity at the focus = %4.2e watt per metre square", I);

// Result
// The area of the spot at the focal plane = 1.77e-010 metre square
// The intensity at the focus = 2.83e+007 watt per metre square
```

Scilab code Exa 11.5 Rate of energy released in a pulsed laser

```
// Scilab Code Ex11.5: Page-251 (2010)
h = 6.626e-034;    // Planck's constant, Js
c = 3e+08;    // Speed of light in free space, m/s
lambda = 1064e-009;    // Wavelength of laser light, m
P = 0.8;    // Average power output per laser pulse, W
dt = 25e-003;    // Pulse width of laser, s
E = P*dt;    // Energy released per pulse, J
N = E/(h*c/lambda);    // Number of photons in a pulse
printf("\nThe energy released per pulse = %2.0e J", E);
printf("\nThe number of photons in a pulse = %4.2e", N);

// Result
// The energy released per pulse = 2e-002 J
// The number of photons in a pulse = 1.07e+017
```

Scilab code Exa 11.6 Angular and linear spread of laser beam

```
// Scilab Code Ex11.6:Page-251 (2010)
lambda = 693e-009;    // Wavelength of laser beam, m
D = 3e-003;    // Diameter of laser beam, m
d_theta = 1.22*lambda/D;    // Angular spread of laser beam, rad
d = 300e+003;    // Height of a satellite above the surface of earth, m
```

```
a = d_theta*d;     // Diameter of the beam on the satellite, m
printf("\nThe height of a satellite above the surface of earth = %4.2e rad", d_theta);
printf("\nThe diameter of the beam on the satellite = %4.1f m", a);

// Result
// The height of a satellite above the surface of earth = 2.82e-004 rad
// The diameter of the beam on the satellite = 84.5 m
```

Chapter 12

HOLOGRAPHY AND FIBRE OPTICS

Scilab code Exa 12.1 Parameters of step index fibre

```
// Scilab Code Ex12.1: Parameters of step index
   fibre : Page-271 (2010)
n1 = 1.43;    // Refractive index of fibre core
n2 = 1.4;    // Refractive index of fibre cladding
// As sin (alpha_c) = n2/n1, solving for alpha_c
alpha_c = asind(n2/n1);    // Critical angle for
   optical fibre, degrees
// AS cos(theta_c) = n2/n1, solving for theta_c
theta_c = acosd(n2/n1);    // Critical propagation
   angle for optical fibre, degrees
NA = sqrt(n1^2 - n2^2);    // Numerical aperture for
    optical fibre
printf("\nThe critical angle for optical fibre = %5
   .2f degrees", alpha_c);
printf("\nThe critical propagation angle for optical
    fibre = %5.2f degrees", theta_c);
printf("\nNumerical aperture for optical fibre = %4
   .2f", NA);

```

```
// Result
// The critical angle for optical fibre = 78.24 degrees
// The critical propagation angle for optical fibre = 11.76 degrees
// Numerical aperture for optical fibre = 0.29
```

Scilab code Exa 12.2 Parameters of optical fibre

```
// Scilab Code Ex12.2: Parameters of optical fibre : Page-271 (2010)
n1 = 1.45;    // Refractive index of fibre core
n2 = 1.4;    // Refractive index of fibre cladding
NA = sqrt(n1^2 - n2^2);    // Numerical aperture for optical fibre
// As sin(theta_a) = sqrt(n1^2 - n2^2), solving for theta_a
theta_a = asind(sqrt(n1^2 - n2^2));    // Half of acceptance angle of optical fibre, degrees
theta_accp = 2*theta_a;    // Acceptance angle of optical fibre
Delta = (n1 - n2)/n1;    // Relative refractive index difference
printf("\nNumerical aperture for optical fibre = %5.3f", NA);
printf("\nThe acceptance angle of optical fibre = %4.1f degrees", theta_accp);
printf("\nRelative refractive index difference = %5.3f", Delta);

// Result
// Numerical aperture for optical fibre = 0.377
// The acceptance angle of optical fibre = 44.4 degrees
// Relative refractive index difference = 0.034
```

Scilab code Exa 12.3 Numerical aperture and acceptance angle of step index fibre

