

Scilab Textbook Companion for
Modern Physics for Scientists and Engineers
by S. T. Thornton and A. Rex¹

Created by
Juhi Rajput
Modern Physics
Physics

Shri Mata Vaishno Devi University
College Teacher
None

Cross-Checked by
Bhavani Jalkrish

July 30, 2019

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

Book Description

Title: Modern Physics for Scientists and Engineers

Author: S. T. Thornton and A. Rex

Publisher: Cengage Learning, USA

Edition: 4

Year: 2013

ISBN: 978-1-133-10372-1

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes	4
2 Special Theory of Relativity	6
3 The Experimental Basis of Quantum Physics	16
4 Structure of the Atom	27
5 Wave Properties of Matter and Quantum Mechanics I	34
6 Quantum Mechanics II	42
7 The Hydrogen Atom	49
8 Atomic Physics	56
9 Statistical Physics	59
10 Molecules Lasers and Solids	65
11 Semiconductor Theory and Devices	71
12 The Atomic Nucleus	76
13 Nuclear Interactions and Applications	89

14 Particle Physics	97
15 General Relativity	101
16 Cosmology and Modern Astrophysics	104

List of Scilab Codes

Exa 2.2	Speed of the aircraft	6
Exa 2.3	Speed of the aircraft from the standpoint of length contraction	7
Exa 2.4	Speed of the aircraft from the standpoint of length contraction	7
Exa 2.6	Time loss of an atomic clock	8
Exa 2.8	Relativistic doppler effect in twin paradox .	10
Exa 2.11	Accelerating electrons to produce X rays . .	10
Exa 2.13	Head on collision of two protons	11
Exa 2.15	Minimum kinetic energy of the protons in head on collision	12
Exa 2.16	Binding energy of He nucleus	13
Exa 2.17	Fractional mass increase of the Na and Cl atoms	14
Exa 2.18	Kinetic energy and the mass of sigma particle	14
Exa 3.1	A moving electron subjected to electric and magnetic fields	16
Exa 3.3	Hydrogen series of spectral lines	17
Exa 3.4	Maximum wavelength emitted from a heated furnace	19
Exa 3.5	Sun as a blackbody	20
Exa 3.10	Exposure time of light to produce a photo- electron of given kinetic energy	21
Exa 3.11	Photoelectric effect for lithium	22
Exa 3.12	Lithium exposed to light radiation	23
Exa 3.13	Number of photons in the light beam of given wavelength and intensity	24
Exa 3.15	Minimum wavelength of the X rays	24

Exa 3.16	X ray scattering from the gold target	25
Exa 4.1	Maximum scattering angle in Geiger and Marsden experiment	27
Exa 4.2	Fraction of alpha particles deflected from a gold foil	27
Exa 4.3	Fraction of alpha particles deflected from gold foil at a given angle	28
Exa 4.5	Size of the nucleus	29
Exa 4.6	Nonrelativistic justification for the speed of the electron	30
Exa 4.7	Longest and shortest wavelengths observed in Paschen series for hydrogen	30
Exa 4.8	Wavelengths of H alpha lines for three isotopes of hydrogen	31
Exa 4.9	Shortest wavelength emitted by doubly positive Li ion	33
Exa 5.1	The wavelength of the X rays incident on rock salt	34
Exa 5.2	de Broglie wavelength of a tennis ball and an electron	35
Exa 5.3	de Broglie wavelength of an electron used by Davisson and Germer	35
Exa 5.4	Wavelength of a neutron at different temperatures	36
Exa 5.7	Distance between first two maxima in fringe pattern	37
Exa 5.8	Momentum uncertainty for a tennis ball and an electron	38
Exa 5.9	Minimum kinetic energy of an electron in hydrogen atom	38
Exa 5.10	Minimum kinetic energy of electron localized within a typical nuclear radius	39
Exa 5.12	Energy width of excited state of atom and uncertainty ratio of frequency of emitted photon	40
Exa 5.13	Energy at different levels	40
Exa 6.4	Probabilities of a particle in the given regions	42
Exa 6.9	Transition energy for a proton confined to a nucleus	43

