Scilab Textbook Companion for Atomic And Nuclear Physics by N. Subrahmanyam, B. Lal And J. Seshan[1]

Created by
Palvi Gupta
Atomic and Nuclear Physics
Physics
shri mata vaishno devi university
College Teacher
Dr. Kamni
Cross-Checked by
Lavitha Pereira

July 31, 2019

[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: Atomic And Nuclear Physics

Author: N. Subrahmanyam, B. Lal And J. Seshan

Publisher: S. Chand And Company Ltd., New Delhi

Edition: 10

Year: 2008

ISBN: 81-219-0414-5

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

Relativity

Scilab code Exa 1.1 Relative Speed of Approach

```
// Scilab Code Ex1.1 Relative Speed of approach: Pg:20 (2008)
c = 1;     // For the sake of simplicity, assume c = 1, m/s
u = 0.87*c;     // Velocity of approach of spaceship A towards spaceship B, m/s
v = -0.63*c;     // Velocity of approach of spaceship B towards spaceship A, m/s
V = (u - v)/(1 - (u*v)/c^2);     // Velocity Addition Rule giving relative speed of approach of particles, m/s
printf("\nThe relative speed of approach of particles = %6.4fc", V);
// Result
// The relative speed of approach of particles = 0.9689c
```

Scilab code Exa 1.2 Relative Speed of Spaceships

```
// Scilab Code Ex1.2 Relative Speed of spaceships: Pg: 20 (2008)
c = 1;    // For the sake of simplicity, assume c = 1, m/s
u = 0.9*c;    // Velocity of approach of spaceship A towards spaceship B, m/s
v = -0.9*c;    // Velocity of approach of spaceship B towards spaceship A, m/s
V = (u - v)/(1 - (u*v)/c^2);    // Velocity Addition Rule giving relative speed of approach of spaceships, m/s
printf("\nThe relative speed of B w.r.t. A = %5.3fc", V);
// Result
// The relative speed of B w.r.t. A = 0.994c
```

Scilab code Exa 1.3 Relativistic Length Contraction

```
// Scilab Code Ex1.3 Relativistic length contraction: Pg: 20 (2008)
L0 = 1;    // Actual length of the metre stick, m
rel_mass = 3/2;    // Relative mass of stick w.r.t. rest its mass
// As m = m0/sqrt(1 - (v/c)^2) and L = L0*sqrt(1 - (v/c)^2)
// Thus L/m = (L0/m0)*(1 - (v/c)^2), solving for L
// L = (m0/m)*L0 i.e.
L = 1/rel_mass*L0;    // Apparent length of the metre rod, m
printf("\nThe apparent length of the metre rod = %5.3f m", L);
// Result
// The apparent length of the metre rod = 0.667 m
```

Scilab code Exa 1.5 Mass Energy Equivalence

```
// Scilab Code Ex1.5 Mass-Energy Equivalence: Pg: 22 (2008)
U = 7.5e+011;     // Total electrical energy generated in a country, kWh
kWh = 1000*3600;     // Conversion factor for kilowatt-hour into joule, J/kWh
c = 3e+08;     // Speed of light, m/s
m = (U*kWh)/c^2;     // Mass equivalent of energy, kg
printf("\nThe mass converted into energy = %2d kg", m);
// Result
// The mass converted into energy = 30 kg
```

Scilab code Exa 1.6 Energy Equivalent of Mass

```
// Scilab Code Ex1.6 Energy equivalent of mass: Pg:22 (2008)
m = 1;     // Mass of a substance, kg
c = 3e+08;     // Speed of light, m/s
U = m*c^2;     // Energy equivalent of mass, J
printf("\nThe energy equivalent of mass = %1.0e J", U);
// Result
// The energy equivalent of mass = 9e+016 J
```

Scilab code Exa 1.7 Relativistic Variation of Mass with Speed

```
// Scilab Code Ex1.7 Relativistic variation of mass with speed: Pg: 22 (2008)
m0 = 1e-024;    // Mass of a particle, kg
v = 1.8e+08;    // Speed of the particle, m/s
c = 3e+08;    // Speed of light, m/s
m = m0/sqrt(1-(v/c)^2);    // Mass of the moving particle, kg
printf("\nThe mass of moving particle = %4.2e kg", m);
// Result
// The mass of moving particle = 1.25e-024 kg
```

Scilab code Exa 1.8 Increase in Mass of Water

```
// Scilab Code Ex1.8 Increase in mass of water: Pg: 23 (2008)
c = 3e+08;    // Speed of light, m/s
T1 = 273;    // Initial temperature of water, K
T2 = 373;    // Final temperature of water, K
M = 1e+06;    // Mass of water, kg
C = 1e+03;    // Specific heat of water, cal/kg-K
J = 4.18;    // Joule's mechanical equivalent of heat, cal/joule
U = M*C*(T2 - T1)*J;    // Increase in energy of water, J
m = U/c^2;    // Increase in mass of water, kg
printf("\nThe increase in mass of water = %4.2e kg", m);
// Result
// The increase in mass of water = 4.64e-006 kg
```

Scilab code Exa 1.9 Ratio of Rest Mass and Mass in Motion

```
// Scilab Code Ex1.9 Ratio of rest mass and mass in motion: Pg:23 (2008)
c = 1;    // For convenience, speed of light is assumed to be unity, m/s
v = 0.5*c;    // Velocity of moving particle, m/s
// As m0 = m*sqrt(1 - (v/c)^2), and m0/m = rel_mass, we have
rel_mass = sqrt(1 - (v/c)^2);    // Ratio of rest mass and the moving mass
printf("\nThe ratio of rest mass and the mass in motion = %6.4f kg", rel_mass);
// Result
// The ratio of rest mass and the mass in motion = 0.8660 kg
```

Scilab code Exa 1.10 Heat Equivalent of Mass

```
// Scilab Code Ex1.10 Heat equivalent of mass: Pg:23 (2008)
c = 3e+08;    // Speed of light, m/s
J = 4.18;    // Joule's equivalent of heat, joule per calorie
m = 4.18e-03;    // Mass of the substance, kg
U = m*c^2;    // Energy equivalent of mass, J
Q = U/J;    // Heat equivalent of mass, calorie
printf("\nThe heat equivalent of mass = %1.0e cal", Q);
// Result
// The heat equivalent of mass = 9e+013 cal
```

Scilab code Exa 1.11 Variation of Space and Time

```
// Scilab Code Ex1.11 Variation of space and time: Pg: 23 (2008)
L = 0.5;    // Shortened length of the rod, m
L0 = 1;    // Actual length of the rod, m
t0 = 1;    // Actual time on the spaceship, s
c = 3e+08;    // Speed of light, m/s
v = sqrt(1 - (L/L0)^2)*c;    // Speed of the spaceship, m/s
t = t0/sqrt(1 - (v/c)^2);        // Dilated time for stationary observer, s
printf("\nThe speed of light = %5.3e m/s", v);
printf("\nThe time dilation corresponding to 1 s on the spaceship = %d s", round(t));
// Result
// The speed of light = 2.598e+008 m/s
// The time dilation corresponding to 1 s on the spaceship = 2 s
```

Scilab code Exa 1.12 `Mean Lifetime of a Moving Meason`

```
// Scilab Code Ex1.12 Mean lifetime of a moving meason: Pg: 24 (2008)
c = 1;    // For convenience, speed of light is assumed to be unity
t0 = 2e-08;    // Mean life time of pi-meson at rest, s
v = 0.8*c;    // Velocity of moving pi-meason, m/s
t = t0/sqrt(1-(v/c)^2);    // Mean lifetime of moving pi-meason, s
printf("\nThe mean lifetime of moving meason = %4.2e s", t);
// Result
// The mean lifetime of moving meason = 3.33e-008 s
```

Scilab code Exa 1.13 Velocity of One Atomic Mass Unit

```
// Scilab Code Ex1.13 Velocity of one atomic mass unit: Pg: 24 (2008)
c = 1;      // For convenience, speed of light is assumed to be unity, m/s
m0 = 1;      // For convenience, rest mass is assumed to be unity
// Here 2*m0*c^2 = m*c^2 - m0*c^2 = KE which gives
m = 3*m0;      // Atomic mass in motion, kg
// As m = m0/sqrt(1 - (v/c)^2), solving for v
v = sqrt(1 - (m0/m)^2)*c;      // Velocity of one atomic mass, m/s
printf("\nThe velocity of one atomic mass = %5.3fc", v);
// Result
// The velocity of one atomic mass = 0.943c
```

Scilab code Exa 1.14 Speed of an Electron for an Equivalent Proton Mass

```
// Scilab Code Ex1.14 Speed of an electron for an equivalent proton mass: Pg: 25 (2008)
c = 3e+08;      // Speed of light, m/s
m0 = 1;      // For convenience, rest mass of an electron is assumed to be unity
m = 2000*m0;      // Rest mass of a proton, units
// As m = m0/sqrt(1 - (v/c)^2), solving for v
v = sqrt(1 - (m0/m)^2)*c;      // Speed of the moving electron, m/s
printf("\nThe speed of the moving electron = %4.2e m/s (approx.)", v);
// Result
```

```
// The speed of the moving electron = 3.00e+008 m/s (approx.)
```

Scilab code Exa 1.15 Speed at Total Energy Twice the Rest Mass Energy

```
// Scilab Code Ex1.15 Speed at total energy twice the rest mass energy: Pg: 25 (2008)
c = 1;      // Speed of light is assumed to be unity, m/s
m0 = 1;     // For convenience, rest mass of the particle is assumed to be unity, kg
m = 2*m0;     // Mass of the moving particle when m*c^2 = 2*m0*c^2, kg
// As m = m0/sqrt(1 - (v/c)^2), solving for v
v = sqrt(1 - (m0/m)^2)*c;     // Speed of the moving particle, m/s
printf("\nThe speed of the moving  particle = %5.3fc", v);
// Result
// The speed of the moving  particle = 0.866c
```

Scilab code Exa 1.16 Relative Velocity and Mass

```
// Scilab Code Ex1.16 Relative velocity and mass: Pg:26 (2008)
c = 3e+08;     // Speed of light, m/s
u = 2e+08;     // Speed of first particle, m/s
v = -2e+08;     // Speed of second particle, m/s
u_prime = (u - v)/(1 - u*v/c^2);     // Velocity addition rule giving relative velocity, m/s
m0 = 3e-025;     // Rest mass of each particle, kg
m = m0/sqrt(1 - (u_prime/c)^2);     // Mass of one particle relative to the other, kg
```

```
printf("\nThe relative speed of one particle w.r.t
    the other = %5.3e m/s", u_prime);
printf("\nThe mass of one particle relative to the
    other = %3.1e kg", m);
// Result
// The relative speed of one particle w.r.t the
    other = 2.769e+008 m/s
// The mass of one particle relative to the other =
    7.8e-025 kg
```

Scilab code Exa 1.17 `Relativistic Variation of density with Velocity`

```
// Scilab Code Ex1.17 Relativistic variation of
    density with velocity: Pg: 26 (2008)
c = 1;    // Speed of light is assumed to be unity
    for convenience, m/s
v = 0.9*c;    // Speed of moving frame, m/s
rho_0 = 19.3e+03;    // Density of gold in rest
    frame, kg metre per cube
L0 = 1;    // Actual length is assumed to be unity,
    m
m0 = 1;    // Rest mass of gold is assumed to be
    unity, kg
V0 = m0/rho_0;    // Volume of gold in rest frame,
    metre cube
L = L0*sqrt(1 - (v/c)^2);    // Relativistic Length
    Contraction Formula, m
y = 1;    // Width of gold block is assumed to be
    unity, m
z = 1;    // Height of gold block is assumed to be
    unity, m
V = L*y*z*V0;    // Volume of gold as observed from
    moving frame, metre cube
m = m0/sqrt(1 - (v/c)^2);    // Mass of gold as
    observed from moving frame, kg
```

```
rho = m/V;     // Density of gold as observed from moving frame, kg per metre cube
printf("\nThe density of gold as observed from moving frame = %5.1fe+003 kg per metre cube", rho/1e+03);
// Result
// The density of gold as observed from moving frame = 101.6e+003 kg per metre cube
```

Scilab code Exa 1.18 Electrons Accelerated to Relativistic Speeds

```
// Scilab Code Ex1.18 Electrons accelerated to relativistic speeds: Pg: 27 (2008)
U = 1e+09*1.6e-019;     // Kinetic energy of the electrons, J
c = 3e+08;     // Speed of light, m/s
// As U = m*c^2, solving for m
m = U/c^2;     // Mass of moving electrons, kg
m0 = 9.1e-031;     // Rest mass of an electron, kg
mass_ratio = m/m0;     // Ratio of a moving electron mass to its rest mass
// As m = m0/sqrt(1 - (v/c)^2), Relativistic mass of electron, kg, solving for v, we have
v = sqrt(1 - (m0/m)^2)*c;     // Velocity of moving electron, m/s
vel_ratio = v/c;     // Ratio of electron velocity to the velocity of light
U0 = m0*c^2;     // Rest mass energy of electron, J
ene_ratio = U/U0;     // Ratio of electron energy to its rest mass energy
printf("\nThe ratio of a moving electron mass to its rest mass %4.2e", mass_ratio);
printf("\nThe ratio of electron velocity to the velocity of light = 1 - %5.3e", (1-vel_ratio^2)/2);
```

```
printf("\nThe ratio of electron energy to its rest mass energy = %5.3e", ene_ratio);
// Result
// The ratio of a moving electron mass to its rest mass 1.95e+003
// The ratio of electron velocity to the velocity of light = 1 - 1.310e-007
// The ratio of electron energy to its rest mass energy = 1.954e+003
```

Scilab code Exa 1.19 Electron Speed Equivalent of Twice its Rest Mass

```
// Scilab Code Ex1.19 Electron speed equivalent of twice its rest mass: Pg: 28 (2008)
m0 = 9.1e-031;     // Rest mass of an electron, kg
m = 2*m0;     // Mass of moving electron, kg
c = 3e+08;     // Speed of light, m/s
// As m = m0/sqrt(1 - (v/c)^2), Relativistic mass of electron, kg, solving for v, we have
v = sqrt(1 - (m0/m)^2)*c;     // Velocity of moving electron, m/s
printf("\nThe speed of electron so that its mass becomes twice its rest mass = %5.3e m/s", v);
// Result
// The speed of electron so that its mass becomes twice its rest mass = 2.598e+008 m/s
```