```
// Scilab Code Ex12.3: Numerical aperture and acceptance angle of step index fibre : Page−271 (2010)
n1 = 1.55;    // Refractive index of fibre core
n2 = 1.53;    // Refractive index of fibre cladding
n0 = 1.3;    // Refractive index of medium
NA = sqrt(n1^2 - n2^2);    // Numerical aperture for optical fibre
// n0∗sin(theta_a) = sqrt(n1ˆ2 − n2ˆ2) = NA, solving for theta_a
theta_a = asind(sqrt(n1^2 - n2^2)/n0);    // Half of acceptance angle of optical fibre, degrees
theta_accp = 2*theta_a;    // Acceptance angle of optical fibre
printf("\nNumerical aperture for step index fibre = %5.3f", NA);
printf("\nThe acceptance angle of step index fibre = %2d degrees", theta_accp);

// Result
// Numerical aperture for step index fibre = 0.248
// The acceptance angle of step index fibre = 22 degrees
```

Scilab code Exa 12.5 Output power in fibre optic communication

```
// Scilab Code Ex12.5: Output power in fibre optic communication : Page−272 (2010)
```

```
alpha = 2;     // Power loss through optical fibre, dB/km
P_in = 500;     // Poer input of optical fibre, micro-watt
z = 10;     // Length of the optical fibre, km
// As alpha = 10/z*log10(P_in/P_out), solving for P_out
P_out = P_in/10^(alpha*z/10);     // Output power in fibre optic communication, W
printf("\nThe output power in fibre optic communication = %1d micro-watt", P_out);

// Result
// The output power in fibre optic communication = 5 micro-watt
```

Chapter 13

DIELECTRIC PROPERTIES OF MATERIALS

Scilab code Exa 13.1 Electronic Polarizability of atom

```
// Scilab Code Ex13.1: Electronic Polarizability of atom : Page-287 (2010)
epsilon_0 = 8.854e-012;     // Absolute electrical permittivity of free space, farad per metre
R = 0.52e-010;     // Radius of hydrogen atom, angstrom
n = 9.7e+026;     // Number density of hydrogen, per metre cube
alpha_e = 4*%pi*epsilon_0*R^3;     // Electronic polarizability of hydrogen atom, farad-metre square
printf("\nThe electronic polarizability of hydrogen atom = %4.2e farad-metre square", alpha_e);

// Result
// The electronic polarizability of hydrogen atom = 1.56e-041 farad-metre square
```

Scilab code Exa 13.2 Parallel plate capacitor

```
// Scilab Code Ex13.2: Parallel plate capacitor: Page-287 (2010)
epsilon_0 = 8.854e-012;    // Absolute electrical permittivity of free space, farad per metre
A = 100e-004;    // Area of a plate of parallel plate capacitor, metre square
d = 1e-002;    // Distance between the plates of the capacitor, m
V = 100;    // Potential applied to the plates of the capacitor, volt
C = epsilon_0*A/d;    // Capacitance of parallel plate capacitor, farad
Q = C/V;    // Charge on the plates of the capacitor, coulomb
printf(”\nThe capacitance of parallel plate capacitor = %5.3e F”, C);
printf(”\nThe charge on the plates of the capacitor = %5.3e C”, Q);

// Result
// The capacitance of parallel plate capacitor = 8.854e-012 F
// The charge on the plates of the capacitor = 8.854e-014 C
```

Scilab code Exa 13.3 Dielectric displacement of medium

```
// Scilab Code Ex13.3: Dielectric displacement of medium: Page-288 (2010)
```

```
epsilon_0 = 8.854e-012;    // Absolute electrical
   permittivity of free space, farad per metre
epsilon_r = 5.0;    // Dielectric constant of the
   material between the plates of capacitor
V = 15;    // Potential difference applied between
   the plates of the capacitor, volt
d = 1.5e-003;    // Separation between the plates of
    the capacitor, m
// Electric displacement, D = epsilon_0*epsilon_r*E,
    as E = V/d, so
D = epsilon_0*epsilon_r*V/d;    // Dielectric
   displacement, coulomb per metre square
printf("\nThe dielectric displacement = %5.3e
   coulomb per metre square", D);