Exa 6.14	Fraction of electrons tunneling through a barrier	43
Exa 6.15	Probability of electron tunneling through the barrier	44
Exa 6.16	A particle penetrating through a potential step	45
Exa 6.17	An alpha particle tunnelling through a nucleus	46
Exa 7.2	Normalization of hydrogen wave function . .	49
Exa 7.4	Degeneracy of M level in hydrogen atom . .	49
Exa 7.7	Energy difference between components of p states of atomic hydrogen placed in an external field	51
Exa 7.8	Separation of the atomic beam	51
Exa 7.9	Number of distinct states for the 4d level of atomic hydrogen	52
Exa 7.10	Energy of allowed transitions for the hydrogen atom	53
Exa 7.13	Probability of the electron in the 1s state of the hydrogen atom	55
Exa 8.3	Splitting of 3p subshell of sodium	56
Exa 8.5	LS coupling of two electrons in an atom . .	57
Exa 8.8	Internal magnetic field causing spin orbit splitting	58
Exa 9.1	Mean translational kinetic energy of gas molecules at room temperature	59
Exa 9.3	Mean molecular speed in light gas hydrogen and heavy radon gas	60
Exa 9.4	Fraction of molecules in an ideal gas having speed near to the most probable speed . . .	61
Exa 9.6	Relative number of atoms in the ground and first excited states in atomix hydrogen . . .	61
Exa 9.7	Fermi energy and Fermi temperature for copper	62
Exa 9.8	Electronic contribution to the molar heat capacity of metals	63
Exa 10.1	Energy of lowest rotational state of nitrogen gas	65
Exa 10.2	Vibrational energy levels of HCl molecule .	66

Exa 10.4	Range parameter for NaCl	67
Exa 10.5	Induced diamagnetism in an atom	67
Exa 10.6	Paramagnetism in a typical material	68
Exa 10.7	Superconductivity in niobium	69
Exa 10.8	Magnetic field perpendicular to the loop	70
Exa 11.1	Relative number of electrons with given energies above the valence band	71
Exa 11.3	Hall effect in zinc strip	72
Exa 11.4	Current through a reverse bias pn junction diode	73
Exa 11.5	Energy produced by a solar cell per day	73
Exa 11.6	Data bits stored into CD ROM	74
Exa 12.1	Minimum kinetic energy of a proton in a medium sized nucleus	76
Exa 12.2	Nuclear radius of calcium	77
Exa 12.3	Radii of U238 and He4 nuclei	78
Exa 12.4	A proton subjected to the magnetic field	78
Exa 12.5	Binding energy of Be	79
Exa 12.6	Total Coulomb energy of U238	80
Exa 12.8	Binding energy per nucleon	81
Exa 12.10	Radiative decay of Po210	82
Exa 12.11	Time of decay of F18 isotope	82
Exa 12.12	Alpha activity of 10 kg sample of U235	83
Exa 12.13	Non emission of a neutron by U230	83
Exa 12.15	Possible reaction with Fe55 isotope	84
Exa 12.16	Allowed decay modes for Ac226	85
Exa 12.17	Error introduced in the gamma ray energy	87
Exa 12.18	Age of the uranium ore	87
Exa 12.19	C14 dating to determine age of bone	88
Exa 13.1	Number of neutrons produced in collision of alpha particle and carbon target	89
Exa 13.2	Likelihood of a neutron production than a proton	90
Exa 13.3	Nuclear reaction observed by Rutherford	90
Exa 13.4	Final energy of excitation of product nucleus in the nuclear reaction	91
Exa 13.5	Ground state Q value of the induced fission reaction	92

Exa 13.6	Excitation energy of the compound nuclei . . .	92
Exa 13.7	Nuclear fission through neutron capture . . .	93
Exa 13.8	Fusion reaction in supergiant stars	94
Exa 13.9	Ignition temperature needed for the fusion re- action between a deuterium and a tritium . . .	94
Exa 13.10	Neutron beam study of atomic structures . . .	95
Exa 14.1	Mass of the meson from Heisenberg uncer- tainty principle	97
Exa 14.2	Range of the weak interaction	98
Exa 14.10	Fixed target accelerators	98
Exa 14.11	Energy required by a fixed target accelera- tor to match with that available for colliding beams at LHC	99
Exa 15.1	Gravitational time dilation effect	101
Exa 15.2	Schwarzschild radius for the sun and the earth	102
Exa 15.3	Time taken by a black hole to radiate its en- ergy	103
Exa 16.1	Hubble constant determination	104
Exa 16.2	Current ratio of protons to neutrons in the universe	104
Exa 16.3	Ratio of protons to neutrons at 10 billion kelvin temperature of the universe	105
Exa 16.4	Mean temperature of the sun	106
Exa 16.5	Radius of the neutron star	107
Exa 16.6	Redshift versus recession velocity	107
Exa 16.7	Difference in the travel times of different mass neutrinos	109
Exa 16.8	Critical density of the universe	110
Exa 16.9	Upper limit of the age of the universe	110