Scilab code Exa 1.20 Electron Speed Equivalent of Twice its Rest Mass

```
// Scilab Code Ex1.20 Electron speed equivalent of twice its rest mass: Pg: 28 (2008)
m0 = 9.1e-031;     // Rest mass of an electron, kg
m = 2*m0;     // Mass of moving electron, kg
```

```
c = 3e+08;      // Speed of light , m/s
// As m = m0/sqrt(1 − (v/c)ˆ2), Relativistic mass of
     electron , kg, solving for v, we have
v = sqrt(1 - (m0/m)^2)*c;     // Velocity of moving
    electron , m/s
printf("\nThe speed of electron so that its mass
    becomes twice its rest mass = %5.3e m/s", v);
// Result
// The speed of electron so that its mass becomes
    twice its rest mass = 2.598e+008 m/s
```

Scilab code Exa 1.21 Fractional Speed of Electron

```
// Scilab Code Ex1.21 Fractional speed of electron:
    Pg:29 (2008)
m0 = 9.1e-031;      // Rest mass of an electron , kg
c = 3e+08;      // Speed of light , m/s
E = 0.5*1e+06*1.6e-019;      // Kinetic energy of
    electron , J
// As E = (m − m0)∗cˆ2, solving for m
m = E/c^2+m0;      // Mass of moving electron , kg
// As m = m0/sqrt(1 − (v/c)ˆ2), Relativistic mass of
     electron , kg, solving for v, we have
v = sqrt(1 - (m0/m)^2)*c;     // Velocity of moving
    electron , m/s
printf("\nThe speed of electron relative to speed of
     light = %5.3f", v/c);
// Result
// The speed of electron relative to speed of light
    = 0.863
```

Scilab code Exa 1.22 Effective Mass and Speed of Electron

```
// Scilab Code Ex1.22 Effective mass and speed of electron: Pg: 29 (2008)
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Electron-volt equivalent of 1 joule, eV/joule
U = 2*1e+06*e;     // Total energy of electron, J
// As E = (m - m0)*c^2, solving for m
m = U/c^2;     // Effective mass of electron, kg
m0 = 0.511*1e+06*e/c^2;     // Rest mass of the electron, kg
// As m = m0/sqrt(1 - (v/c)^2), Relativistic mass of electron, kg, solving for v, we have
v = sqrt(1 - (m0/m)^2)*c;     // Velocity of moving electron, m/s
printf("\nThe effective mass of electron = %4.1e kg", m);
printf("\nThe relativistic speed of electron = %4.2fc m", v/c);
// Result
// The effective mass of electron = 3.6e-030 kg
// The relativistic speed of electron = 0.97c m
```

Scilab code Exa 1.23 Energy Released in Fission

```
// Scilab Code Ex1.23 Energy released in fission: Pg: 30 (2008)
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Charge on an electron, coulomb
r0 = 1.2e-015;     // Equilibrium nuclear radius, m
A = 238;     // Twice the mass of each fragment
q1 = 46*e;     // Charge on first fragment, coulomb
q2 = 46*e;     // Charge on second fragment, coulomb
R = r0*(A/2)^(1/3);
d = 2*R;     // Distance between two fragments, m
U = q1*q2*9e+09/d;     // Energy released in fission,
```

```
    J
printf("\nThe energy released in fission of U
    (92,238) = %3d MeV", U/(e*1e+06));
// Result
// The energy released in fission of U(92,238) = 258
    MeV
```

Scilab code Exa 1.24 Relativistic Speed Form Relativistic Mass

```
// Scilab Code Ex1.24 Relativistic speed form
    relativistic mass: Pg: 30 (2008)
c = 3e+08;    // Speed of light, m/s
m0 = 1/2;    // Rest mass of the particle, MeV/c^2
m = 1/sqrt(2);    // Relativistic mass of the
    particle, MeV/c^2
// As m = m0/sqrt(1 - (v/c)^2), Relativistic mass of
    electron, kg, solving for v, we have
v = sqrt(1 - (m0/m)^2)*c;    // Relativistic
    velocity of particle, m/s
printf("\nThe relativistic velocity of particle = %4
    .2e m/s", v);
// Result
// The relativistic velocity of particle = 2.12e+008
    m/s
```

Scilab code Exa 1.25 Decay of muon

```
// Scilab Code Ex1.25 Decay of muon: Pg: 31 (2008)
c = 3e+08;    // Speed of light, m/s
v = 0.992*c;    // Relativistic speed of muon, m/s
S = 60*1e+03;    // Distance travelled by muon
    before it decays, m
```

```
t_prime = S/v;    // Time measured by observer on earth (Dilated Time), s
t = t_prime*sqrt(1 - (v/c)^2);    // Time measured by muon in its own frame, s
s = v*t;    // Distance covered by the muon in its own frame of reference, m
printf("\nThe time measured by observer on earth (Dilated Time) = %5.3e s", t_prime);
printf("\nThe time measured by muon in its own frame = %4.2e s", t);
printf("\nThe distance covered by the muon in its own frame of reference = %4.2f km", s/1e+03);
// Result
// The time measured by observer on earth (Dilated Time) = 2.016e-004 s
// The time measured by muon in its own frame = 2.55e-005 s
// The distance covered by the muon in its own frame of reference = 7.57 km
```

Scilab code Exa 1.26 Decay of Unstable Particle

```
// Scilab Code Ex1.26 Decay of unstable particle: Pg: 31 (2008)
c = 3e+08;    // Speed of light, m/s
v = 0.9*c;    // Relativistic speed of unstable particle, m/s
t0 = 1e-06;    // Time of decay of unstable particle in rest frame, s
t = t0/sqrt(1 - (v/c)^2);    //Time of decay of unstable particle in moving frame, s
s = v*t;    // Distance travelled by unstable particle before it decays in moving frame, m
printf("\nThe distance travelled before the unstable particle decays = %4.2e m", s);
```

```
// Result
// The distance travelled before the unstable
   particle decays = 6.19e+002 m
```

Chapter 2

Quantum Mechanicsq

Scilab code Exa 2.1 Threshold Wavelength of Tungsten

```
// Scilab Code Ex2.1 Threshold wavelength of
   tungsten: Pg:4 (2008)
phi = 4.5*1.6e-019;     // Work function for
   tungesten, joule
h = 6.6e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
// As phi = h*c/L0, solving for L0
L0 = h*c/phi;     // Threshold wavelength of
   tungesten, m
printf("\nThe threshold wavelength of tungesten =
   %4d angstrom", L0/1e-010);
// Result
// The threshold wavelength of tungesten = 2750
   angstrom
```

Scilab code Exa 2.2 Maximum Velocity of Photoelectrons

```
// Scilab Code Ex2.2 Maximum velocity of
   photoelectrons: Pg:44 (2008)
```

```
phi = 4*1.6e-019;     // Work function for photoelectric surface, joule
h = 6.6e-034;     // Planck's constant, Js
e = 1.6e-019;     // Electronic charge, coulomb
m = 9.1e-031;     // Mass of the electron, kg
f = 1e+15;     // Frequency of incident photons, Hz
c = 3e+08;     // Speed of light, m/s
// KE = 1/2*m*v^2 = h*f - phi, solving for v, we have
v = sqrt(2*(h*f - phi)/m);     // Maximum velocity of photoelectrons, m/s
printf("\nThe maximum velocity of photoelectrons = %5.3e m/s", v);
// Result
// The maximum velocity of photoelectrons = 2.097e+005 m/s
```

Scilab code Exa 2.3 Energy of Photoelectrons

```
// Scilab Code Ex2.3 Energy of photoelectrons: Pg:45 (2008)
h = 6.6e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Energy equivalent of 1 joule, joule/eV
L = 1800e-010;     // Wavelength of incident light, m
L0 = 2300e-010;     // Threshold wavelength of tungsten, m
E = h*c*(1/L - 1/L0);     // Energy of photoelectrons emitted from tungsten, joule
printf("\nThe energy of photoelectrons emitted from tungsten = %3.1f eV", E/e);
// Result
// The energy of photoelectrons emitted from tungsten = 1.5 eV
```

Scilab code Exa 2.4 Longest Wavelength of Incident Radiation

```
// Scilab Code Ex2.4 Longest wavelength of incident
   radiation: Pg:45 (2008)
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Energy equivalent of 1 eV, joule
   /eV
phi = 6*e;          // Work function of metal, joule
f0 = phi/h;     // Threshold frequency for metal
   surface, Hz
L0 = c/f0;      // Threshold (Longest) wavelength for
    metal, m
printf("\nThe longest wavelength of incident
   radiation = %4d angstrom", L0/1e-010);
// Result
// The longest wavelength of incident radiation =
   2070 angstrom
```

Scilab code Exa 2.5 Threshold Frequency and Wavelength

```
// Scilab Code Ex2.5 Threshold frequency and
   wavelength: Pg:46 (2008)
h = 6.62e-034;     // Planck's constant, Js
phi = 3.31e-019;        // Work function of metal,
   joule
c = 3e+08;     // Speed of light, m/s
f0 = phi/h;     // Threshold frequency for metal
   surface, Hz
L0 = c/f0;      // Threshold wavelength for metal, m
printf("\nThe threshold frequency for metal = %1.0e
   Hz", f0);
```

```
printf("\nThe threshold wavelength for metal = %4d angstrom", round(L0/1e-10));
// Result
// The threshold frequency for metal = 5e+014 Hz
// The threshold wavelength for metal = 6000 angstrom
```

Scilab code Exa 2.6 `Maximum Velocity of Emitted Electrons`

```
// Scilab Code Ex2.6 Maximum velocity of emitted electrons: Pg:46 (2008)
h = 6.624e-034;    // Planck's constant, Js
c = 3e+08;    // Speed of light, m/s
m = 9.1e-031;    // Mass of an electron, kg
e = 1.6e-019;    // Energy equivalent of 1 eV, joule/eV
L = 4300e-010;    // Wavelength of incident light, m
phi = 5*e;    // Work function of nickel surface, joule
f0 = phi/h;    // Threshold frequency for nickel, Hz
L0 = c/f0;    // Threshold wavelength for nickel, m
printf("\nThe threshold wavelength for nickel = %4d angstrom", L0/1e-10);
printf("\nSince %4d A < %4d A, the electrons will not be emitted.", L0/1e-010, L/1e-010);
phi = 2.83*e;    // Work function of potassium surface, joule
f0 = phi/h;    // Threshold frequency for potassium, Hz
L0 = c/f0;    // Threshold wavelength for potassium, m
printf("\nThe threshold wavelength for potassium = %4d angstrom", L0/1e-10);
printf("\nSince %4d A > %4d A, the electrons will be emitted.", L0/1e-010, L/1e-010);
```

```
// Now KE = 1/2*m*v0^2 = h*f - h*f0, where v0 is the maximum velocity
// solving for v0, we have
v0 = sqrt(2*h*c/m*(1/L - 1/L0));    // Maximum velocity of photoelectrons, m/s
printf("\nThe maximum velocity of photoelectrons = %5.3e m/s", v0);
// Result
// The threshold wavelength for nickel = 2484 angstrom
// Since 2484 A < 4300 A, the electrons will not be emitted.
// The threshold wavelength for potassium = 4388 angstrom
// Since 4388 A > 4300 A, the electrons will be emitted.
// The maximum velocity of photoelectrons = 1.433e+005 m/s
```

Scilab code Exa 2.7 Maximum Energy of Ejected Electrons

```
// Scilab Code Ex2.7 Maximum energy of ejected electrons: Pg:47 (2008)
h = 6.6e-034;    // Planck's constant, Js
c = 3e+08;    // Speed of light, m/s
L = 2537e-010;    // Wavelength of incident light, m
L0 = 3250e-010;    // Threshold wavelength of silver, m
// As U = h*(f - f0), the kinetic energy of ejected electrons
U = h*c*(1/L - 1/L0);    // Maximum energy of ejected electrons, J
printf("\nThe maximum energy of ejected electrons = %5.3e J", U);
// Result
```

```
// The maximum energy of ejected electrons = 1.712e
    -019 J
```

Scilab code Exa 2.8 Maximum Kinetic Energy and Stopping Potential of Ejected Elect

```
// Scilab Code Ex2.8 Maximum kinetic energy and
    stopping potential of ejected electrons: Pg:47
    (2008)
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Energy equivalent of 1 eV, joule
    /eV
phi_0 = 1.51*e;    // Work function of the metal
    surface, J
L = 4000e-010;     // Wavelength of incident light, m
f = c/L;     // Frequency of incident light, Hz
U = h*f - phi_0;     // Maximum kinetic energy of
    ejected electrons, J
V = U/e;     // Stopping potential for ejected
    electrons, volt
printf("\nThe maximum energy of ejected electrons =
    %5.3f eV", U/e);
printf("\nThe stopping potential of ejected
    electrons = %5.3f V", V);
// Result
// The maximum energy of ejected electrons = 1.595
    eV
// The stopping potential of ejected electrons =
    1.595 V
```

Scilab code Exa 2.9 Work Function of Metal

```
// Scilab Code Ex2.9 Work function of metal: Pg:48 (2008)
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Energy equivalent of 1 eV, joule/eV
V = 1;     // Stopping potential for the electrons emitted from the metal, V
L = 2500e-010;     // Wavelength of incident light, m
f = c/L;     // Frequency of incident light, Hz
// Now KE = h*f - phi = e*V, Einstein's Photoelectric equation, solving for phi
phi = h*f - e*V;     // Work function of metal
printf("\nThe work function of metal = %5.3f eV", phi/e);
// Result
// The work function of metal = 3.968 eV
```

Scilab code Exa 2.10 Energy of Electrons Emitted From the Surface of Tungsten

```
// Scilab Code Ex2.10 Energy of electrons emitted from the surface of tungsten: Pg:48 (2008)
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Energy equivalent of 1 eV, joule/eV
L = 1800e-010;     // Wavelength of incident light, m
L0 = 2300e-010;     // Threshold wavelength of tungsten, m
E = h*c*(1/L - 1/L0);     // Einstein's photoelectric equation for kinetic energy of emitted electrons, J
printf("\nThe energy of electrons emitted from the surface of tungsten = %3.1f eV", E/e);
// Result
```

```
// The energy of electrons emitted from the surface of tungsten = 1.5 eV
```

Scilab code Exa 2.11 `Energy of Photon`

```
// Scilab Code Ex2.11 Energy of photon : Pg:49 (2008)
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
e = 1.6e-019;     // Energy equivalent of 1 eV, joule/eV
L = 1800e-010;     // Wavelength of incident light, m
L0 = 2300e-010;     // Threshold wavelength of tungsten, m
E = h*c*(1/L - 1/L0);     // Einstein's photoelectric equation for kinetic energy of emitted electrons, J
printf("\nThe energy of electrons emitted from the surface of tungsten = %3.1f eV", E/e);
// Result
// The energy of electrons emitted from the surface of tungsten = 1.5 eV
```