// Result
// The dielectric displacement = 4.427e-007 coulomb
   per metre square
```

Scilab code Exa 13.4 Relative dielectric constant

```
// Scilab Code Ex13.4: Relative dielectric constant
   : Page-288 (2010)
epsilon_0 = 8.854e-012;    // Absolute electrical
   permittivity of free space, farad per metre
N = 3.0e+028;    // Number density of solid
   elemental dielectric, atoms per metre cube
alpha_e = 1e-040;    // Electronic polarizability,
   farad metre square
epsilon_r = 1 + N*alpha_e/epsilon_0;    // Relative
   dielectric constant of the material
printf("\nThe Relative dielectric constant of the
   material = %5.3f", epsilon_r);

// Result
```

```
// The Relative dielectric constant of the material = 1.339
```

Scilab code Exa 13.5 Atomic polarizability of sulphur

```
// Scilab Code Ex13.5: Atomic polarizability of sulphur : Page-288 (2010)
N_A = 6.023e+023;     // Avogadro's number, per mole
epsilon_0 = 8.854e-012;     // Absolute electrical permittivity of free space, farad per metre
epsilon_r = 3.75;     // Relative dielectric constant
d = 2050;     // Density of sulphur, kg per metre cube
y = 1/3;     // Internal field constant
M = 32;     // Atomic weight of sulphur, g/mol
N = N_A*1e+03*d/M;     // Number density of atoms of sulphur, per metre cube
// Lorentz relation for local fields give
// E_local = E + P/(3*epsilon_0) which gives
// (epsilon_r - 1)/(epsilon_r + 2) = N*alpha_e/(3*epsilon_0), solving for alpha_e
alpha_e = (epsilon_r - 1)/(epsilon_r + 2)*3*epsilon_0/N;     // Electronic polarizability of sulphur, farad metre square
printf("\nThe electronic polarizability of sulphur = %5.3e farad metre square", alpha_e);

// Result
// The electronic polarizability of sulphur = 3.292e-040 farad metre square
```

Scilab code Exa 13.6 Electronic polarizability from refractive index

```
// Scilab Code Ex13.6: Electronic polarizability from refractive index : Page-289 (2010)
N = 3e+028;     // Number density of atoms of dielectric material, per metre cube
epsilon_0 = 8.854e-012;     // Absolute electrical permittivity of free space, farad per metre
n = 1.6;     // Refractive index of dielectric material
// As (n^2 - 1)/(n^2 + 2) = N*alpha_e/(3*epsilon_0), solving for alpha_e
alpha_e = (n^2 - 1)/(n^2 + 2)*3*epsilon_0/N;     // Electronic polarizability of dielectric material, farad metre square
printf("\nThe electronic polarizability of dielectric material = %4.2e farad metre square", alpha_e);

// Result
// The electronic polarizability of dielectric material = 3.03e-040 farad metre square
```

Scilab code Exa 13.7 Ratio of electronic polarizability to ionic polarizability

```
// Scilab Code Ex13.7: Ratio of electronic polarizability to ionic polarizability: Page-289 (2010)
epsilon_r = 4.9;     // Absolute relative dielectric constant of material, farad per metre
n = 1.6;     // Refractive index of dielectric material
// As (n^2 - 1)/(n^2 + 2)*(alpha_e + alpha_i)/alpha_e = N*(alpha_e + alpha_i)/(3*epsilon_0) = (epsilon_r - 1)/(epsilon_r + 2), solving for alpha_i/alpha_e
```

```
alpha_ratio = ((epsilon_r - 1)/(epsilon_r + 2)*(n^2 + 2)/(n^2 - 1) - 1)^(-1);    // Ratio of electronic polarizability to ionic polarizability
printf("\nThe ratio of electronic polarizability to ionic polarizability = %4.2f", alpha_ratio);

// Result
// The ratio of electronic polarizability to ionic polarizability = 1.53
```

Chapter 14

MAGNETIC PROPERTIES OF MATERIALS

Scilab code Exa 14.1 Spontaneous magnetisation of the substance