List of Figures

16.1 Redshift versus recession velocity	108
---	-----

Chapter 2

Special Theory of Relativity

Scilab code Exa 2.2 Speed of the aircraft

```
1 // Scilab Code Ex2.2 : Page-34 (2013)
2 clc; clear;
3 ly = 9.46e+015; // Distance travelled by light in
   an year, m
4 c = 3e+008; // Speed of light, m/s
5 L = 4.30*ly; // Distance of Alpha Centauri from
   earth, m
6 T0 = 16*365.25*24*60*60; // Proper time in system
   K_prime, s
7 // As time measured on earth,  $T = 2*L/v = T0\_prime/$ 
    $\sqrt{1-(v/c)^2}$ , solving for v
8 v = sqrt(4*L^2/(T0^2+4*L^2/c^2)); // Speed of the
   aircraft, m/s
9 gama = 1/sqrt(1-(v/c)^2); // Relativistic factor
10 T = gama*T0/(365.25*24*60*60); // Time interval
   as measured on Earth, y
11 printf("\\nThe speed of the aircraft = %4.2e m/s", v)
   ;
12 printf("\\nThe time interval as measured on earth =
   %4.1f y", T);
13
```

```

14 // Result
15 // The speed of the aircraft = 1.42e+008 m/s
16 // The time interval as measured on earth = 18.2 y

```

Scilab code Exa 2.3 Speed of the aircraft from the standpoint of length contraction

```

1 // Scilab Code Ex2.3 : Page-38 (2013)
2 clc; clear;
3 L0 = 4.30; // Distance of Alpha Centauri from
   earth, ly
4 c = 3e+008; // Speed of light, m/s
5 T = 8; // Proper time in system K_prime, y
6 // As  $v/c = L0 \cdot \sqrt{1 - (v/c)^2} / (c \cdot T)$  or  $\text{bita} = L0 \cdot$ 
    $\sqrt{1 - \text{bita}^2} / (c \cdot T)$ , solving for bita
7 bita = sqrt(L0^2 / (T^2 + L0^2)); // Boost
   parameter
8 v = L0 * sqrt(1 - bita^2) / T; // Speed of the aircraft
   , c units
9 printf("\nThe boost parameter = %5.3f", bita);
10 printf("\nThe speed of the aircraft = %5.3f c units"
   , v);
11
12 // Result
13 // The boost parameter = 0.473
14 // The speed of the aircraft = 0.473 c units

```

Scilab code Exa 2.4 Speed of the aircraft from the standpoint of length contraction

```

1 // Scilab Code Ex2.4 : Page-40 (2013)
2 clc; clear;
3 c = 1; // For simplicity assume speed of light to
   be unity, m/s
4 bita = 0.600; // Boost parameter

```

```

5 gama = 1/sqrt(1-bit^2); // Relativistic factor
6 u_x_prime = 0; // Speed of the protons in
    spaceship frame along x-axis, m/s
7 u_y_prime = 0.99*c; // Speed of the protons in
    spaceship frame along y-axis, m/s
8 u_z_prime = 0; // Speed of the protons in
    spaceship frame along z-axis, m/s
9 v = 0.60*c; // Speed of the spaceship w.r.t.
    space station, m/s
10 u_x = (u_x_prime + v)/(1 + v/c^2*u_x_prime); //
    Speed of protons in space station frame along x-
    axis, m/s
11 u_y = u_y_prime/(gama*(1 + v/c^2*u_x_prime)); //
    Speed of protons in space station frame along y-
    axis, m/s
12 u_z = u_z_prime/(gama*(1 + v/c^2*u_x_prime)); //
    Speed of protons in space station frame along y-
    axis, m/s
13 u = sqrt(u_x^2 + u_y^2 + u_z^2); // The speed of
    the protons measured by an observer in the space
    station, m/s
14 printf("\nThe speed of the protons measured by an
    observer in the space station = %5.3f c units", u
    );
15
16 // Result
17 // The speed of the protons measured by an observer
    in the space station = 0.994 c units

```

Scilab code Exa 2.6 Time loss of an atomic clock

```

1 // Scilab Code Ex2.6 : Page-45 (2013)
2 clc; clear;
3 c = 2.998e+008; // Speed of light in free space,
    m/s