Scilab code Exa 2.12 `Velocity of the Emitted Electron`

```
// Scilab Code Ex2.12 Velocity of the emitted electron: Pg:49 (2008)
m = 9.1e-031;     // Mass of electron, kg
c = 3e+08;     // Speed of light, m/s
h= 6.626 * 10^-34;
e = 1.6e-019;     // Energy equivalent of 1 eV, joule/eV
phi = 2.3*e;     // Work function of metal, J
```

```
L = 4300e-010;      // Wavelength of incident light, m
// As 1/2*m*v^2 = h*f - phi = h*c/L - phi, Einstein's photoelectric equation
// Solving for v
v = sqrt(2*(h*c/L - phi)/m);      // Velocity of emitted electron, m/s
printf("\nThe velocity of emitted electron = %4.2e eV", v);
// Result
// The velocity of emitted electron = 4.55e+005 eV
```

Scilab code Exa 2.13 Energy of a Quantum of Light

```
// Scilab Code Ex2.13 Energy of a quantum of light: Pg:50 (2008)
c = 3e+08;      // Speed of light, m/s
e = 1.6e-019;      // Energy equivalent of 1 eV, joule/eV
h= 6.626 * 10^-34;
L = 5.3e-07;      // Wavelength of incident light, m
E = h*c/L;      // Energy of the incident light, J
printf("\nThe energy of incident light = %4.2f eV", E/e);
// Result
// The energy of incident light = 2.34 eV
```

Scilab code Exa 2.14 Ratio of Masses of a Proton and an Electron

```
// Scilab Code Ex2.14 Ratio of masses of a proton and an electron: Pg:54 (2008)
RH = 1.09678e+07;      // Rydberg constant for hydrogen, per metre
```

```
RHe = 1.09722e+07;     // Rydberg constant for helium, per metre
MH_m_ratio = (RH - 1/4*RHe)/(RHe - RH);     // Ratio of mass of a proton to that of an electron
printf("\nThe ratio of mass of a proton to that of an electron = %4d", MH_m_ratio);
// Result
// The ratio of mass of a proton to that of an electron = 1869
```

Scilab code Exa 2.15 First Bohr Orbit in Hydrogen Atom

```
// Scilab Code Ex2.15 First Bohr Orbit in hydrogen atom: Pg:56 (2008)s
n = 1;     // Principle quantum number of first orbit in H-atom
h = 6.624e-034;     // Planck's Constant, Js
c = 3e+08;     // Speed of light, m/s
epsilon_0 = 8.85e-012;     // Absolute electrical permittivity of free space, coulomb square per newton per metre square
Z = 1;     // Atomic number of hydrogen
m = 9.1e-031;     // Mass of an electron, kg
e = 1.6e-019;     // Charge on an electron, coulomb
r = epsilon_0*n^2*h^2/(%pi*m*Z*e^2);     // Radius of first Bohr's orbit, m
v = Z*e^2/(2*8.85e-012*h*n);     // Velocity of electron in the first Bohr orbit, m/s
printf("\nThe radius of first Bohr orbit = %5.3f angstrom", r/1e-010);
printf("\nThe velocity of electron in first Bohr orbit = (1/%3d)c", 1/v*c);
// Result
// The radius of first Bohr orbit = 0.531 angstrom
// The velocity of electron in first Bohr orbit =
```

```
(1/137)c
```

Scilab code Exa 2.16 Wavelength of Balmer H beta Line

```
// Scilab Code Ex2.16 Wavelength of Balmer H_beta
   line: Pg:57 (2008)s
L_Hb = 6563e-010;    // Wavelength of H_beta line, m
R = 1.097e+07;    // Rydberg constant, per metre
L1 = 36/(5*R);    // Wavenumber of H_alpha line, per
    metre
L2 = 16/(3*R);    // Wavenumber of H_beta line, per
   metre
L_ratio = L2/L1;    // Ratio of wavelengths of
   H_beta and H_alpha lines
L2 = L_ratio*L1;    // Wavelength of Balmer H_beta
   line, m
printf("\nThe wavelength of Balmer H_beta line = %4d
    angstrom", L2/1e-010);
// Result
// The wavelength of Balmer H_beta line = 4861
   angstrom
```

Scilab code Exa 2.17 First Excitation Energy of Hydrogen Atom

```
// Scilab Code Ex2.17 First excitation energy of
   hydrogen atom: Pg: 58 (2008)s
n1 = 1;    // Principle quantum number of first
   orbit in H-atom
n2 = 2;    // Principle quantum number of second
   orbit in H-atom
m = 9.1e-031;    // Mass of the electron, C
e = 1.6e-019;    // Charge on an electron, coulomb
h = 6.624e-034;    // Planck's Constant, Js
```

```
epsilon_0 = 8.85e-012;    // Absolute electrical permittivity of free space, coulomb square per newton per metre square
U = m*e^4/(8*epsilon_0^2*h^2)*(1/n1^2 - 1/n2^2);    // First excitation energy of hydrogen atom, J
printf("\nThe first excitation energy of hydrogen atom = %5.2f eV", U/e);
// Result
// The first excitation energy of hydrogen atom = 10.17 eV
```

Scilab code Exa 2.18 Energy Difference in the Emission or Absorption of Sodium D1

```
// Scilab Code Ex2.18 Energy difference in the emission or absorption of sodium D1 line: Pg:58 (2008)s
h = 6.624e-034;    // Planck's Constant, Js
c = 3e+08;    // Speed of light, m/s
L = 590e-09;    // Wavelenght of sodium D1 line, m
E = h*c/L;    // Energy difference in the emission or absorption of sodium D1 line, J
printf("\nThe energy difference in the emission or absorption of sodium D1 line = %4.2e J", E);
// Result
// The energy difference in the emission or absorption of sodium D1 line = 3.37e-019 J
```

Scilab code Exa 2.19 Wavelength of First Line of Balmer Series

```
// Scilab Code Ex2.19 Wavelength of first line of Balmer series: Pg:58 (2008)s
n1 = 2;    // Ground level of Balmer line in H-atom
n2 = 4;    // Third level of Balmer line in H-atom
```

```
R = 1.097e+07;     // Rydberg constant, per metre
L2 = 1/((1/n1^2 - 1/n2^2)*R);     // Wavelength of second line of Balmer series, m
n2 = 3;     // Second level of Balmer line in H-atom
L1 = 1/((1/n1^2 - 1/n2^2)*R);     // Wavelength of first line of Balmer series, m
L_ratio = L1/L2;     // Wavelength ratio of first and second line of Balmer series, m
L2 = 4861;     // Given wavelength of second line of Balmer series, angstrom
L1 = L2*L_ratio;     // Wavelength of first line of Balmer series, angstrom
printf("\nThe wavelength of first line of Balmer series = %4d angstrom", L1);
// Result
// The wavelength of first line of Balmer series = 6562 angstrom
```

Scilab code Exa 2.20 Minimum Energy of the Electrons in Balmer Series

```
// Scilab Code Ex2.20 Minimum energy of the electrons in Balmer series: Pg:59 (2008)
n1 = 2;     // Ground level of Balmer line in H-atom
n2 = 3;     // Second level of Balmer line in H-atom
m = 9.1e-031;     // Mass of the electron, C
e = 1.6e-019;     // Charge on an electron, coulomb
h = 6.624e-034;     // Planck's Constant, Js
epsilon_0 = 8.85e-012;     // Absolute electrical permittivity of free space, coulomb square per newton per metre square
E = m*e^4/(8*epsilon_0^2*h^2)*(1/n1^2 - 1/n2^2);
    // Minimum energy required by an electron to correspond to first wavenumber of Balmer series, J
printf("\nMinimum energy required by an electron to
```

```
correspond to first wavenumber of Balmer series =
%4.2f", E/e);
// Result
// Minimum energy required by an electron to
correspond to first wavenumber of Balmer series =
1.88
```

Scilab code Exa 2.21 Ionization Potential of Hydrogen Atom

```
// Scilab Code Ex2.21 Ionization potential of
hydrogen atom: Pg:59 (2008)
m = 9.1e-031;     // Mass of the electron, C
e = 1.6e-019;     // Charge on an electron, coulomb
h = 6.626e-034;     // Planck's Constant, Js
epsilon_0 = 8.85e-012;     // Absolute electrical
permittivity of free space, coulomb square per
newton per metre square
phi = m*e^4/(8*epsilon_0^2*h^2);     // Work function
or ionization energy of hydrogen atom, J
printf("\nThe ionization energy of hydrogen atom =
%5.2f eV", phi/e);
// Result
// The ionization energy of hydrogen atom = 13.55 eV
```

Scilab code Exa 2.22 Wavelength of Second Number of Balmer Series of Hydrogen

```
// Scilab Code Ex2.22 Wavelength of second number of
Balmer series of hydrogen: Pg:60 (2008)
n1 = 2;     // Principle quantum number of second
orbit in H-atom
n2 = 3;     // Principle quantum number of third
orbit in H-atom
R = 1.097e+07;     // Rydberg constant, per metre
```

```
L1 = 1/((1/n1^2 - 1/n2^2)*R);    // Wavelength of
    first Balmer line, m
n2 = 4;    // Principle quantum number of third
    orbit in H-atom
L2 = 1/((1/n1^2 - 1/n2^2)*R);    // Wavelength of
    second Balmer line, m
L_ratio = L2/L1;    // Wavelength ratio of second
    and first line of Balmer series
L1 = 6563e-010;    // Given wavelength of first line
    of Balmer series, m
L2 = L_ratio*L1;    // Wavelength of second Balmer
    line, m
printf("\nThe wavelength of second Balmer line = %4e
    m", L2);
// Result
// The wavelength of second Balmer line = 4.861481e
    -007 m
```

Scilab code Exa 2.23 Wavelength of Emitted Light

```
// Scilab Code Ex2.23 Wavelength of emitted light:
    Pg:60 (2008)
e = 1.6e-019;    // Charge on an electron, coulomb
h = 6.624e-034;    // Planck's Constant, Js
n = 2;    // Principal quantum number for second
    orbit in H-atom
V = 13.6;    // Ionization potential of H-atom, V
U1 = -1*V*e;    // Energy of electron in first orbit
    , J
U2 = U1/n^2;    // Energy of electron in second
    orbit, J
// As U2 - U1 = h*c/L, solving for L
L = h*c/(U2 - U1);    // Wavelength of light emitted
    in the transition from second orbit to the first
    orbit, m
```

```
printf("\nThe wavelength of light emitted in the transition from second orbit to the first orbit = %4d angstrom", L/1e-010);
// Result
// The wavelength of light emitted in the transition from second orbit to the first orbit = 1217 angstrom
```

Scilab code Exa 2.24 Radius and Speed of Electron in the First Bohr Orbit

```
// Scilab Code Ex2.24 Radius and speed of electron in the first Bohr orbit: Pg:61 (2008)s
m = 9.1e-031;    // Mass of the electron, C
e = 1.6e-019;    // Charge on an electron, coulomb
h = 6.626e-034;    // Planck's Constant, Js
epsilon_0 = 8.85e-012;    // Absolute electrical permittivity of free space, coulomb square per newton per metre square
Z = 1, n = 1;
r_H = epsilon_0*n^2*h^2/(%pi*m*Z*e^2);    // Radius of first Bohr orbit, m
v_H = Z*e^2/(2*epsilon_0*n*h);    // Velocity of the electron in the first Bohr orbit, m/s
printf("\nThe radius of first Bohr orbit = %4.2e m", r_H);
printf("\nThe velocity of the electron in the first Bohr orbit = %3.1e m/s", v_H);
// Result
// The radius of first Bohr orbit = 5.31e-011 m
// The velocity of the electron in the first Bohr orbit = 2.2e+006 m/s
```

Scilab code Exa 2.25 Radius and Velocity of Electron for H and He

```
// Scilab Code Ex2.25 Radius and velocity of electron for H and He: Pg:61 (2008)s
m = 9.1e-031;     // Mass of the electron, kg
e = 1.6e-019;     // Charge on an electron, coulomb
h = 6.624e-034;     // Planck's Constant, Js
epsilon_0 = 8.85e-012;     // Absolute electrical permittivity of free space, coulomb square per newton per metre square
c = 3e+08;     // Speed of light, m/s
Z = 1, n = 1;     // Atomic number and principal quantum number of H-atom
r_H = epsilon_0*n^2*h^2/(%pi*m*Z*e^2);     // Radius of first Bohr orbit for H-atom, m
v_H = Z*e^2/(2*epsilon_0*n*h);     // Velocity of the electron in the first Bohr orbit of H-atom, m/s
printf("\nThe radius of first Bohr orbit = %4.2e m", r_H);
printf("\nThe velocity of the electron in the first Bohr orbit = %3.1e m/s", v_H);
printf("\nThe velocity of the electron in H-atom compared to the velocity of light = %4.2e", v_H/c);
Z = 2;     // Atomic number of He-atom
r_He = r_H/Z;     // Radius of first Bohr orbit for He-atom, m
v_He = 2*v_H;     // Velocity of the electron in the first Bohr orbit of He-atom, m/s
printf("\nThe radius of first Bohr orbit = %4.2e m", r_He);
printf("\nThe velocity of the electron in the first Bohr orbit = %3.1e m/s", v_He);
printf("\nThe velocity of the electron in He-atom compared to the velocity of light = %5.3e", v_He/c);
// Result
// The radius of first Bohr orbit = 5.31e-011 m
// The velocity of the electron in the first Bohr orbit = 2.2e+006 m/s
```

```
// The velocity of the electron in H-atom compared to the velocity of light = 7.28e-003
// The radius of first Bohr orbit = 2.65e-011 m
// The velocity of the electron in the first Bohr orbit = 4.4e+006 m/s
// The velocity of the electron in He-atom compared to the velocity of light = 1.456e-002
```

Scilab code Exa 2.26 `Difference in Wavelength in the Spectra of Hydrogen and Deute`

```
// Scilab Code Ex2.26 Difference in wavelength in the spectra of hydrogen and deuterium: Pg:62 (2008)
R_H = 1.097e+07;     // Rydberg constant for H-atom, per metre
M_H = 1;     // Mass of H-atom, amu
M_D = 2*M_H;     // Mass of D-atom, amu
m = 0.000549*M_H;     // Mass of an electron, amu
R_D = R_H*(1+m/M_H)/(1+m/M_D);     // Rydberg constant for D-atom, per metre
n1 = 2, n2 = 3;     // Principal qunatum numbers for first line of Balmer series
L_H = 1/(R_H*(1/n1^2 - 1/n2^2));     // Wavelength of H-atom, m
L_D = 1/(R_D*(1/n1^2 - 1/n2^2));     // Wavelength of D-atom, m
delta_H = (L_H - L_D)/1e-010;     // Difference in wavelength in the spectra of hydrogen and deuterium, angstrom
printf("\nThe difference in wavelength in the spectra of hydrogen and deuterium = %3.1f angstrom", delta_H);
// Result
// The difference in wavelength in the spectra of hydrogen and deuterium = 1.8 angstrom
```