```
// Scilab Code Ex14.1: Spontaneous magnetisation of the substance: Page-306 (2010)
N = 6.023e+023;    // Avogadro's number. per mole
A = 56;    // Atomic weight of the substance, g/mole
d = 7.9;    // Density of the substance, gram per cm cube
m_B = 9.27e-024;    // Bohr's Magneton, joule per tesla
m = 2.2*m_B;    // Magnetic moment of substance, joule per tesla
n = d*N/A*1e+006;    // Number of atoms per unit volume of the substance, per metre cube
M = n*m;    // Spontaneous magnetisation of the substance, ampere per metre
printf("\nThe spontaneous magnetisation of the substance = %4.2e ampere per metre", M);

// Result
// The spontaneous magnetisation of the substance =
```

```
1.73e+006 ampere per metre
```

Scilab code Exa 14.2 Relative permeability of ferromagnetic material

```
// Scilab Code Ex14.2: Relative permeability of
   ferromagnetic material : Page-307 (2010)
H = 200;    // Field strength to which the
   ferromagnetic material is subjected, ampere per
   metre
M = 3100;    // Magnetisation of the ferromagnetic
   material, ampere per metre
chi = M/H;    // Magnetic susceptibility
mu_r = 1 + chi;    // Relative permeability of
   ferromagnetic material
printf("\nThe relative permeability of ferromagnetic
    material = %4.1f", mu_r);

// Result
// The relative permeability of ferromagnetic
   material = 16.5
```

Scilab code Exa 14.3 Relative permeability from magnetisation

```
// Scilab Code Ex14.3: Relative permeability from
   magnetisation : Page-307 (2010)
H = 300;    // Field strength to which the
   ferromagnetic material is subjected, ampere per
   metre
M = 4400;    // Magnetisation of the ferromagnetic
   material, ampere per metre
chi = M/H;    // Magnetic susceptibility
mu_r = 1 + chi;    // Relative permeability of
   ferromagnetic material
```

```
printf(”\nThe relative permeability of ferromagnetic material = %5.2f”, mu_r);

// Result
// The relative permeability of ferromagnetic material = 15.67
```

Scilab code Exa 14.4 Magnetic flux density and magnetisation of diamagnetic material

```
// Scilab Code Ex14.4: Magnetic flux density and magnetisation of diamagnetic material : Page−307 (2010)
mu_0 = 4*%pi*1e-07;    // Magnetic permeability of free space, tesla metre per ampere
H = 10000;    // Field strength to which the diamagnetic material is subjected, ampere per metre
chi = -0.4e-005;    // Magnetic susceptibility
M = chi*H;    // Magnetisation of the diamagnetic material, ampere per metre
B = mu_0*(H + M);    // Magnetic flux density of diamagnetic material, T
printf(”\nThe magnetisation of diamagnetic material = %4.2f ampere per metre”, M);
printf(”\nThe magnetic flux density of diamagnetic material = %6.4f T”, B);

// Result
// The magnetisation of diamagnetic material = −0.04 ampere per metre
// The Magnetic flux density of diamagnetic material = 0.0126 T
```

Scilab code Exa 14.5 Magnetisation Magnetic flux density relative permeability of diamagnetic material

```
// Scilab Code Ex14.5: Magnetisation−Magnetic flux density−relative permeability of diamagnetic material : Page−307 (2010)
mu_0 = 4*%pi*1e-07;    // Magnetic permeability of free space, tesla metre per ampere
H = 1.2e+005;    // Field strength to which the diamagnetic material is subjected, ampere per metre
chi = -4.2e-006;    // Magnetic susceptibility
M = chi*H;    // Magnetisation of the diamagnetic material, ampere per metre
B = mu_0*(H + M);    // Magnetic flux density of diamagnetic material, T
mu_r = M/H + 1;    // The relative permeability of diamagnetic material
printf("\nThe magnetisation of diamagnetic material = %5.3f ampere per metre", M);
printf("\nThe magnetic flux density of diamagnetic material = %5.3f T", B);
printf("\nThe relative permeability of diamagnetic material = %f T", mu_r);
// Result
// The magnetisation of diamagnetic material = −0.504 ampere per metre
// The magnetic flux density of diamagnetic material = 0.151 T
// The relative permeability of diamagnetic material = 0.999996 T
```

Scilab code Exa 14.6 Mean radius of body centered cubic structure