```

```

4 v = 7712;      // Speed of the space shuttle , m/s
5 bita = v/c;    // Boost parameter
6 T_loss = 295.02; // Total measured loss in time,
   ps/sec
7 Add_T_loss = 35.0; // Additional time loss for
   moving clock from general relativity prediction ,
   ps/s
8 // From time dilation relation ,  $T_0\text{-prime} = T\sqrt{1 - \text{bita}^2}$ , solving for  $(T - T_0\text{-prime})/T$ 
9 Calc_T_loss = bita^2/2*1e+012; // Expected time
   lost per sec by the moving clock , ps/sec
10 Measured_T_loss = T_loss + Add_T_loss; // Total
   measured loss in time per sec as per special
   relativity , ps/s
11 percent_T_loss = (Calc_T_loss - Measured_T_loss)/
   Calc_T_loss*100; // Percentage deviation of
   measured and the calculated time loss per sec
12 T = 6.05e+05; // Total time of the seven-day
   mission , s
13 delta_T = Calc_T_loss*1e-012*T; // The total time
   difference between the moving and stationary
   clocks during the entire shuttle flight , s
14 printf("\nThe expected time lost per second for the
   moving clock = %6.2f ps", Calc_T_loss);
15 printf("\nThe percentage deviation of measured and
   the calculated time loss per sec for moving clock
   = %3.1f percent", percent_T_loss);
16 printf("\nThe total time difference between the
   moving and stationary clocks during the entire
   shuttle flight = %3.1f ms", delta_T/1e-003);
17
18 // Result
19 // The expected time lost per second for the moving
   clock = 330.86 ps
20 // The percentage deviation of measured and the
   calculated time loss per sec for moving clock =
   0.3 percent
21 // The total time difference between the moving and

```

```
stationary clocks during the entire shuttle  
flight = 0.2 ms
```

Scilab code Exa 2.8 Relativistic doppler effect in twin paradox

```
1 // Scilab Code Ex2.8 : Page-57 (2013)
2 clc; clear;
3 f0 = 1; // For simplicity assume frequency of the
          signals sent by Frank, Hz
4 // Outbound trip
5 bita = -0.8; // Boost parameter for receding
              frames
6 f = sqrt(1+bita)/sqrt(1-bita)*f0; // The
          frequency of the signals received by Mary in
          outbound trip, Hz
7 printf("\\nThe frequency of the signals received by
          Mary in outbound trip = f0/%d", ceil(f*9));
8 // Return trip
9 bita = +0.8; // Boost parameter for approaching
              frames
10 f = sqrt(1+bita)/sqrt(1-bita)*f0; // The
          frequency of the signals received by Mary in
          return trip, Hz
11 printf("\\nThe frequency of the signals received by
          Mary in return trip = %df0", f);
12
13 // Result
14 // The frequency of the signals received by Mary in
          outbound trip = f0/3
15 // The frequency of the signals received by Mary in
          return trip = 3f0
```

Scilab code Exa 2.11 Accelerating electrons to produce X rays


```

1 // Scilab Code Ex2.11: Page-64 (2013)
2 clc; clear;
3 q = 1.6e-019; // Charge on an electron , C
4 V = 25e+003; // Accelerating potential , volt
5 K = q*V; // Kinetic energy of electrons , J
6 m = 9.11e-031; // Rest mass of an electron , kg
7 c = 3.00e+08; // Speed of light , m/s
8 // From relativistic kinetic energy formula,  $K = (\text{gama} - 1)*m*c^2$ , solving for gama
9 gama = 1 + K/(m*c^2); // Relativistic factor
10 bita = sqrt((gama^2-1)/gama^2); // Boost
    parameter
11 u = bita*c; // Speed of the electrons , m/s
12 // From non-relativistic expression,  $K = 1/2*m*u^2$ ,
    solving for u
13 u_classical = sqrt(2*K/m); // Non-relativistic
    speed of the electrons , m/s
14 u_error = (u_classical - u)/u_classical*100; //
    Percentage error in speed of electrons , m/s
15 printf("\\nThe relativistic speed of the accelerated
    electrons = %4.2e m/s", u);
16 printf("\\nThe classical speed is about %d percent
    greater than the relativistic speed", ceil(
    u_error));
17
18 // Result
19 // The relativistic speed of the accelerated
    electrons = 9.04e+007 m/s
20 // The classical speed is about 4 percent greater
    than the relativistic speed

```

Scilab code Exa 2.13 Head on collision of two protons

```

1 // Scilab Code Ex2.13: Page-69 (2013)
2 clc; clear;

```

```

3 c = 1;      // For simplicity assume speed of light to
    be unity, m/s
4 K = 2.00;   // Kinetic energy of protons, GeV
5 E0 = 0.938; // Rest mass of a proton, GeV
6 E = K + E0; // Total energy of the proton, GeV
7 // From relativistic mass energy relation,  $E^2 = (p*c)^2 + E0^2$ , solving for p
8 p = sqrt(E^2 - E0^2)/c; // Momentum of the
    protons, GeV/c
9 // As  $E = \text{gama} * E0$ , solving for gama
10 gama = E/E0; // Relativistic factor
11 bita = sqrt((gama^2-1)/gama^2); // Boost
    parameter
12 v = bita*3.00e+08; // Speed of 2 GeV proton, m/s
13 printf("\nThe energy of each initial proton = %5.3f
    GeV", E);
14 printf("\nThe momentum of each initial proton = %4.2
    f GeV/c", p);
15 printf("\nThe speed of each initial proton = %3.1e m
    /s", v);
16 printf("\nThe relativistic factor, gama = %4.2f",
    gama);
17 printf("\nThe boost parameter, beta = %5.3f", bita);
18
19 // Result
20 // The energy of each initial proton = 2.938 GeV
21 // The momentum of each initial proton = 2.78 GeV/c
22 // The speed of each initial proton = 2.8e+008 m/s
23 // The relativistic factor, gama = 3.13
24 // The boost parameter, beta = 0.948