Scilab code Exa 2.27 Ionization Energy of Hydrogen Atom With Orbiting Muon

```
// Scilab Code Ex2.27 Ionization energy of hydrogen
   atom with orbiting muon: Pg:63 (2008)
m = 9.1e-031;     // Mass of the electron , kg
e = 1.6e-019;     // Charge on an electron , coulomb
h = 6.624e-034;     // Planck's Constant , Js
epsilon_0 = 8.85e-012;     // Absolute electrical
   permittivity of free space , coulomb square per
   newton per metre square
m1 = 200*m;     // Mass of muon, kg
phi1 = m1*e^4/(8*epsilon_0^2*h^2);     // Ionization
   energy of H-atom with muon, J
printf("\nThe ionization energy of hydrogen atom
   with orbiting muon = %4.2e eV", phi1/1.6e-019);
// Result
// The ionization energy of hydrogen atom with
   orbiting muon = 2.71e+003 eV
```

Scilab code Exa 2.28 Photon Emitted by Hydrogen Atom

```
// Scilab Code Ex2.28 Photon emitted by hydrogen
   atom: Pg:64 (2008)
e = 1.6e-019;     // Energy equivalent of 1 eV, joule
   /eV
h = 6.624e-034;     // Planck's constant , Js
c = 3e+08;     // Speed of light , m/s
E1 = -13.6;     // Energy of electron in the first
   orbit of hydrogen atom, eV
n = 2;     // Principal quantum number for second
   orbit
```

```
E2 = E1/n^2;    // Energy of electron in the second orbit of hydrogen atom, eV
E = (E2 - E1)*e;    // Energy of photon emitted, joule
P = E/c;    // Momentum of photon, kg-m/s
L = (h/P)/1e-010;    // de_Broglie wavelength of photon, angstrom
printf("\nThe energy of photon emitted by hydrogen atom %5.2e J", E);
printf("\nThe momentum of photon = %4.2e kg-m/s", P);
printf("\nThe de_Broglie wavelength of photon = %4d angstrom", L);
// Result
// The energy of photon emitted by hydrogen atom 1.63e-018 J
// The momentum of photon = 5.44e-027 kg-m/s
// The de_Broglie wavelength of photon = 1217 angstrom
```

Scilab code Exa 2.29 Energy Required to Create a Vacancy in Cu

```
// Scilab Code Ex2.29 Energy required to create a vacancy in Cu: Pg:64 (2008)
n = 1;    // Principal quantum number of K shell
Z = 29;    // Atomic number of copper
U = 13.6;    // Ionization potential of hydrogen atom, eV
E1 = Z^2*U/n^2;    // Energy required to create a vacancy in K-shell of copper atom, eV
n = 2;    // Principal quantum number of L shell
E2 = Z^2*U/n^2;    // Energy required to create a vacancy in K-shell of copper atom, eV
printf("\nThe energy required to create a vacancy in K-shell of copper atom = %5.2e eV", E1);
```

```
printf("\nThe energy required to create a vacancy in
    L-shell of copper atom = %5.2e eV", E2);
// Result
// The energy required to create a vacancy in K-
    shell of copper atom = 1.14e+004 eV
// The energy required to create a vacancy in L-
    shell of copper atom = 2.86e+003 eV
```

Scilab code Exa 2.30 Excitation Potential for Mercury

```
// Scilab Code Ex2.30 Excitation potential for
    mercury: Pg:65 (2008)
e = 1.6e-019;     // Energy equivalent of 1 eV, joule
    /eV
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Speed of light, m/s
L = 2537e-010;     // Wavelength of absorbed line of
    Hg, m
V = h*c/(e*L);     // Excitation potential for Hg, v
printf("\nThe excitation potential for Hg = %3.1f V"
    , V);
// Result
// The excitation potential for Hg = 4.9 V
```

Scilab code Exa 2.31 Atomic Number of Impurity in Zinc Target

```
// Scilab Code Ex2.31 Atomic number of impurity in
    Zinc target: Pg:65 (2008)
L1 = 1.43603e-010;     // Wavelength of
    characteristic K_alpha line from Zn, m
Z1 = 30;     // Atomic number of zinc
L2 = 0.53832e-010;     // Wavelength of unknown line
    from Zn, m
```

```
// As (1/L1)/(1/L2) = (Z1/Z2)^2, solving for Z2
Z2 = Z1*(L1/L2)^(1/2);    // Atomic number of impurity in Zn target
printf("\nThe atomic number of impurity in Zn target = %2d", round(Z2));
// Result
// The atomic number of impurity in Zn target = 49
```

Scilab code Exa 2.32 Mu mesonic Atom Subjected to Bohr Orbit

```
// Scilab Code Ex2.32 Mu-mesonic atom subjected to Bohr orbit: Pg:65 (2008)
Z = 3;    // Atomic number of Mu-mesonic atom
m_e = 9.1e-031;    // Mass of the electron, kg
e = 1.6e-019;    // Charge on an electron, coulomb
h = 6.624e-034;    // Planck's Constant, Js
epsilon_0 = 8.85e-012;    // Absolute electrical permittivity of free space, coulomb square per newton per metre square
m = 200*m_e;    // Mass of a muon, kg
// As r_H = epsilon_0*^h^2/(%pi*m*(e^2) and r = epsilon_0*n^2*h^2/(%pi*m*Z*(e^2)
// r = r_H gives
n = sqrt(m/m_e*Z);    // Value of n for which r = r_H
n1 = 1, n2 = 2;    // Principal quantum numbers corresponding to first excitation
U = m*e^4*Z^2/(8*epsilon_0^2*h^2*1.6e-019)*(1/n1^2-1/n2^2);    // First excitation potential of the atom, eV
printf("\nThe value of n for which radius of orbit is equal to Bohr radius = %2d", round(n));
printf("\nThe first excitation potential of the atom = %4.2e eV", U);
// Result
```

```
// The value of n for which radius of orbit is equal
    to Bohr radius = 24
// The first excitation potential of the atom = 1.83
   e+004 eV
```

Chapter 3

Matter Waves Wave Particle Duality and Uncertainty Principle

Scilab code Exa 3.1 Kinetic Energy of an Electron

```
// Scilab code: Ex3.1 : Kinetic energy of an
   electron: Pg: 77 (2008)
h = 6.6e-034;    // Planck's constant, J-s
m = 9.1e-031;    // mass of an electron, kg
L = 9e-010;    // wavelength of an electron, m
// since E = (m*v^2)/2, Energy of an electron, joule
// thus v = sqrt(2*E/m), solving for L in terms of E
   , we have
// L = h/sqrt(2*m*E), wavelength of an electron, m
// On solving for E
E = h^2/(2*m*L^2)
printf("\nThe kinetic energy of an electron = %6.4f
   eV", E/1.6e-019);
// Result
// The kinetic energy of an electron = 1.8468 eV
```

Scilab code Exa 3.2 Wavelength of Electrons

```
// Scilab code: Ex3.2 : Wavelength of electrons: Pg:
    78 (2008)
h = 6.6e-034;     // Planck's constant, J-s
m = 9.1e-031;     // mass of an electron, kg
e = 1.6e-019;     // Charge on an electron, coulomb
E = 100*e;    // Energy of beam of electrons, joule
// since E = (m*v^2)/2;    // Energy of beam of
    electron, joule
p = sqrt(2*m*E);     // Momentum of beam of electrons
    , kg-m/s
L = h/p;    // wavelength of a beam of electron, m
printf("\nThe wavelength of electrons = %4.2f
    angstorm", L/1e-010);
// Result
// The wavelength of electrons = 1.22 angstorm
```

Scilab code Exa 3.3 Momentum of Photon

```
// Scilab code: Ex3.3 : Momentum of photon: Pg: 78
    (2008)
h = 6.624e-034;     // Planck's constant, J-s
L = 6e-07;    // wavelength of photon, m
M = h/L;    // Momentum of photon, kg-m/s
printf("\nThe momentum of photon = %5.3e kg-m/s", M)
    ;
// Result
// The momentum of photon = 1.104e-027 kg-m/s
```

Scilab code Exa 3.4 Momentum of an electron

```
// Scilab code: Ex3.4 : Momentum of an electron: Pg:
    78 (2008)
m = 9.1e-031;    // Mass of an electron, kg
E = 1.6e-010;    // Kinetic energy of an electron,
   joule
// Since E = p^2/2*m;    // Kinetic energy of an
   electron, joule
p = sqrt(2*m*E);    // Momentum of an electron, kg-m
   /s
printf("\nThe momentum of an electron = %3.1e kg-m/s
   ", p);
// Result
// The momentum of an electron = 1.7e-020 kg-m/s
```

Scilab code Exa 3.5 Wavelength of a Particle

```
// Scilab code: Ex3.5 : wavelength of a particle: Pg
   : 79 (2008)
h = 6.624e-034;    // Planck's constant, J-s
m = 9e-031;    // Mass of an electron, kg
U = 1.6e-017;    // Kinetic energy of an particle,
   joule
// Since U = (m*v^2)/2;    // Kinetic energy of a
   particle, joule
// such that v = sqrt(2*U/m);    // Velocity of the
   particle, m/s
L = h/sqrt(2*m*U);    // wavelength of a particle, m
printf("\nThe wavelength of a particle = %5.3f
   angstorm", L/1e-010);
// Result
// The wavelength of a particle = 1.234 angstorm
```

Scilab code Exa 3.6 `Comparison of Energy of Photon and Neutron`

```
// Scilab code: Ex3.6 : Comparison of energy of photon and neutron: Pg: 79 (2008)
m = 1.67e-027;      // Mass of neutron, kg
L = 1e-010;     // Wavelength of neutron and photon, m
c = 3e+08;     // Velocity of light, m/s
h = 6.624e-034;     // Plancks constant, joule second
U_1 = h*c/L;     // Energy of photon, joule
// Since  U_2 = (m*v^2)/2, Energy of neutron, joule
// Thus v = h/m*L_2, Velocity of the particle, m/s
// on solving for U_2
U_2 = h^2/(2*m*L^2);     // Energy of photon, joule
printf("\nThe ratio of energy of photon and neutron = %4.2e ", U_1/U_2);
// Result
// The ratio of energy of photon and neutron = 1.51e+005
```

Scilab code Exa 3.7 `de Broglie Wavelength of Electrons`

```
// Scilab code: Ex3.7: de-Broglie wavelength of electrons: Pg: 80 (2008)
L_1 = 3e-07;     // Wavelength of ultraviolet light, m
L_0 = 4e-07;     // Threshold wavelength of ultraviolet light, m
m = 9.1e-031;     // Mass of an electron, kg
c = 3e+08;     // Velocity of light, m/s
h = 6.624e-034;     // Plancks constant, joule-second
```

```
U = h*c*(1/L_1-1/L_0);     // Maximum Kinetic energy of emitted electrons, joule
// since U = m*v^2/2, Kinetic energy of electrons, joule
// Thus v = sqrt(2*U/m), so that L_2 becomes
L_2 = h/sqrt(2*m*U);     // wavelength of electrons, m
printf("\nThe wavelength of the electrons = %3.1f angstorm", L_2/1e-010);
// Result
// The wavelength of the electrons = 12.1 angstorm
```

Scilab code Exa 3.8 de Broglie Wavelength of Accelerated Electrons

```
// Scilab code: Ex3.8 : de-Broglie wavelength of accelerated electrons:Pg: 80 (2008)
m = 9.1e-031;     // Mass of an electron, kg
e = 1.6e-019;     // Charge on an electron, Coulamb
h = 6.624e-034;     // Plancks constant, joule second
V = 1;     // For simplicity, we assume retarding potential to be unity, volt
// Since  e*V = (m*v^2)/2;     // Energy of electron, joule
v = sqrt(2*e*V/m);     // Velocity of electrons, m/s
L = h/(m*v);     // Wavelength of electrons, m
printf("\nThe de-Broglie wavelength of accelerated electrons = %5.2f/sqrt(V) ", L/1e-010);
// Result
// The de-Broglie wavelength of accelerated electrons = 12.28/sqrt(V)
```

Scilab code Exa 3.9 Wavelength of Matter Waves

```
// Scilab code: Ex3.9 : Wavelength of matter waves: Pg: 81 (2008)
E = 2e-016;     // Energy of electrons, joule
h = 6.624e-034;     // Planck's constant, J-s
m = 9.1e-031;     // mass of the electron, kg
// since E = (m*v^2)/2, the energy of an electron, joule
// such that v = sqrt(2*E/m);     // Velocity of electron, m/s
// As L = h/m*v, wavelength of the electron, m
// on solving for L in terms of E
L = h/sqrt(2*m*E);     // wavelength of the electron, m
printf("\nThe wavelength of the electron = %5.3f angstorm", L/1e-010);
// Result
// The wavelength of the electron = 0.347 angstorm
```

Scilab code Exa 3.10 Momentum of Proton

```
// Scilab code: Ex3.10 : Momentum of proton: Pg: 81 (2008)
U = 1.6e-010;     // Kinetic energy of proton, joule
h = 6.624e-034;     // Planck's constant, J-s
m = 1.67e-027;     // mass of proton, kg
v = sqrt(2*U/m);     // Velocity of proton, m/s
p = m*v;     // Momentum of proton, kg m/s
printf("\nThe momentum of proton = %4.2e kgm/s", p);
// Result
// The momentum of proton = 7.31e-019 kgm/s
```

Scilab code Exa 3.11 Wavelength of an Electron

```
// Scilab code: Ex3.11 : Wavelength of an electron:
    Pg: 82 (2008)
U = 1.6e-013;    // Kinetic energy of the electron,
    joule
h = 6.624e-034;    // Planck's constant, J-s
m = 9.1e-031;    // Mass of the electron, kg
v = sqrt(2*U/m);    // Velocity of the electron, m/s
L = h/(m*v);    // Wavelength of the electron, m
printf("\nThe wavelength of an electron = %5.3e
    angstorm", L/1e-010);
// Result
// The wavelength of an electron = 1.228e-002
    angstorm
```

Scilab code Exa 3.12 de Broglie Wavelength of Thermal Neutrons

```
6// Scilab code: Ex3.12: De-Broglie wavelength of
    thermal neutrons:Pg: 82 (2008)
m = 1.6749e-027;    // Mass of neutron, kg
h = 6.624e-034;    // Plancks constant, joule second
k = 1.38e-021;    // Boltzmann constant, joule per
    kelvin
T = 300;    // Temperature of thermal neutrons,
    kelvin
// Since m*v^2/2 = (3/2)*k*T;    // Energy of
    neutron, joule
v = sqrt(3*k*T/m);    // Velocity of neutrons, m/s
L = h/(m*v);    // Wavelength of neutrons, m
printf("\nThe de-Broglie wavelength of thermal
    neutrons = %5.3f angstorm ", L/1e-010);
// Result
// The de-Broglie wavelength of thermal neutrons =
    0.145 angstorm
```