```
// Scilab Code Ex14.6: Mean radius of body centered cubic structure: Page-308 (2010)
chi = 5.6e-006;     // Magnetic susceptibility of diamagnetic material
m = 9.1e-031;     // Mass of an electron, kg
mu_0 = 4*%pi*1e-07;     // Magnetic permeability of free space, tesla metre per ampere
Z = 1;     /// Atomic number
e = 1.6e-019;     // Electronic charge, C
a = 2.53e-010;     // Lattice parameter of bcc structure, m
N = 2/a^3;     // The number of electrons per unit volume, per metre cube
r = sqrt(chi*6*m/(mu_0*Z*e^2*N));     // Mean radius of body centered cubic structure as per Langevin relation for Diamagnetic susceptibility, m
printf("\nThe mean radius of body centered cubic structure = %5.3e angstrom", r/1e-010);

// Result
// The mean radius of body centered cubic structure = 8.773e-001 angstrom
```

Scilab code Exa 14.7 Susceptibility and magnetisation of paramagnetic salt

```
// Scilab Code Ex14.7: Susceptibility and magnetisation of paramagnetic salt: Page-308 (2010)
mu_0 = 4*%pi*1e-07;     // Magnetic permeability of free space, tesla metre per ampere
N_A = 6.02e+026;     // Avogadro's number, per kmol
rho = 4370;     // Density of paramegnetic salt, kg
```

```
per metre cube
M = 168.5;    // Molecular weight of paramagnetic salt, g/mol
T = 27+273;    // Temperature of paramagnetic salt, K
H = 2e+005;    // Field strength to which the paramagnetic salt is subjected, ampere per metre
mu_B = 9.27e-024;    // Bohr's magneton, ampere metre square
p = 2;    // Number of Bohr magnetons per molecule
k = 1.38e-023;    // Boltzmann constant, J/K
N = rho*N_A/M;    // Total density of atoms in the paramagnetic salt, per metr cube
chi = mu_0*N*p^2*mu_B^2/(3*k*T);    // Magnetic susceptibility of paramagnetic salt
M = chi*H;    // Magnetisation of paramagnetic salt, ampere per metre
printf("\nThe magnetic susceptibility of paramagnetic salt = %4.2e per metre", chi);
printf("\nThe magnetisation of paramagnetic salt = %4.2e ampere per metre", M);

// Result
// The magnetic susceptibility of paramagnetic salt = 5.43e-004 per metre
// The magnetisation of paramagnetic salt = 1.09e+002 ampere per metre
```

Chapter 15

THERMAL PROPERTIES

Scilab code Exa 15.1 Debye temperature of aluminium

```
// Scilab Code Ex15.1: Page-323 (2010)
k = 1.38e-023;     // Boltzmann constant, J/K
h = 6.626e-034;     // Planck's constant, Js
f_D = 64e+011;     // Debye frequency for Al, Hz
theta_D = h*f_D/k;     // Debye temperature, K
printf("\nThe Debye temperature of aluminium = %5.1f
    K", theta_D);

// Result
// The Debye temperature of aluminium = 307.3 K
```

Scilab code Exa 15.2 Lattice specific heat of carbon

```
// Scilab Code Ex15.2: Page-323 (2010)
N = 6.02e+026;     // Avogadro's number, per kmol
k = 1.38e-023;     // Boltzmann constant, J/K
h = 6.626e-034;     // Planck's constant, Js
f_D = 40.5e+012;     // Debye frequency for Al, Hz
```

```
T = 30;     // Temperature of carbon, Ks
theta_D = h*f_D/k;     // Debye temperature, K
C_l = 12/5*%pi^4*N*k*(T/theta_D)^3;     // Lattice
    specific heat of carbon, J/k-mol/K
printf("\nThe lattice specific heat of carbon = %4.2
    f J/k-mol/K", C_l);

// Result
// The lattice specific heat of carbon = 7.13 J/k-
    mol/K
```

Scilab code Exa 15.3 Einstein frequency for Cu