```

Scilab code Exa 2.15 Minimum kinetic energy of the protons in head on collision

```

1 // Scilab Code Ex2.15: Page-71 (2013)
2 clc; clear;

```

```

3 E_d = 1875.6;    // Rest mass energy of the
    deuterium , MeV
4 E_pi = 139.6;   // Rest mass energy of the pion ,
    MeV
5 E_p = 938.3;    // Rest mass energy of the proton ,
    MeV
6 K = 1/2*(E_d + E_pi - 2*E_p);    // Minimum kinetic
    energy of the protons , MeV
7 printf("\nThe minimum kinetic energy of the protons
    = %2d MeV", K);
8
9 // Result
10 // The minimum kinetic energy of the protons = 69
    MeV

```

Scilab code Exa 2.16 Binding energy of He nucleus

```

1 // Scilab Code Ex2.16: Page-72 (2013)
2 clc; clear
3 u = 931.5;    // Energy equivalent of 1 amu, MeV
4 c = 1;    // Speed of light in vacuum, m/s
5 m_e = 0.000549*u;    // Rest mass of an electron ,
    MeV/c^2
6 m_p = 1.007276*u;    // Rest mass of a proton , MeV/c
    ^2
7 m_n = 1.008665*u;    // Rest mass of a neutron , MeV/
    c^2
8 m_He = 4.002603*u;    // Rest mass of a helium , MeV/
    c^2
9 M_He = m_He - 2*m_e;    // Mass of He nucleus , MeV/c
    ^2
10 E_B_He = (2*m_p + 2*m_n - M_He)*c^2;    // Binding
    energy of the He nucleus , MeV
11 printf("\nThe binding energy of the He nucleus = %4
    .1f MeV", E_B_He);

```

```

12
13 // Result
14 // The binding energy of the He nucleus = 28.3 MeV

```

Scilab code Exa 2.17 Fractional mass increase of the Na and Cl atoms

```

1 // Scilab Code Ex2.17: Page-72 (2013)
2 clc; clear
3 u = 931.5; // Energy equivalent of 1 amu, MeV/u
4 c = 1; // For simplicity assume speed of light in
// vacuum to be unity, m/s
5 E_B = 4.24; // The dissociation energy of the NaCl
// molecule, MeV
6 M = 58.44*u; // Energy corresponding to molecular
// mass of NaCl, MeV
7 f_r = E_B/M; // The fractional mass increase of
// the Na and Cl atoms
8 printf("\nThe fractional mass increase of the Na and
// Cl atoms when they are not bound together in
// NaCl = %4.1e", f_r/1e+006);
9
10 // Result
11 // The fractional mass increase of the Na and Cl
// atoms when they are not bound together in NaCl =
// 7.8e-011

```

Scilab code Exa 2.18 Kinetic energy and the mass of sigma particle

```

1 // Scilab Code Ex2.18: Page-72 (2013)
2 clc; clear
3 c = 1; // For simplicity assume speed of light in
// vacuum to be unity, m/s
4 E0_n = 940; // Rest energy of a neutron, MeV

```

```

5 E0_pi = 140;    // Rest energy of a pion , MeV
6 p_n = 4702;    // Momentum of the neutron , MeV/c
7 p_pi = 169;    // Momentum of the pion , MeV/c
8 E_n = sqrt((p_n*c)^2+E0_n^2);    // Total energy of
  the neutron , MeV
9 E_pi = sqrt((p_pi*c)^2+E0_pi^2);    // Total energy
  of the pion , MeV
10 E = E_n + E_pi;    // Total energy of the reaction ,
  MeV
11 p_sigma = p_n + p_pi;    // Momentum of the sigma
  particle , MeV/c
12 E0_sigma = sqrt(E^2 - (p_sigma*c)^2);    // Rest
  mass energy of the sigma particle , MeV
13 m_sigma = E0_sigma/c^2;    // Rest mass of sigma
  particle , MeV/c^2;
14 K = E - E0_sigma;    // Kinetic energy of sigma
  particle , MeV
15 printf("\nThe rest mass of sigma particle = %4d MeV/
  c^2", ceil(m_sigma));
16 printf("\nThe kinetic energy of sigma particle = %4d
  MeV", ceil(K));
17
18 // Result
19 // The rest mass of sigma particle = 1192 MeV/c^2
20 // The kinetic energy of sigma particle = 3824 MeV
21 // The answers are given wrongly in the textbook