Scilab code Exa 3.13 Kinetic Energy of a Proton

```
// Scilab code: Ex3.13 : Kinetic energy of a proton:
    Pg: 82 (2008)
L = 1e-010;     // wavelength of proton, m
m = 1.67e-027;     // Mass of proton, kg
h = 6.624e-034;     // Plancks constant, joule second
// Since L = h/(m*v);     // wavelength of proton, m
v = h/m*L;     // Velocity of proton, m/s
v_k = h^2/(2*L^2*m);     // Kinetic energy of proton,
    joule
printf("\nThe kinetic energy of proton = %3.1e eV ",
    v_k/1.6e-019);
// Result
// The kinetic energy of proton = 8.2e-002 eV
```

Scilab code Exa 3.14 Energy of Electrons in a One Dimensional Box

```
// Scilab Code Ex3.14: Energy of electrons in a one
    dimensional box: Pg: 85 (2008)
n1 = 1, l = 0, ml = 0, ms = 1/2;     // Quantum
    numbers of first electron
n2 = 1, l = 0, ml = 0, ms = -1/2;     // Quantum
    numbers of second electron
// The lowest energy corresponds to the ground state
    of electrons
n = n1;     // n1 = n2 = n
m = 9.1e-031;     // Mass of electron, kg
h = 6.626e-034;     // Planck's constant, Js
a = 1;     // For convenience, length of the box is
    assumed to be unity
```

```
E = 2*n^2*h^2/(8*m*a^2);     // Lowest energy of electron, joule
printf("\nThe lowest energy of electron = %6.4e/a^2", E);
// Result
// The lowest energy of electron = 1.2062e-037/a^2
```

Scilab code Exa 3.15 Lowest Energy of Three Electrons in Box

```
// Scilab Code Ex3.15: Lowest energy of three electrons in box: Pg:85 (2008)
n1 = 1, l = 0, ml = 0, ms = 1/2;     // Quantum numbers of first electron
n2 = 1, l = 0, ml = 0, ms = -1/2;     // Quantum numbers of second electron
n3 = 2, l = 0, ml = 0, ms = +1/2;     // Quantum numbers of third electron
// The lowest energy corresponds to the ground state of electrons
m = 9.1e-031;     // Mass of electron, kg
h = 6.626e-034;     // Planck's constant, Js
a = 1;     // For convenience, length of the box is assumed to be unity
E = (n1^2*h^2/(8*m*a^2)+n2^2*h^2/(8*m*a^2))+n3^2*h^2/(8*m*a^2);     // Lowest energy of electron, joule
printf("\nThe lowest energy of electron = %6.4e/a^2", E);
// Result
// The lowest energy of electron = 3.6185e-037/a^2
```

Scilab code Exa 3.16 Zero Point Energy of System

```
// Scilab code: Ex3.16 : Zero point energy of system
    :Pg: 86 (2008)
m = 9.1e-031;     // Mass of an electron, kg
a = 1e-010;     // Length of box, m
h = 6.624e-034;     // Plancks constant, joule second
n = 1;     // Principal quantum number for the lowest
     energy level
E1 = 2*h^2/(8*m*a^2);     // Energy for the two
    electron system in the n =1 energy level, joule
E2 = 8*(2^2*h^2)/(8*m*a^2);     // Energy for the
    eight electron system in the n = 2 energy level,
    joule
E = E1 +E2;     // Total lowest energy of system,
    joule
printf("\nThe zero point energy of system = %4.2e J
    ", E);
// Result
// The zero point energy of system = 2.05e-016 J
```

Scilab code Exa 3.17 Mean Energy Per Electron at 0K

```
// Scilab code: Ex3.17 : Mean energy per electron at
     0K:Pg: 86 (2008)
m = 9.1e-031;     // Mass of an electron, kg
a = 50e-010;     // Length of molecule, m
h = 6.624e-034;     // Plancks constant, joule second
E = h^2/(8*m*a^2);     // Energy per electron, joule
printf("\nThe mean energy per electron at 0K = %3.1e
     eV ", E/1.6e-019);
// Result
// The mean energy per electron at 0K = 1.5e-002 eV
```

Scilab code Exa 3.18 Lowest Energy of Two Electron System

```
// Scilab code: Ex3.18 : Lowest energy of two
    electron system:Pg: 87 (2008)
m = 9.1e-031;     // Mass of an electron, kg
a = 1e-010;     // Length of box, m
h = 6.624e-034;     // Plancks constant, joule second
E = 2*h^2/(8*m*a^2);     // Energy of two electron
    system, joule
printf("\nThe lowest energy of two electron system =
    %4.1f, eV", E/1.6e-019);
// Result
// The lowest energy of two electron system = 75.3,
    eV
```

Scilab code Exa 3.19 `Total Energy of the Three Electron System`

```
// Scilab code: Ex3.19 : Total energy of the three
    electron system:Pg: 87 (2008)
m = 9.1e-031;     // Mass of an electron, kg
h = 6.624e-034;     // Plancks constant, joule second
a = 1e-010;     // Length of the molecule, m
E = 6*h^2/(8*m*a^2);     // Energy of three electron
    system, joule
printf("\nThe total energy of three electron system
    = %6.2f, eV ", E/1.6e-019);
// Result
// The total energy of three electron system =
    226.02, eV
```

Scilab code Exa 3.20 `Minimum Uncertainity in the Velocity of an Electron`

```
// Scilab code: Ex3.20 : Minimum uncertainity in the
    velocity of an electron:Pg: 92 (2008)
m = 9.1e-031;     // Mass of an electron, kg
```

```
del_x = 1e-010;     // Length of the box, m
h_bar = 1.054e-034;     // Reduced Plancks constant, joule second
del_v = h_bar/(m*del_x);     // Minimum uncertainity in velocity, m/s
printf("\nThe minimum uncertainity in the velocity of electron = %4.2e m/s ", del_v);
// Result
// The minimum uncertainity in the velocity of electron = 1.16e+006 m/s
```

Scilab code Exa 3.21 Uncertainity in Momentum and Kinetic Energy of the Proton

```
// Scilab code: Ex3.21 : Uncertainity in momentum and kinetic energy of the proton:Pg: 92 (2008)
m = 1.67e-027;     // Mass of a proton, kg
del_x = 1e-014;     // Uncertainity in position, m
h_bar = 1.054e-034;     // Reduced Plancks constant, joule second
del_p = h_bar/del_x;     // Minimum uncertainity in momentum, kgm/s
del_E = del_p^2/(2*m);     // Minimum uncertainity in kinetic energy, joule
printf("\nThe minimum uncertainity in momentum of the proton = %5.3e kgm/s", del_p);
printf("\nThe minimum uncertainity in kinetic energy of the proton = %5.3e eV", del_E/1.6e-019);
// Result
// The minimum uncertainity in momentum of the proton = 1.054e-020 kgm/s
// The minimum uncertainity in kinetic energy of the proton = 2.079e+005 eV
```

Scilab code Exa 3.22 Uncertainity in the Position of an Electron

```
// Scilab code: Ex3.22 : Uncertainity in the
   position of an electron:Pg: 93 (2008)
m = 9.1e-031;    // Mass of an electron, kg
v = 600;    // Speed of electron, m/s
h_bar = 6.6e-034;    // Reduced Plancks constant,
   joule second
p = m*v;    // Momentum of electron, kgm/s
del_p = 5e-05*m*v;    // Minimum uncertainity in
   momentum, kgm/s
del_x = h_bar/(4*%pi*del_p);    // Uncertainity in
   position, m
printf("\nThe uncertainity in the position of the
   electron = %5.3f mm", del_x/1e-03);
// Result
// The uncertainity in the position of the electron
   = 1.924 mm
```

Scilab code Exa 3.23 Uncertainity in the Position of a Bullet

```
// Scilab code: Ex3.23 : Uncertainity in the
   position of a bullet:Pg: 93 (2008)
m = 0.025;    // Mass of an bullet, kg
v = 400;    // Speed of bullet, m/s
h_bar = 6.6e-034;    // Reduced Plancks constant,
   joule second
p = m*v;    // Momentum of bullet, kgm/s
del_p = 2e-04*p;    // Minimum uncertainity in
   momentum, kgm/s
del_x = h_bar/(4*%pi*del_p);    // Uncertainity in
   position, m
printf("\nThe uncertainity in the position of the
   bullet = %5.3e m", del_x);
// Result
```

```
//  The uncertainity in the position of the bullet = 2.626e-032 m
```

Scilab code Exa 3.24 Uncertainity in the Position of an Electron

```
// Scilab code: Ex3.24 : Uncertainity in the position of an electron:Pg: 94 (2008)
m = 9.1e-31;    // Mass of an electron, kg
v = 300;    // Speed of electron, m/s
h_bar = 6.6e-034;    // Reduced Plancks constant, joule second
p = m*v;    // Momentum of electron, kgm/s
del_p = 1e-04*p;    // Minimum uncertainity in momentum, kgm/s
del_x = h_bar/(4*%pi*del_p);    // Uncertainity in position, m
printf("\nThe uncertainity in the position of the electron = %5.3f mm", del_x/1e-03);
// Result
//  The uncertainity in the position of the electron = 1.924 mm
```

Scilab code Exa 3.25 Uncertainity in the Velocity of an Electron

```
// Scilab code: Ex3.25 : Uncertainity in the velocity of an electron:Pg: 94 (2008)
m = 9.1e-31;    // Mass of an electron, kg
del_x = 1e-10;    // Length of box, m
h_bar = 6.6e-034;    // Reduced Plancks constant, joule second
del_v = h_bar/(2*%pi*del_x*m);    // Minimum uncertainity in velocity of an electron, m/s
```

```
del_p = m*del_v;     // Uncertainity in Momentum of electron , kgm/s
printf("\nThe uncertainity in the velocity of the electron = %3.2e m/s", del_v);
// Result
// The uncertainity in the velocity of the electron = 1.15e+006 m/s
```

Scilab code Exa 3.26 Minimum Uncertainity in the Energy of the Excited State of an

```
// Scilab code: Ex3.26 : Minimum uncertainity in the energy of the excited state of an atom:Pg: 94 (2008)
del_t = 1e-08;     // Life time of an excited state of an atom, seconds
h_bar = 1.054e-034;     // Reduced Plancks constant, joule second
del_E = h_bar/del_t;     // Minimum uncertainity in the energy of excited state, joule
printf("\nThe minimum uncertainity in the energy of the excited state = %5.3e joule", del_E);
// Result
// The minimum uncertainity in the energy of the excited state = 1.054e-026 joule
```

Chapter 4

Mechanics

Scilab code Exa 4.1 Percentage Transmission of Beam Through Potential Barrier

```
// Scilab code: Ex4.1 : Percentage transmission of beam through potential barrier: Pg: 124 (2008)
eV = 1.6e-019;    // Energy required by an electron to move through a potential barrier of one volt, joules
m = 9.1e-031;    // Mass of electron, kg
E = 4*eV;    // Energy of each electron, joule
Vo = 6*eV    // Height of potential barrier, joule
a = 10e-010;    // Width of potential barrier, m
h_bar = 1.054e-34;    // Reduced Planck's constant, J-s
k = 2*m*(Vo-E)/h_bar^2
// Since 2*k*a = 2*a*[2*m*(Vo-E)^1/2]/h_bar so
pow = 2*a/h_bar*[2*m*(Vo-E)]^(1/2);    // Power of exponential in the expression for T
T = [16*E/Vo]*[1-E/Vo]*exp(-1*pow);    // Transmission coefficient of the beam through the potential barrier
percent_T = T*100;
printf("\nThe percentage transmission of beam throught potential barrier = %5.3e percent",
```

```
    percent_T);
// Result
// The percentage transmission of beam throught
    potential barrier = 1.828e-004 percent
```

Scilab code Exa 4.2 Width of the Potential Barrier

```
// Scilab code: Ex4.2 : Width of the potential
    barrier: Pg: 125 (2008)
A = 222;    // Atomic weight of radioactive atom
Z = 86;    // Atomic number of radioactive atom
eV = 1.6e-19;    // Energy required by an electron
    to move through a potential barrier of one volt,
    joules
epsilon_0 = 8.854e-012;    // Absolute electrical
    permittivity of free space, coulomb square per
    newton per metre square
e = 1.6e-19;    // Charge on an electron, coulomb
r0 = 1.5e-015;    // Nuclear radius constant, m
r = r0*A^(1/3);    // Radius of the radioactive atom
    , m
E = 4*eV*1e+06;    // Kinetic energy of an alpha
    particle, joule
// At the distance of closest approach, r1, E = 2*(Z
    -2)*e^2/(4*%pi*epsilon_0*r1)
// Solving for r1, we have
r1 = 2*(Z-2)*e^2/(4*%pi*epsilon_0*E);    // The
    distance form the centre of the nucleus at which
    PE = KE
a = r1 - r;    // Width of the potential barrier, m
printf("\nThe width of the potential barrier of the
    alpha particle = %5.2e m", a);
// Result
// The width of the potential barrier of the alpha
    particle = 5.13e-014 m
```

Scilab code Exa 4.3 Energy of Electrons Through the Potential Barrier

```
// Scilab code: Ex4.3: Energy of electrons through
   the potential barrier : Pg : 125 (2008)
h_bar = 1.054e-34;    // Reduced Planck's constant,
   J-s
Vo = 8e-019;    // Height of potential barrier,
   joules
m = 9.1e-031;    // Mass of an electron, kg
a = 5e-010;    // Width of potential barrier, m
T = 1/2;    // Transmission coefficient of electrons
// As T = 1/((1 + m*Vo^2*a^2)/2*E*h^2), solving for
   E we have
E = m*Vo^2*a^2/(2*(1/T-1)*h_bar^2*1.6e-019);    //
   Energy of half of the electrons through the
   potential barrier, eV
printf("\nThe energy of electrons through the
   potential barrier = %5.2f eV", E);
// Result
// The energy of electrons through the potential
   barrier = 40.96 eV
```

Scilab code Exa 4.4 Zero Point Energy of a System

```
// Scilab code: Ex4.4 : Zero point energy of a
   system : Pg: 126 (2008)
h = 6.626e-034;    // Planck's constant, Js
x = 1e-02;    // Displacement of the spring about
   its mean position, m
F = 1e-02;    // Force applied to the spring-mass
   system, N
```

```
m = 1e-03;     // Mass of attached to the spring, kg
// As F = k*x, k = 4*%pi^2*f^2*m is the stiffness
    constant, solving for f,
f = sqrt(F/(4*%pi^2*m*x));     // Frequency of
    oscillations of mass-spring system, Hz
U = 1/2*h*f;     // Zero point energy of the mass-
    spring system, J
printf("\nThe zero point energy of the mass-spring
    system = %4.2e J", U);
// Result
// The zero point energy of the mass-spring system =
    1.67e-033 J
```