```
// Scilab Code Ex15.3: Page-323 (2010)
k = 1.38e-023;     // Boltzmann constant, J/K
h = 6.626e-034;     // Planck's constant, Js
theta_E = 1990;     // Einstein temperature of Cu, K
f_E = k*theta_E/h;     // Einstein frequency for Cu,
    K
printf("\nThe Einstein frequency for Cu = %4.2e Hz",
    f_E);
printf("\nThe frequency falls in the near infrared
    region");

// Result
// The Einstein frequency for Cu = 4.14e+013 Hz
// The frequency falls in the near infrared region
```

Scilab code Exa 15.4 Electronic and lattice heat capacities for Cu

```
// Scilab Code Ex15.4: Page-323 (2010)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
N = 6.02e+023;     // Avogadro's number, per mol
```

```
T = 0.05;    // Temperature of Cu, K
E_F = 7;    // Fermi energy of Cu, eV
k = 1.38e-023;    // Boltzmann constant , J/K
h = 6.626e-034;    // Planck's constant , Js
theta_D = 348;    // Debye temperature of Cu, K
C_e = %pi^2*N*k^2*T/(2*E_F*e);    // Electronic heat capacity of Cu, J/mol/K
C_V = 12/5*%pi^4*N*k*(T/theta_D)^3; // Lattice heat capacity of Cu, J/mol/K
printf("\nThe electronic heat capacity of Cu = %4.2e J/mol/K", C_e);
printf("\nThe lattice heat capacity of Cu = %4.2e J/mol/K", C_V);

// Result
// The electronic heat capacity of Cu = 2.53e-005 J/mol/K
// The lattice heat capacity of Cu = 5.76e-009 J/mol/K
```

Scilab code Exa 15.5 Einstein lattice specific heat

```
// Scilab Code Ex15.5: Page-324 (2010)
T = 1;    // For simplicity assume temperature to be unity , K
R = 1;    // For simplicity assume molar gas constant to be unity , J/mol/K
theta_E = T;    // Einstein temperature , K
C_V = 3*R*(theta_E/T)^2*exp(theta_E/T)/(exp(theta_E/T)-1)^2;    // Einstein lattice specific heat , J/mol/K
printf("\nThe Einstein lattice specific heat , C_v = %4.2f X 3R", C_V/3);

// Result
```

```
// The Einstein lattice specific heat, C_v = 0.92 X 3R
```

Scilab code Exa 15.6 Molar electronic heat capacity of zinc

```
// Scilab Code Ex15.6: Page-324 (2010)
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
v = 2;     // Valency of Zn atom
N = v*6.02e+023;     // Avogadro's number, per mol
T = 300;     // Temperature of Zn, K
E_F = 9.38;     // Fermi energy of Zn, eV
k = 1.38e-023;     // Boltzmann constant, J/K
h = 6.626e-034;     // Planck's constant, Js
C_e = %pi^2*N*k^2*T/(2*E_F*e);     // Electronic heat capacity of Zn, J/mol/K
printf("\nThe molar electronic heat capacity of zinc = %5.3f J/mol/K", C_e);

// Result
// The molar electronic heat capacity of zinc = 0.226 J/mol/K
```

Chapter 17

ULTRASONICS

Scilab code Exa 17.1 Thickness of vibrating quartz at resonance

```
// Scilab Code Ex17.1: Thickness of vibrating quartz
    at resonance : Page-352 (2010)
f = 3e+006;    // Fundamental vibrational frequency
    of quartz crystal, MHz
Y = 7.9e+010;    // Young's modulus of quartz,
    newton per metre
rho = 2650;    // Density of quartz, kg per metre
    cube
// We have for resonant frequency
// f = 1/(2*l)*sqrt(Y/rho), solving for l
l = 1/(2*f)*sqrt(Y/rho);    // Thickness of
    vibrating quartz at resonance, m
printf("\nThe thickness of vibrating quartz at
    resonance = %3.1f mm", l/1e-003);

// Result
// The thickness of vibrating quartz at resonance =
    0.9 mm
```

Chapter 18

ACOUSTICS OF BUILDINGS

Scilab code Exa 18.1 Output power of the sound source