```

Chapter 3

The Experimental Basis of Quantum Physics

Scilab code Exa 3.1 A moving electron subjected to electric and magnetic fields

```
1 // Scilab Code Ex3.1: Page-87 (2013)
2 clc; clear
3 E = 1.2e+004; // Electric field , V/m
4 B = 8.8e-004; // Magnetic field , T
5 l = 0.05; // Length of the deflection plates , m
6 v0 = E/B; // Initial velocity of the electron , m/
   s
7 theta = 30; // Angular deflection of the electron
   , degrees
8 q_ratio_m = E*tand(theta)/(B^2*l); // Specific
   charge of the electron , C/kg
9 printf("\\nThe initial velocity of the electron = %3
   .1e m/s", v0);
10 printf("\\nThe specific charge of the electron = %3.1
   e C/kg", q_ratio_m);
11
12 // Result
13 // The initial velocity of the electron = 1.4e+007 m
   /s
```

```
14 // The specific charge of the electron = 1.8e+011 C/
    kg
```

Scilab code Exa 3.3 Hydrogen series of spectral lines

```
1 // Scilab Code Ex3.3: Page-94 (2013)
2 clc; clear
3 function flag = check_visible(lambda)
4     if lambda >= 400 & lambda < 700 then
5         flag = 1;
6     else flag = 0;
7     end
8 endfunction
9 R_H = 1.0968e+007; // Rydberg constant, per metre
10 f = zeros(7);
11 // Lyman series
12 printf("\nFor Lyman series, the wavelengths are:\n")
13 n = 1; // The lowest level of Lyman series
14 for k = 2:1:3
15     lambda = 1/(R_H*(1/n^2-1/k^2))/1e-009;
16     printf("k = %d, %5.1f nm", k, lambda);
17     f(k) = check_visible(lambda);
18     if f(k) == 1 then
19         printf(" (Visible) \n");
20     else
21         printf(" (Ultraviolet)\n");
22     end
23 end
24 if f(1) == 1 | f(2) == 1 | f(3) == 1 then
25     printf("Some wavelengths of Lyman series
26         fall in the visible region.\n")
27 else
28     printf("All the wavelengths of Lyman series
29         fall in the UV-region.\n")
30 end
```

```

29
30 // Balmer series
31 printf("\nFor Balmer series , the wavelengths are:\n"
    )
32 n = 2; // The lowest level of Balmer series
33 for k = 3:1:7
34     lambda = 1/(R_H*(1/n^2-1/k^2))/1e-009;
35     printf("k = %d, %5.1f nm", k, lambda);
36     f(k) = check_visible(lambda);
37     if f(k) == 1 then
38         printf(" (Visible) \n");
39     else
40         printf(" (Ultraviolet)\n");
41     end
42 end
43
44 // Paschen series
45 printf("\nFor Paschen series , the wavelengths are:\n"
    ")
46 n = 3; // The lowest level of Lyman series
47 for k = 4:1:5
48     lambda = 1/(R_H*(1/n^2-1/k^2))/1e-009;
49     printf("k = %d, %5.1f nm", k, lambda);
50     f(k) = check_visible(lambda);
51     if f(k) == 1 then
52         printf(" (Visible) \n");
53     else
54         printf(" (Infrared)\n");
55     end
56 end
57 // For limiting member
58 k = %inf;
59 lambda = 1/(R_H*(1/n^2-1/k^2))/1e-009;
60 printf("k = %d, %5.1f nm", %inf, lambda);
61 f(6) = check_visible(lambda);
62 if f(6) == 1 then
63     printf(" (Visible) \n");
64 else

```



```

65         printf(" (Infrared)\n");
66     end
67     if f(4) == 1 | f(5) == 1 | f(6) == 1 then
68         printf("Some wavelengths of Paschen series
69             fall in the visible region.")
70     else
71         printf("All the wavelengths of Paschen
72             series fall in the IR-region.")
73     end
74 // Result
75 // For Lyman series , the wavelengths are:
76 // k = 2, 121.6 nm (Ultraviolet)
77 // k = 3, 102.6 nm (Ultraviolet)
78 // All the wavelengths of Lyman series fall in the
79 // UV-region.
80 // For Balmer series , the wavelengths are:
81 // k = 3, 656.5 nm (Visible)
82 // k = 4, 486.3 nm (Visible)
83 // k = 5, 434.2 nm (Visible)
84 // k = 6, 410.3 nm (Visible)
85 // k = 7, 397.1 nm (Ultraviolet)
86 // For Paschen series , the wavelengths are:
87 // k = 4, 1875.6 nm (Infrared)
88 // k = 5, 1282.1 nm (Infrared)
89 // k = Inf, 820.6 nm (Infrared)
90 // All the wavelengths of Paschen series fall in the
91 // IR-region.