Chapter 5

Atomic Physics

Scilab code Exa 5.1 L S coupling for two electrons

```
// Scilab Code Ex5.1 L-S coupling for two electrons:
    Pg:145 (2008)
// For 2D(3/2) state
l2 = 1;     // Orbital quantum number for p state
l1 = 1;     // Orbital quantum number for p state
printf("\nThe values of orbital quantum number L,
   for l1 = %d and l2 = %d are: \n", l1, l2);
for L = l2-l1:1:l2+l1
printf("%d ", L);
end
// Result
// The values of orbital quantum number L, for l1 =
   1 and l2 = 1 are:
// 0 1 2
```

Scilab code Exa 5.2 Term Values for L S Coupling

```
// Scilab Code Ex5.2 Term values for L-S coupling:
   Pg:145 (2008)
```

```
// For 2D(3/2) state
// Set-I values of L and S
L = 1;     // Orbital quantum number
S = 1/2;     // Spin quantum number
printf("\nThe term values for L = %d and S = %2.1f (
   P-state) are:\n",L, S);
J1 = 3/2;     // Total quantum number
printf("%dP(%2.1f)\t", 2*S+1,J1);
J2 = 1/2;     // Total quantum number
printf("%dP(%2.1f)", 2*S+1,J2);

// Set-II values of L and S
L = 2;     // Orbital quantum number
S = 1/2;     // Spin quantum number
printf("\nThe term values for L = %d and S = %2.1f (
   P-state) are:\n",L, S);
J1 = 5/2;     // Total quantum number
printf("%dD(%2.1f)\t", 2*S+1,J1);
J2 = 3/2;     // Total quantum number
printf("%dD(%2.1f)", 2*S+1,J2);

// Result
// The term values for L = 1 and S = 0.5 (P-state)
    are:
// 2P(1.5)  2P(0.5)
// The term values for L = 2 and S = 0.5 (P-state)
    are:
// 2D(2.5)  2D(1.5)
```

Scilab code Exa 5.4 Angle Between l and s State

```
// Scilab Code Ex5.4 Angle between l and s for 2D
    (3/2) state: Pg:146 (2008)
// For 2D(3/2) state
l = 2;     // Orbital quantum number
```

```
s = 1/2;     // Spin quantum number
j = l+s;     // Total quantum number
// Now by cosine rule of L-S coupling
// cos(theta) = (j*(j+1)-l*(l+1)-s*(s+1))/(2*sqrt(s
    *(s+1))*sqrt(l*(l+1))), solving for theta
theta = acosd((l*(l+1)+s*(s+1)-j*(j+1))/(2*sqrt(s*(s
    +1))*sqrt(l*(l+1))));     // Angle between l and s
     for 2D(3/2) state
printf("\nThe angle between l and s for 2D(3/2)
    state = %5.1f degrees", theta);
// Result
// The angle between l and s for 2D(3/2) state =
    118.1 degrees
```

Chapter 6

X Rays

Scilab code Exa 6.1 Wavelength of X rays

```
// Scilab code: Ex6.1 : Wavelength of X-rays: Pg:
   156 (2008)
h = 6.6e-034;    // Planck's constant, J-s
V = 50000;    // Potential difference, volts
c = 3e+08;    // Velocity of light, m/s
e = 1.6e-019;    // Charge of an electron, coulombs
L_1 = h*c/(e*V);    // wavelength of X-rays, m
L = L_1/1e-010;    // wavelength of X-rays, angstorm
printf("\nThe shortest wavelength of X-rays = %6.4f
   angstorm", L);
// Result
// The shortest wavelength of X-rays = 0.2475
   angstorm
```

Scilab code Exa 6.2 Plancks constant

```
// Scilab code: Ex6.2 : Planck's constant: Pg: 156
   (2008)
```

```
L = 24.7e-012;     // Wavelength of X-rays, m
V = 50000;     // Potential difference, volts
c = 3e+08;     // Velocity of light, m/s
e = 1.6e-019;     // Charge of an electron, coulombs
// Since e*V = h*c/L;     // Energy required by an
    electron to move through a potential barrier of
    one volt, joules
// solving for h
h = e*V*L/c;     // Planck's constant, Joule second
printf("\nh = %3.1e Js ", h);
// Result
// h = 6.6e-034 Js
```

Scilab code Exa 6.3 Short Wavelength Limit

```
// Scilab code: Ex6.3 : Short wavelength limit : Pg:
    156 (2008)
V = 50000;     // Potential difference, volts
h = 6.624e-034;     // Planck's constant, Js
c = 3e+08;     // Velocity of light, m/s
e = 1.6e-019;     // Charge of an electron, coulombs
// Since e*V = h*c/L;     // Energy required by an
    electron to move through a potential barrier of
    one volt, joules
// solving for L
L = h*c/(e*V);     // Short wavelength limit of X-ray
    , m
printf("\nShort wavelength limit of X-ray = %6.4f
    angstorm", L/1e-010);
// Result
// Short wavelength limit of X-ray = 0.2484 angstorm
```

Scilab code Exa 6.4 Wavelength Limit of X rays

```
// Scilab code: Ex6.4 : Wavelength limit of X-rays : Pg: 157 (2008)
V = 20000;    // Potential difference, volt
h = 6.624e-034;    // Planck's constant, Js
c = 3e+08;    // Velocity of light, m/s
e = 1.6e-019;    // Charge of an electron, coulombs
// Since e*V = h*c/L;    // Energy required by an electron to move through a potential barrier of one volt, joules
// solving for L
L = h*c/(e*V);    // Wavelength limit of X-rays, m
printf("\nShort wavelength limit of X-ray = %6.4f angstorm", L/1e-010);
// Result
// Short wavelength limit of X-ray = 0.6210 angstorm
```

Scilab code Exa 6.5 Minimum Voltage of an X ray Tube

```
// Scilab code: Ex6.5 : Minimum voltage of an X-ray tube : Pg: 157 (2008)
h = 6.625e-034;    // Planck's constant, Js
c = 3e+08;    // Velocity of light, m/s
e = 1.6e-019;    // Charge of an electron, coulombs
L = 1e-010;    // Wavelength of X-rays, m
// Since e*V = h*c/L;    // Energy required by an electron to move through a potential barrier of one volt, joules
// solving for V
V = h*c/(L*e);    // Potential difference, volts
printf("\nThe minimum voltage of an X-ray tube = %5.2f kV", V/1e+03);
// Result
// The minimum voltage of an X-ray tube = 12.42 kV
```

Scilab code Exa 6.6 Minimum Wavelength Emitted by an X ray Tube

```
// Scilab code: Ex6.6 : Minimum wavelength emitted by an X-ray tube : Pg: 157 (2008)
h = 6.625e-034;    // Planck's constant, Js
c = 3e+08;    // Velocity of light, m/s
e = 1.6e-019;    // Charge of an electron, coulombs
V = 4.5e+04;    // Accelerating potential of X-ray tube, volt
// Since e*V = h*c/L_min;    // Energy required by an electron to move through a potential barrier of one volt, joules
// solving for L_min
L_min = h*c/(V*e);    // Minimum wavelength emitted by an X-ray tube, m
printf("\nThe minimum wavelength emitted by the X-ray tube = %5.3f angstrom", L_min/1e-010);
// Result
// The minimum wavelength emitted by the X-ray tube = 0.276 angstrom
```

Scilab code Exa 6.7 Critical Voltage for Stimualted Emission

```
// Scilab code: Ex6.7: Critical voltage for stimualted emission : Pg: 158 (2008)
h = 6.625e-034;    // Planck's constant, Js
c = 3e+08;    // Velocity of light, m/s
e = 1.6e-019;    // Charge of an electron, coulombs
L_k = 0.178e-010;    // Wavelength of k absorption egde of X-rays, m
// Since e*V_critical = h*c/L;    // Energy required by an electron to move through a potential
```

```
barrier of one volt, joules
// solving for V_critical
V_critical = h*c/(L_k*e);    // Crtical voltage for
    stimulated enission, volt
printf("\nThe critical voltage for stimulated
    emission = %4.1f kV", V_critical/1e+03);
// Result
// The critical voltage for stimulated emission =
    69.8 kV
```

Chapter 7

Molecular Physics

Scilab code Exa 7.1 Frequency of Oscillation of a Hydrogen Molecule

```
// Scilab code: Ex7.1 : Frequency of oscillation of
    a hydrogen molecule: Pg: 170 (2008)
K = 4.8e+02;    // Force constant, N/m
m = 1.67e-027;    // Mass of hydrogen atom, kg
mu = m/2;    // Reduced mass of the system, kg
v = 1/(2*%pi)*sqrt(K/mu);    // Frequency of
    oscillation of a hydrogen molecule, Hz
printf("\nThe frequency of oscillation of a hydrogen
     molecule = %3.1e Hz", v);
// Result
// The frequency of oscillation of a hydrogen
    molecule = 1.2e+014 Hz
```

Scilab code Exa 7.2 Bond Length of Carbon Monoxide

```
// Scilab code: Ex7.2: bond Length of carbon
    monoxide: Pg: 170 (2008)
h = 6.626e-034;    // Planck's constant, Js
```

```
c = 2.997e+010;     // Speed of light, cm/s
B = 1.921;     // Rotational constant for CO, per cm
nu_bar = 2*B;     // Wavenumber of first line in
    rotation spectra of CO, per cm
mu = 11.384e-027;     // Reduced mass of the CO
    system, per cm
I = 2*h/(8*%pi^2*nu_bar*c);     // Moment of inertia
    of CO molecule about the axis of rotation, kg-m/s
r = sqrt(I/mu);     // Bond length of CO molecule, m
printf("\nThe bond length of CO molecule = %5.2f
    angstrom", r/1e-010);
// Result
// The bond length of CO molecule =  1.13 angstrom
```

Scilab code Exa 7.3 Intensity Ratio of J states for HCL Molecule

```
// Scilab code: Ex7.3: Intensity ratio of J states
    for HCL molecule: Pg: 171 (2008)
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
K = 1.38e-23;     // Boltzmann constant, J/K
T = 300;     // Absolute room temperature, K
J1 = 0;     // Rotational quantum number for ground
    level
J2 = 10;     // Rotational quantum number for 10th
    level
EJ1 = J1*(J1+1)*1.3e-03;     // Energy of ground
    level of HCL molecule, eV
EJ2 = J2*(J2+1)*1.3e-03;     // Energy of 10th level
    of HCL molecule, eV
// As n10/n0 is propotional to (2J+1)*exp(-(EJ2-EJ1)
    )/KT, so
I_ratio = (2*J2+1)/(2*J1+1)*exp(-(EJ2 - EJ1)/(K*T/e)
    );     // Intensity ratio of J10 and J1 states
printf("\nThe intensity ratio of J-states for HCL
    molecule = %4.2f", I_ratio);
```

```
// Result
// The intensity ratio of J−states for HCL molecule
   = 0.08
```

Scilab code Exa 7.4 CO Molecule in Lower State

```
// Scilab code: Ex7.4: CO molecule in lower state:
   Pg: 171 (2008)
R = 1.13e-010;    // Bond length of CO molecule, m
h_red = 1.054e-034;    // Reduced Planck's constant,
    Js
mu = 1.14e-026;    // Reduced mass ofthe system, kg
J = 1;    // Rotational quantum number for lowest
   state
I = mu*R^2;    // Moment of inertia of CO molecule
   about the axis of rotation, kg−metre square
EJ = J*(J + 1)*h_red^2/(2*I);    // Energy of the CO
    molecule in the lowest state, J
omega = sqrt(2*EJ/I);    // Angular velocity of the
   CO molecule in the lowest state, rad per sec
printf("\nThe energy of the CO molecule in the
   lowest state = %4.2e J", EJ);
printf("\nThe angular velocity of the CO molecule in
    the lowest state = %4.2e rad/sec", omega);
// Result
// The energy of the CO molecule in the lowest state
    = 7.63e−023 J
// The angular velocity of the CO molecule in the
   lowest state = 1.02e+012 rad/sec
```

Chapter 8

Raman Effect and Spectroscopic techniques

Scilab code Exa 8.1 `Stokes and Anti Stokes Wavelength`

```
// Scilab code: Ex8.1 : Stokes and anti stokes wavelength: Pg: 184 (2008)
c = 3e+08;    // Speed of light, m/s
Lo = 2537e-010;    // Wavelength of the exciting line, metre
Ls = 2683e-010;    // Wavelength of stokes line, metre
Lm = (Ls * Lo)/(Ls - Lo);    // Raman shift, per m
printf("\nThe Raman shift = %5.3e per cm", 1/Lm*1e-02);
Lo1 = 5461e-010;    // Wavelength of exciting line for stokes wavelength, metre
Ls = (Lm * Lo1)/(Lm - Lo1);    // Stokes wavelength for the new exciting line, metre
Las = (Lm * Lo1)/(Lm + Lo1);    // Anti-Stokes wavelength for the new exciting line, metre
printf("\nThe stokes wavelength for the new exciting line = %4d angstrom", Ls/1e-010);
printf("\nThe anti-stokes wavelength for the new
```

```
exciting line = %4d angstrom", Las/1e-010);
// Result
// The Raman shift = 2.145e+003 per cm
// The stokes wavelength for the new exciting line = 6185 angstrom
// The anti-stokes wavelength for the new exciting line = 4888 angstrom
```

Scilab code Exa 8.2 `Wvelength of Infrared Absorption Line`

```
// Scilab code: Ex8.2 : Wvelength of infrared absorption line: Pg: 185 (2008)
L1 = 4554;     // wavelength of the stokes line, angstorm
L2 = 4178;     // wavelength of antistokes line, angstorm
Lm = 2*L1*L2/[L1-L2];     // Wavelength of infrared absorption line, angstorm
printf("\nThe Wavelength of infrared absorption line = %5.3e angstorm", Lm);
// Result
// The Wavelength of infrared absorption line = 1.012e+005 angstorm
```

Chapter 9

Interaction of Charged Particles and Neutrons With Matter

Scilab code Exa 9.1 Maximum Energy Transferred by Alpha Particles

```
// Scilab Code Ex9.1 Maximum energy transferred by
   alpha particles: Pg:201 (2008)
E_alpha = 3e+06;    // Incident energy of alpha
   particles, eV
m = 9.1e-031;    // Mass of an electron, kg
M = 4*1.67e-027;    // Mass of an alpha particle, kg
// As E_alpha = 1/2*M*v^2 so E_electron = 1/2*m*(2*v
   )^2
// From the two equations
E_electron = 4*E_alpha*m/M;    // Maximum energy of
   electron, eV
printf("\nThe maximum energy transferred by alpha
   particles to the electron = %5.3f keV",
   E_electron/1e+03);
// Result
// The maximum energy transferred by alpha particles
   to the electron = 1.635 keV
```