```
// Scilab Code Ex18.1: Output power of the sound source : Page-361 (2010)
r = 200;    // Distance of the point of reduction from the source, m
I_0 = 1e-012;    // Final intensity of sound, watt per metre square
I_f = 60;        // Intensity gain of sound at the point of reduction, dB
// As A_I = 10*log10(I/I_0), solving for I
I = I_0*10^(I_f/10);    // Initial Intensity of sound, watt per metre square
P = 4*%pi*r^2*I;    // Output power of the sound source, watt
printf("\nThe output power of the sound source = %3.1f W", P);

// Result
// The output power of the sound source = 0.5 W
```

Scilab code Exa 18.2 Change in sound level for doubling intensity

```
// Scilab Code Ex18.2: Change in sound level for doubling intensity: Page-361 (2010)
I1 = 1;    // For simplicity assume first intensity level to be unity, W per metre square
I2 = 2*I1;    // Intensity level after doubling, watt per metre square
dA_I = 10*log10(I2/I1);    // Difference in gain level, dB
printf("\nThe sound intensity level is increased by = %1d dB", dA_I);

// Result
// The sound intensity level is increased by = 3 dB
```

Scilab code Exa 18.3 Total absorption of sound in the hall

```
// Scilab Code Ex18.3: Total absorption of sound in the hall: Page-361 (2010)
V = 8000;    // Volume of the hall, metre cube
T = 1.5;    // Reverbration time of the hall, s
alpha_s = 0.167*V/T;    // Sabine Formula giving total absorption of sound in the hall, OWU
printf("\nThe total absorption of sound in the hall = %5.1f OWU", alpha_s);

// Result
// The total absorption in the hall = 890.7 OWU
```

Scilab code Exa 18.4 Average absorption coefficient of the surfaces of the hall

```
// Scilab Code Ex18.4: Average absorption coefficient of the surfaces of the hall: Page-362 (2010)
V = 25*20*8;        // Volume of the hall, metre cube
S = 2*(25*20+25*8+20*8);     // Total surface area of the hall, metre square
T = 4;     // Reverbration time of the hall, s
alpha = 0.167*V/(T*S);     // Sabine Formule giving total absorption in the hall, OWU
printf("\nThe total absorption in the hall = %5.3f OWU per metre square", alpha);

// Result
// The total absorption in the hall = 0.097 OWU per metre square
```

Scilab code Exa 18.5 Reverbration time for the hall

```
// Scilab Code Ex18.5: Reverbration time for the hall : Page-362 (2010)
V = 475;        // Volume of the hall, metre cube
s = [200, 100, 100];     // Area of wall, floor and ceiling of the hall resp., metre square
T = 4;     // Reverbration time of the hall, s
alpha = [0.025, 0.02, 0.55];     // Absorption coefficients of the wall, ceiling and floor resp., OWU per metre square
alpha_s = 0;
for i=1:1:3
    alpha_s = alpha_s + alpha(i)*s(i);
end
T = 0.167*V/alpha_s;     // Sabine Formula for reverbration time, s
printf("\nThe reverbration time for the hall = %4.2f s", T);
```

```

// Result
// The reverberation time for the hall = 1.28 s
```

Scilab code Exa 18.6 Gain of resultant sound intensity

```
// Scilab Code Ex18.6: Gain of resultant sound intensity: Page-362 (2010)
I0 = 1;    // For simplicity assume initial sound intensity to be unity, watt per metre square
A_I1 = 80;    // First intensity gain of sound, dB
A_I2 = 70;    // Second intensity gain of sound, dB
// As A_I = 10*log10(I/I_0), solving for I1 and I2
I1 = 10^(A_I1/10)*I0;    // First intensity of sound, watt per metre square
I2 = 10^(A_I2/10)*I0;    // Second intensity of sound, watt per metre square
I = I1 + I2;    // Resultant intensity level of sound, watt per metre square
A_I = 10*log10(I/I0);    // Intensity gain of resultant sound, dB
printf("\nThe intensity gain of resultant sound = %6.3f dB", A_I);

// Result
// The intensity gain of resultant sound = 80.414 dB
```