```

Scilab code Exa 3.4 Maximum wavelength emitted from a heated furnace

```

1 // Scilab Code Ex3.4: Page-98 (2013)
2 clc; clear

```

```

3 T = 1600 + 273;      // Temperature of the furnace , K
4 b = 2.898e-003;     // Wein's constant , m-K
5 lambda_max = ceil(b/(T*1e-009)); // Maximum
    wavelength from Wein's Displacement Law, nm
6 printf("\nThe maximum wavelength emitted from the
    heated furnace = %4d nm", lambda_max);
7 printf("\nThis wavelength falls in the IR-region.");
8
9 // Result
10 // The maximum wavelength emitted from the heated
    furnace = 1548 nm
11 // This wavelength falls in the IR-region.

```

Scilab code Exa 3.5 Sun as a blackbody

```

1
2 // Scilab Code Ex3.5: Page-98 (2013)
3 clc; clear
4 lambda_max = 500e-009; // Maximum intensity
    wavelength emitted by the sun, m
5 b = 2.898e-003; // Wein's constant , m-K
6 sigma = 5.67e-008; // Stefan's constant , W/Sq.m-K
    ^4
7 r = 6.96e+008; // Radius of the sun, m
8 r_E = 6.37e+006; // Radius of the earth, m
9 R_E = 1.49e+011; // Mean-earth sun distance, m
10 S = 4*pi*r^2; // Surface area of the sun, Sq.m
11 T_sun = b/lambda_max; // The temperature of the
    sun's surface, K
12 R_T = sigma*T_sun^4; // Power per unit area
    radiated by the sun, W/Sq.m
13 P_sun = R_T*S; // The total power radiated from
    the sun's surface, W
14 F = r_E^2/(4*R_E^2); // Fraction of sun's
    radiation received by Earth

```

```

15 P_Earth_received = P_sun*F;    // The radiation
    received by the Earth from the sun, W
16 U_Earth = P_Earth_received*60*60*24;    // The
    radiation received by the Earth from the sun in
    one day, J
17 R_Earth = P_Earth_received/(%pi*r_E^2);    // Power
    received by the Earth per unit of exposed area, W
    /Sq.m
18 printf("\nThe surface temperature of the sun = %4d K
    ", ceil(T_sun));
19 printf("\nThe power per unit area emitted from the
    surface of the sun = %4.2e W/Sq.m", R_T);
20 printf("\nThe energy received by the Earth each day
    from the radiation of sun = %4.2e J", U_Earth);
21 printf("\nThe power received by the Earth per unit
    of exposed area = %4d W/Sq.m", ceil(R_Earth));
22
23 // Result
24 // The surface temperature of the sun = 5796 K
25 // The power per unit area emitted from the surface
    of the sun = 6.40e+007 W/Sq.m
26 // The energy received by the Earth each day from
    the radiation of sun = 1.54e+022 J
27 // The power received by the Earth per unit of
    exposed area = 1397 W/Sq.m
28 // The answers are given wrong in the textbook

```

Scilab code Exa 3.10 Exposure time of light to produce a photoelectron of given ki

```

1 // Scilab Code Ex3.10: Page-106 (2013)
2 clc; clear
3 phi = 2.36;    // Work function of sodium, eV
4 N_A = 6.02e+023;    // Avogadro's number
5 e = 1.6e-019;    // Energy equivalent of 1 eV, J
6 I = 1e-008;    // Intensity of incident radiation, W

```

```

/Sq.m
7 K = 1.00; // Kinetic energy of the ejected
photoelectron , eV
8 rho = 0.97; // Density of Na atoms, g/cc
9 M = 23; // Gram atomic mass of Na, g/mol
10 n = N_A*1e+006/M*rho; // Number of Na atoms per
unit volume, atoms/metre-cube
11 // Assume a cubic structure, then 1/d^3 = n, solving
for d
12 d = (1/n)^(1/3); // Thickness of one layer of
sodium atoms, m
13 N = n*d; // Number of exposed atoms per Sq.m
14 R = I/(N*e); // Rate of energy received by each
atom, eV/s
15 t = (phi+K)/R; // Time needed to absorb 3.36 eV
energy
16 printf("\nThe exposure time of light to produce the
photoelectron of %4.2f kinetic energy = %4.1f
years", K, t/(60*60*24*365.25));
17
18 // Result
19 // The exposure time of light to produce the
photoelectron of 1.00 kinetic energy = 14.7 years