Scilab code Exa 9.2 Rate of Energy Loss and Range of Deuteron and Alpha Particle

```
// Scilab Code Ex9.2 Rate of energy loss and range of deuteron and alpha particle: Pg:201 (2008)
E_loss_P = 59;    // Specific rate of energy loss per unit mass per unit area of proton, keV per mg cm square
R_prime_P = 50;    // Range of proton, mg per cm
Z_D = 1;    // Atomic number of deuteron
m_D = 2;    // Mass of deuteron, units
E_loss_D = Z_D^2*E_loss_P;    // Specific rate of energy loss per unit mass per unit area of deuteron, keV per mg cm square
R_prime_D = R_prime_P*m_D/Z_D^2;    // Range of deuteron, mg per cm square
Z_alpha = 2;    // Atomic number of alpha particle
m_alpha = 4;    // Mass of alpha particle, units
E_loss_alpha = Z_alpha^2*E_loss_P;    // Specific rate of energy loss per unit mass per unit area of alpha particle, keV per mg cm square
R_prime_alpha = R_prime_P*m_alpha/Z_alpha^2;    // Range of alpha particle, mg per cm square
printf("\nThe specific rate of energy loss per unit mass per unit area of deuteron = %2d keV per mg cm square", E_loss_D);
printf("\nThe range of deuteron = %3d mg per cm square", R_prime_D);
printf("\nThe specific rate of energy loss per unit mass per unit area of alpha particle = %2d keV per mg cm square", E_loss_alpha);
printf("\nThe range of alpha particle = %2d mg per cm square", R_prime_alpha);
// Result
// The specific rate of energy loss per unit mass
```

```
   per unit area of deuteron = 59 keV per mg cm
   square
// The range of deuteron = 100 mg per cm square
// The specific rate of energy loss per unit mass
   per unit area of alpha particle = 236 keV per mg
   cm square
// The range of alpha particle = 50 mg per cm square
```

Scilab code Exa 9.3 Thickness of Concrete Collimator

```
// Scilab Code Ex9.3 Thickness of concrete
   collimator: Pg:202 (2008)
rho = 2200e-03;    // Density of concrete, g per cm
mu_m = 0.064;    // Mass attenuation coefficient of
   concrete, cm square per g
mu = rho*mu_m;    // Linear attenuation coefficient
   o concrete, per cm
// As attenuation exponential is exp(-mu*x) = 1e+06,
    solving for x
x = -log(1e-06)/mu;
printf("\nThe required thickness of concrete to
   attenuate a collimated beam = %2d cm", x);
// Result
// The required thickness of concrete to attenuate a
    collimated beam = 98 cm
```

Scilab code Exa 9.4 Average Number of Collsions for Thermalization of Neutrons

```
// Scilab Code Ex9.4 Average number of collsions for
    thermalization of neutrons: Pg:202 (2008)
A = 9;    // Mass number of beryllium
xi = 2/A - 4/(3*A^2);    // Logarithmic energy
   decrement of energy distribution of neutron
```

```
E0 = 2;     // Initial energy of neutrons, MeV
En_prime = 0.025e-06;     // Thermal energy of the neutrons, MeV
n = 1/xi*log(E0/En_prime);     // Average number of collisions needed for neutrons to thermalize
En_half = 1/2*E0;     // Half of the initial energy of neutrons, MeV
n_half = 1/xi*log(E0/En_half);     // Number of collsions for half the initial energy of neutrons
printf("\nThe average number of collsions for thermalization of neutrons = %2d", n);
printf("\nThe number of collsions for half the initial energy of neutrons = %3.1f", n_half);
// Result
// The average number of collsions for thermalization of neutrons = 88
// The number of collsions for half the initial energy of neutrons = 3.4
```

Scilab code Exa 9.5 Change in Voltage Across a G M Tube

```
// Scilab Code Ex9.5 Change in voltage across a G.M. tube: Pg:202 (2008)
e= 1.6e-019;     // Charge on an electron, coulomb
W = 25;     // Ionization potential of gas (Ar/N2), eV
E = 5e+06;     // Energy of incident alpha particles, eV
C = 1e-010;     // Capacity of the system, farad
N = E/W;     // Number of ions produced
delta_V = N*e/C;     // Change in voltage across the G.M. tube, volt
printf("\nThe change in voltage across the G.M. tube = %3.1e volt", delta_V);
// Result
```

```
// The change in voltage across the G.M. tube = 3.2e
   -004 volt
```

Chapter 10

Structure of Nuclei

Scilab code Exa 10.1.1 Energy and Mass Equivalence of Wavelength

```
// Scilab Code Ex10.1.1 Energy and mass equivalence
    of wavelength: Pg:209 (2008)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
me = 9.1e-031;    // Mass of en electron, kg
L = 4.5e-013;    // Wavelength of gamma ray, m
h = 6.626e-034;    // Planck's constant, Js
c = 3e+08;    // Speed of light, m/s
U = h*c/L;    // Energy equivalence of wavelength, J
m = U/c^2;    // Mass equivalent of wavelength, kg
printf("\nThe energy equivalence of wavelength %3.1e
    m = %4.2f MeV", L, U/(e*1e+06));
printf("\nThe mass equivalence of wavelength %3.1e m
    = %4.2f me", L, m/me);
// Result
// The energy equivalence of wavelength 4.5e-013 m =
    2.76 MeV
// The mass equivalence of wavelength 4.5e-013 m =
    5.39 me
```

Scilab code Exa 10.1.2 `Binding Energy per Nucleon for Oxygen Isotopes`

```
// Scilab Code Ex10.1.2 Binding energy per nucleon for oxygen isotopes: Pg:210 (2008)
mp = 1.007276;      // Mass of proton, amu
mn = 1.008665;      // Mass of neutron, amu
amu = 931;      // Energy equivalent of 1 amu, MeV
// For Isotope O-16
M_O16 = 15.990523;      // Mass of O-16 isotope, amu
Z = 8;      // Number of protons
N = 8;      // Number of neutrons
BE = (8*(mp+mn)-M_O16)*amu;      // Binding energy of O-16 isotope, MeV
BE_bar16 = BE/(Z+N);      // Binding energy per nucleon of O-16 isotope, MeV
// For Isotope O-18
M_O18 = 17.994768;      // Mass of O-18 isotope, amu
Z = 8;      // Number of protons
N = 10;      // Number of neutrons
BE = (8*mp+10*mn-M_O18)*amu;      // Binding energy of O-18 isotope, MeV
BE_bar18 = BE/(Z+N);      // Binding energy per nucleon of O-18 isotope, MeV
printf("\nThe binding energy per nucleon of O-16 isotope = %5.3f MeV", BE_bar16);
printf("\nThe binding energy per nucleon of O-18 isotope = %5.3f MeV", BE_bar18);
// Result
// The binding energy per nucleon of O-16 isotope = 7.972 MeV
// The binding energy per nucleon of O-18 isotope = 7.763 MeV
```

Scilab code Exa 10.2.1 `Range of Alpha Emitters of Uranium`

```
// Scilab Code Ex10.2.1 Range of alpha-emitters of uranium: Pg:214 (2008)
L1 = 4.8e-018;     // Decay constant of first alpha-emitter, per sec
L2 = 4.225e+03;     // Decay constant of second alpha-emitter, per sec
L3 = 3.786e-03;     // Decay constant of third alpha-emitter, per sec
R1 = 4.19;     // Range of first alpha-emitter, cm
R2 = 7.86;     // Range of second alpha-emitter, cm
// From Geiger Nuttal law, log R = A log L + B
// Putting R1, L1 and R2, L2, subtracting and solving for A
A = log(R2/R1)/log(L2/L1);     // Slope of straight line between R and L
B = poly(0,"B");     // Intercept of straight line between R and L
B = roots(log(R2)-A*log(L2)-B);     // Other constant of Geiger-Nuttal law
R3 = exp(A*log(L3)+B);     // Range of third alpha-emitter of uranium, cm
printf("\nThe range of third alpha-emitter of uranium = %5.3f cm", R3);
// Result
// The range of third alpha-emitter of uranium = 6.554 cm
```

Scilab code Exa 10.3.1 Binding Energy per Nucleon of Helium

```
// Scilab Code Ex10.3.1 Binding energy per nucleon of helium: Pg:219 (2008)
amu = 931;     // Energy equivalent of amu, MeV
mp = 1.007895;     // Mass of proton, amu
mn = 1.008665;     // Mass of neutron, amu
M_He = 4.00260;     // Atomic weight of helium, amu
```

```
dm = 2*(mp+mn)-M_He;     // Mass difference, amu
BE = dm*amu;     // Binding energy of helium, MeV
BE_bar = BE/4;     // Binding energy per nucleon, MeV
printf("\nThe binding energy per nucleon of helium =
    %6.4f MeV", BE_bar);
// Result
// The binding energy per nucleon of helium = 7.1035
    MeV
```

Scilab code Exa 10.3.2 Energy Released in the Fusion of Deuterium

```
// Scilab Code Ex10.3.2 Energy released in the
    fusion of deuterium: Pg:220 (2008)
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
Q = 43;     // Energy released in fusion of six
    deuterium atoms, MeV
N = 6.023e+026;     // Avogadro's number, No. of
    atoms per kg
n = N/2;     // Number of atoms contained in 1 kg of
    deuterium
U = Q/6*n*e*1e+06;     // Energy released due to
    fusion of 1 kg of deuterium, J
printf("\nThe energy released due to fusion of 1 kg
    of deuterium = %5.3e J", U);
// Result
// The energy released due to fusion of 1 kg of
    deuterium = 3.453e+014 J
```

Scilab code Exa 10.3.3 Mass of Deuterium Nucleus

```
// Scilab Code Ex10.3.3 Mass of deuterium nucleus:
    Pg: 220 (2008)
amu = 1.6e-027;     // Mass of a nucleon, kg
```

```
mp = 1.007895;    // Mass of proton, amu
mn = 1.008665;    // Mass of neutron, amu
BE = 2/931;    // Binding energy of two nucleons,
    amu
M_D = (mp+mn-BE)*amu;    // Mass of a deuterium
    nucleus, kg
printf("\nThe mass of deuterium nucleus = %5.3e kg",
    M_D);
// Result
// The mass of deuterium nucleus = 3.223e-027 kg
```

Scilab code Exa 10.3.4 Binding Energy per Nucleon of Ni

```
// Scilab Code Ex10.3.4 Binding energy per nucleon
    of Ni-64: Pg: 220 (2008)
amu = 931;    // Mass of a nucleon, MeV
MH = 1.007825;    // Mass of hydrogen, amu
Me = 0.000550;    // Mass of electron, amu
Mp = MH-Me;    // Mass of proton, amu
Mn = 1.008665;    // Mass of neutron, amu
m_Ni = 63.9280;    // Mass of Ni-64 atom, amu
MNi = m_Ni-28*Me;    // Mass of ni-64 nucleus, amu
m = (28*Mp+36*Mn)-MNi;    // Mass difference, amu
BE = m*amu;    // Binding energy of Ni-64, MeV
BE_bar = BE/64;    // Binding energy per nucleon of
    Ni-64, MeV
printf("\nThe binding energy per nucleon of Ni-64 =
    %4.2f MeV", BE_bar);
// Result
// The binding energy per nucleon of Ni-64 = 8.77
    MeV
```

Scilab code Exa 10.3.5 Energy Released during Fusion of two Deuterons

```
// Scilab Code Ex10.3.5 Energy released during fusion of two deuterons: Pg: 221 (2008)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
x = 1.1;    // Binding energy per nucleon of deuterium, MeV
y = 7.0;    // Binding energy per nucleon of helium-4, MeV
E = (y - 2*x)*1e+06*e;    // Energy released when two deutron nuclei fuse together, MeV
printf("\nThe binding energy per nucleon of deuterium = %4.2e J", E);
// Result
// The binding energy per nucleon of deuterium = 7.68e-013 J
```

Scilab code Exa 10.3.6 Binding Energy and Packing Fraction of Helium

```
// Scilab Code Ex10.6 Binding energy and packing fraction of helium: Pg: 221 (2008)
amu = 931;    // Energy equivalent of amu, MeV
mp = 1.00814;    // Mass of proton, amu
mn = 1.00898;    // Mass of neutron, amu
m_He = 4.00387;    // Mass of helium, amu
A = 4;    // Mass number of helium
m = 2*(mp+mn)-m_He;    // Mass difference, amu
dm = m_He - A;    // Mass defect of He
BE = dm*amu;    // Binding energy of He, MeV
p = dm/A;    // Packing fraction of He
printf("\nThe binding energy of helium = %6.3f MeV", BE);
printf("\nThe packing fraction of helium = %5.3e", p);
// Result
// The binding energy of helium = 28.414 MeV
// The packing fraction of helium = 9.675e-004
```

Scilab code Exa 10.3.7 Mass of Yukawa Particle

```
// Scilab Code Ex10.7 Mass of Yukawa particle: Pg:
   222 (2008)
h = 6.626e-034;     // Reduced Planck's constant, Js
e = 1.6e-019;     // Charge on an electron, coulomb
R0 = 1.2e-015;     // Nuclear radius constant, m
R = 2*R0;     // Range of nuclear force, m
v = 1e+08;     // Speed of the particle, m/s
S = R;     // Distance travelled by particle within
   the nucleus, m
dt = S/v;     // time taken by the particle to travel
    across the nucleus, s
// From Heisenberg's uncertainty principle, dE.dt =
   h_bar, solving for dE
dE = h/(1e+06*e*dt);     // Energy of Yukawa paeticle
   , MeV
m = dE/0.51;     // Approximate mass of Yukawa
   particle, electronic mass unit
printf("\nThe mass of Yukawa particle = %3d me", m);
// Result
// The mass of Yukawa particle = 338 me
```

Scilab code Exa 10.3.8 Maximum Height of the Potential Barrier for Alpha Penetrati

```
// Scilab Code Ex10.8 Maximum height of the
   potential barrier for alpha penetration: Pg:222
   (2008)
epsilon_0 = 8.854e-12;     // Absolute electrical
   permittivity of free space, coulomb square per
   newton per metre square
```

```
Z = 92;    // Atomic number of U-92 nucleus
z = 2;    // Atomic number of He nucleus
e = 1.6e-019;    // Charge on an electron, coulomb
R = 9.3e-015;    // Radius of residual nucleus, m
U = 1/(4*%pi*epsilon_0)*Z*z*e^2/(R*1.6e-013);    // Maximum height of potential barrier, MeV
printf("\nThe maximum height of the potential barrier for alpha penetration = %2d MeV", U);
// Result
// The maximum height of the potential barrier for alpha penetration = 28 MeV
```

Chapter 11

Nuclear Reactions

Scilab code Exa 11.1 Energy Balance of a Nuclear Reaction

```
// Scilab code: Ex11.1 : Energy balance of a nuclear reaction: Pg: 229 (2008)
mu = 931.5;     // Energy equivalent of 1 amu, MeV
M_D = 2.0141;     // Mass of deuterium atom, amu
M_He = 3.01603;     // Mass of helium-3, amu
mn = 1.008665;     // Mass of neutron, amu
MD = (2*M_D - M_He - mn);     // Mass defect of the reaction, amu
Q = MD*mu;     // Energy balance of the nuclear reaction, MeV
printf("\nThe energy balance of the nuclear reaction = %4.2f MeV", Q);
// Result
// The energy balance of the nuclear reaction = 3.26 MeV
```

Scilab code Exa 11.2 Threshold Energy for the Reaction

```
// Scilab code: Ex11.2: Threshold energy for the reaction: Pg:229 (2008)
mu = 931.5;     // Energy equivalent of 1 amu, MeV
mx = 1.008665;     // Mass of neutron, amu
Mx = 13.003355;     // Mass of carbon atom, amu
M_alpha = 4.002603;     // Mass of alpha particle, amu
M_Be = 10.013534;     // Mass of beryllium, amu
MD = (Mx + mx - M_Be - M_alpha);     // Mass defect of the reaction, amu
Q = MD*mu;     // Q-value of the nuclear reaction, MeV
E_th = -Q*(1 + mx/Mx);     // Threshold energy for the reaction in the laboratory, MeV
printf("\nThe threshold energy of the reaction is = %4.2f MeV", E_th);
// Result
// The threshold energy of the reaction is = 4.13 MeV
```

Scilab code Exa 11.3 Gamma Ray Emission

```
// Scilab code: Ex11.3 : Gamma ray emission: Pg: 229 (2008)
h_bar = 1.0e-034;     // Order of reduced Planck's constant, Js
e = 1.0e-019;     // Order of energy equivalent of 1 eV, J/eV
tau1 = 1e-009;     // Life time of gamma ray emission, sec
tau2 = 1e-012;     // Life time of gamma ray emission, sec
W1 = h_bar/tau1;     // Full width at half maxima for tau1, eV
W2 = h_bar/tau2;     // Full width at half maxima for
```

```
    tau2, eV
printf("\nThe full width at half maxima for %1.0e =
    %1.0e eV", tau1, W1/e);
printf("\nThe full width at half maxima for %1.0e =
    %1.0e eV", tau2, W2/e);
// Result
// The full width at half maxima for 1e-009 = 1e-006
    eV
// The full width at half maxima for 1e-012 = 1e-003
    eV
```

Chapter 12

Nuclear Models

Scilab code Exa 12.1 Rate of Consumption of U235 Per Year

```
// Scilab Code Ex12.1 Rate of consumption of U-235
    per year: Pg:246 (2008)
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
amu = 1.6e-027;     // Mass of a nucleon, kg
P_out = 250e+06;     // Output power of nuclear
    reactor, J/s
E = 200e+06*e;     // Energy released per fission of
    U-235, J
n = P_out/E;     // Number of fissions per second
m = 235*amu;     // Mass of a nucleon, kg
m_sec = m*n;     // Consumption per second of U-235,
    kg
m_year = m_sec*365*24*60*60;     // Consumption per
    year of U-235, kg
printf("\nThe rate of consumption of U-235 per year
    = %5.2f kg", m_year);
// Result
// The rate of consumption of U-235 per year = 92.64
    kg
```

Scilab code Exa 12.2 Rate of Fission of U 235

```
// Scilab Code Ex12.2 Rate of fission of U-235: Pg:246 (2008)
e = 1.6e-019;      // Energy equivalent of 1 eV, J/eV
E1 = 32e+06;      // Energy released per second, J
E2 = 200e+06;      // Energy released per fission, J
N = E1/E2;      // Number of atoms undergoing fission per second
printf("\nThe number of atoms undergoing fission per second = %1.0e", N/e);
// Result
// The number of atoms undergoing fission per second = 1e+018
```

Scilab code Exa 12.3 Binding Energy of Helium Nucleus

```
// Scilab Code Ex12.3 Binding energy of helium nucleus: Pg: 247 (2008)
e = 1.6e-019;      // Energy equivalent of 1 eV, J/eV
amu = 931;      // Energy equivalent of 1 amu, MeV
m = 2*1.007825+2*1.008665-4.002603;      // Mass difference in formation of He, amu
E = m*amu;      // Energy equivalent of mass difference for He nucleus, MeV
printf("\nThe minimum energy required to break He nucleus = %5.2f MeV", E);
// Result
// The minimum energy required to break He nucleus = 28.28 MeV
```

Scilab code Exa 12.4 Energy Released During Fusion of Deuterium Nuclei

```
// Scilab Code Ex12.4 Energy released during fusion
   of deuterium nuclei: PG: 247 (2008)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
amu = 931.5;    // Energy equivalent of 1 amu, MeV
M_H = 2.014102;    // Mass of hydrogen nucleus, amu
M_He = 4.002603;    // Mass of helium nucleus, amu
m = 2*M_H-M_He;    // Mass difference, amu
E = m*amu;    // Energy released during fusion of
   deuterium nuclei, MeV
printf("\nThe energy released during fusion of
   deuterium nuclei = %6.3f MeV", E);
// Result
// The energy released during fusion of deuterium
   nuclei = 23.847 MeV
```

Scilab code Exa 12.5 Energy Required to Break One Gram Mole of Helium

```
// Scilab Code Ex12.5 Energy required to break one
   gram mole of helium: Pg: 247 (2008)
amu = 931.5;    // Energy equivalent of 1 amu, MeV
mp = 1.007825;    // Mass of proton, amu
mn = 1.008665;    // Mass of neutron, amu
M_He = 4.002603;    // Mass of helium nucleus, amu
N = 6.023e+023;    // Avogadro's number, g/mol
m = 2*mp+2*mn-M_He;    // Mass difference, amu
E1 = m*amu;    // Energy required to break one atom
   of He, MeV
E = N*E1;    // Energy required to break one gram
   mole of He, MeV
```

```
printf("\nThe energy required to break one gram mole
    of He = %5.3e MeV", E);
// Result
// The energy required to break one gram mole of He
   = 1.704e+025 MeV
```

Scilab code Exa 12.6 Energy Liberated During Production of Alpha Particles

```
// Scilab Code Ex12.6 Energy liberated during
   production of alpha particles: Pg: 248 (2008)
amu = 931;     // Energy equivalent of 1 amu, MeV
mp = 1.007825;     // Mass of proton, amu
M_Li = 7.016005;     // Mass of lithium nucleus, amu
M_He = 4.002604;     // Mass of helium nucleus, amu
dm = M_Li+mp-2*M_He;     // Mass difference, amu
disp(dm)
U = dm*amu;     // Energy liberated during production
    of two alpha particles, MeV
printf("\nThe energy liberated during production of
   two alpha particles = %5.2f MeV", U);
// Result
// The energy liberated during production of two
   alpha particles = 17.34 MeV
```

Scilab code Exa 12.7 Kinetic Energy of Neutrons

```
// Scilab Code Ex12.7 Kinetic energy of neutrons: Pg
   : 248 (2008)
d = 2.2;     // Binding energy of deuterium, MeV
H3 = 8.5;     // Binding energy of tritium, MeV
He4 = 28.3;     // Binding energy of helium, MeV
KE = He4-d-H3;     // Kinetic energy of the neutron,
   MeV
```

```
printf("\nThe kinetic energy of the neutron = %4.1f
    MeV", KE);
// Result
// The kinetic energy of the neutron = 17.6 MeV
```

Scilab code Exa 12.8 Consumption Rate of U 235

```
// Scilab Code Ex12.8 Consumption rate of U-235: Pg:
     248 (2008)
N = 6.023e+026;      // Avogadro's number, No. of
    atoms per kg
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
P = 100e+06;     // Average power generation, J/s
U = P*365*24*60*60;      // Energy required in one
    year, J
U1 = 180e+06*e;      // Energy produced by one atom
    fission of U-235
n = U/U1;     // Number of atoms required to produce
    energy in one year
M = n*235/N;      // Mass of U-235 required per year,
    kg
printf("\nThe rate of consumption of U-235 per year
    = %7.4f kg", M);
// Result
// The rate of consumption of U-235 per year =
    42.7237 kg
```

Scilab code Exa 12.9 Minimum Disintegraton Energy of Nucleus

```
// Scilab Code Ex12.9 Minimum disintegraton energy
    of nucleus: Pg: 249 (2008)
mn = 1.008665;      // Mass of neutron, amu
mp = 1.007276;      // Mass of proton, amu
```

```
amu = 931;     // Energy equivalent of 1 amu, MeV
BE = 2.21;     // Binding energy of deutron nucleus ,
    MeV
E = BE/amu;     // Binding energy of deutron nucleus ,
     amu
M_D = mp+mn-E;     // Mass of deuterium nucleus , amu
printf("\nThe mass of deuterium nucleus = %8.6f amu"
    , M_D);
// Result
// The mass of deuterium nucleus = 2.013567 amu
```

Scilab code Exa 12.10 `Rate of Fission of U 235`

```
// Scilab Code Ex12.10 Rate of fission of U-235 : Pg
    : 249 (2008)
N = 6.023e+026;     // Avogadro's number, No. of
    atoms per kg
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
P = 1;     // Average power generation , J/s
U = P*365*24*60*60;     // Energy required in one
    year , J
U1 = 200e+06*e;     // Energy produced by one atom
    fission of U-235
n = U/U1;     // Number of atoms undergoing fission
    per year
M = n/N;     // Mass of U-235 required per year , kg
printf("\nThe rate of fission of U-235 per year = %5
    .3e kg", M);
// Result
// The rate of fission of U-235 per year = 1.636e
    -009 kg
```

Scilab code Exa 12.11 `Energy Released During Fission of U 235`

```
// Scilab Code Ex12.11 Energy released during fission of U-235: Pg: 250 (2008)
N = 6.023e+023;     // Avogadro's number
A = 235;     // Mass number of U-235
n = N/235;     // Number of atoms in 1g of U-235
E = 200;     // Energy produced by fission of 1 U-235 atom, MeV
U = n*E;     // Energy produced by fission of 1g of U-235 atoms, MeV
printf("\nThe energy produced by fission of 1g of U-235 atoms = %5.3e MeV", U);
// Result
// The energy produced by fission of 1g of U-235 atoms = 5.126e+023 MeV
```

Scilab code Exa 12.12 `Minimum Energy of Gamma Photon for Pair Production`

```
// Scilab Code Ex12.12 Minimum energy of gamma photon for pair production: Pg: 250 (2008)
c = 3.0e+08;     // Speed of light, m/s
me = 9.1e-031;     // Mass of electron, kg
e = 1.6e-019;     // Energy equivalent of 1 eV, J/eV
mp = me;     // Mass of positron, kg
U = (me+mp)*c^2/(e*1e+06);     // Energy of gamma-ray photon, MeV
printf("\nThe energy of gamma-ray photon = %5.3f MeV", U);
// Result
// The energy of gamma-ray photon = 1.024 MeV
```

Scilab code Exa 12.13 `Uranium Atom Undergoing Fission in a Reactor`

```
// Scilab Code Ex12.13 Uranium atom undergoing fission in a reactor: Pg: 250 (2008)
P_out = 800e+06;    // Output power of the reactor, J/s
E1 = P_out*24*60*60;    // Energy required one day, J
eta = 0.25;    // Efiiciency of reactor
N=poly(0,"N");    // Declare N as the variable
E2 = N*200e+06*1.6e-019*eta;    // Useful energy produced by N atoms in a day, J
N=roots(E2-E1);         // Number of U-235 atoms consumed in one day
m = N*235/6.023e+026;    // Mass of uranium consumption in one day, kg
printf("\nThe number of U-235 atoms consumed in one day = %4.2e atoms", N);
printf("\nThe mass of uranium consumption in one day = %4.2f kg", m);

// Result
// The number of U-235 atoms consumed in one day = 8.64e+024 atoms
// The mass of uranium consumption in one day = 3.37 kg
```

Scilab code Exa 12.14 Amount of Uranium Fuel Required For One Day Operation

```
// Scilab Code Ex12.14 Amount of uranium fuel required for one day operation: Pg: (2008)
e = 1.6e-019;    // Energy equivalent of 1 eV, J/eV
eta = 0.20;    // Efficiency of the nuclear reactor
E1 = 100e+06*24*60*60;    // Average energy required per day, J
m = poly(0,"m");    // Suppose amount of fuel required be m kg
```

```
n = m*6.023e+026/235;     // Number of uranium atoms
E = 200e+06*e;     // Energy released per fission of
   U-235, J
U = E*n;     // Total energy released by fission of U
   -235, J
E2 = U*eta;     // Useful energy produced by n atoms
   in a day, J
m = roots(E2-E1);
printf("\nThe mass of uranium fuel required for one
   day operation = %6.4f kg/day", m);
// Result
// The mass of uranium fuel required for one day
   operation = 0.5267 kg/day
```

Scilab code Exa 12.15 Binding Energy of Fe Using Weizsaecker Formula

```
// Scilab Code Ex12.15 Binding energy of Fe using
   Weizsaecker formula: Pg: 251 (2008)
amu = 931.5;     // Energy equivalent of 1 amu, MeV
A = 56;     // Mass number of Fe
Z = 26;     // Atomic number of Fe
av = 15.7;     // Binding energy per nucleon due to
   volume effect, MeV
as = 17.8;     // Surface energy constant, MeV
ac = 0.711;     // Coulomb energy constant, MeV
aa = 23.7;     // asymmetric energy constant, MeV
ap = 11.18;     // Pairing energy constant, MeV
BE = av*A - as*A^(2/3) - ac*Z^2*A^(-1/3)-aa*(A-2*Z)
   ^2*A^(-1)+ap*A^(-1/2);     // Weizsaecker
   Semiempirical mass formula
M_Fe = 55.939395;     // Atomic mass of Fe-56
mp = 1.007825;     // Mass of proton, amu
mn = 1.008665;     // Mass of neutron, amu
E_B = (Z*mp+(A-Z)*mn-M_Fe)*amu;     // Binding energy
    of Fe-56, MeV
```

```
printf("\nThe binding energy of Fe-56 using Weizsaecker formula = %6.2f MeV", BE);
printf("\nThe binding energy of Fe-56 using mass defect = %6.2f MeV", E_B);
printf("\nThe result of the semi empirical formula agrees with the experimental value within %3.1f percent", abs((BE-E_B)/BE*100));
// Result
// The binding energy of Fe-56 using Weizsaecker formula = 487.75 MeV
// The binding energy of Fe-56 using mass defect = 488.11 MeV
// The result of the semi empirical formula agrees with the experimental value within 0.1 percent
```