```

Scilab code Exa 3.11 Photoelectric effect for lithium

```

1 // Scilab Code Ex3.11: Page-109 (2013)
2 clc; clear
3 phi = 2.93; // Work function of lithium, eV
4 lambda = 400e-009; // Wavelength of incident
light, m
5 e = 1.6e-019; // Energy equivalent of 1 eV, J
6 c = 2.998e+008; // Speed of light in vacuum, m/s
7 h = 6.626e-034; // Planck's constant, Js
8 E = h*c/(lambda*e); // Energy of incident light,

```

```

    eV
9  V0 = E - phi;    // Stopping potential, V
10 printf("\nThe energy of incident photons = %4.2f eV"
    , E);
11 printf("\nThe stopping potential = %4.2f V", V0);
12
13 // Result
14 // The energy of incident photons = 3.10 eV
15 // The stopping potential = 0.17 V

```

Scilab code Exa 3.12 Lithium exposed to light radiation

```

1
2 // Scilab Code Ex3.12: Page-109 (2013)
3 clc; clear
4 phi = 2.93;    // Work function of lithium, eV
5 c = 2.998e+008; // Speed of light in vacuum, m/s
6 K = 3.00;    // Kinetic energy of photoelectron, eV
7 E = phi + K; // Total energy of the incident
    light, eV
8 h = 6.626e-034; // Planck's constant, Js
9 e = 1.6e-019; // Energy equivalent of 1 eV, J
10 f = E*e/h; // Frequency of incident light, Hz
11 lambda = c/f; // Wavelength of the incident light
    , m
12 printf("\nThe frequency of incident light = %4.2e Hz
    ", f);
13 printf("\nThe wavelength of the incident light = %4
    .2f nm", lambda/1e-009);
14
15 // Result
16 // The frequency of incident light = 1.43e+015 Hz
17 // The wavelength of the incident light = 209.37 nm

```

Scilab code Exa 3.13 Number of photons in the light beam of given wavelength and i

```
1 // Scilab Code Ex3.13: Page-110 (2013)
2 clc; clear
3 lambda = 350; // Wavelength of incident light, nm
4 e = 1.6e-019; // Energy equivalent of 1 eV, J
5 E = 1.250e+003/lambda; // Total energy of the
   incident light, eV
6 I = 1e-008; // Intensity of incident light, W/Sq.
   m
7 // As Intensity, I = N*e, solving for N
8 N = I/(E*e); // The number of photons in the
   light beam
9 printf("\nThe number of photons in the light beam =
   %3.1e photons/Sq.m/s", N);
10
11 // Result
12 // The number of photons in the light beam = 1.8e
   +010 photons/Sq.m/s
```

Scilab code Exa 3.15 Minimum wavelength of the X rays

```
1 // Scilab Code Ex3.15: Page-113 (2013)
2 clc; clear
3 e = 1.6e-019; // Energy equivalent of 1 eV, J
4 c = 2.998e+008; // Speed of light in vacuum, m/s
5 h = 6.626e-034; // Planck's constant, Js
6 V0 = 35e+003; // Electron acceleration voltage, V
7 lambda_min = h*c/(e*V0); // Duane-Hunt rule for
   wavelength, m
8 printf("\nThe minimum wavelength of X-rays = %4.2e m
   ", lambda_min);
```

```

9
10 // Result
11 // The minimum wavelength of X-rays = 3.55e-011 m

```

Scilab code Exa 3.16 X ray scattering from the gold target

```

1 // Scilab Code Ex3.16: Page-116 (2013)
2 clc; clear
3 c = 2.998e+008; // Speed of light in vacuum, m/s
4 h = 6.626e-034; // Planck's constant, Js
5 m_e = 9.11e-031; // Rest mass of an electron, kg
6 lambda = 0.050; // Wavelength of the X-ray, nm
7 theta = 180; // The angle at which the recoil
// electron Ke becomes the largest, degree
8 E_x_ray = 1.240e+003/lambda; // Energy of the X-
// ray, eV
9 lambda_prime = lambda + (1-cosd(theta))*h/(m_e*c*1e
// -009); // The largest wavelength of the
// scattered photon, nm
10 E_prime_x_ray = 1.240e+003/lambda_prime; //
// Energy of the scattered photon, eV
11 K = (E_x_ray - E_prime_x_ray)/1e+003; // Kinetic
// energy of the most energetic recoil electron, keV
12 if (E_prime_x_ray < E_x_ray) then
13     printf("\nThe X-ray is Compton-scattered by the
// electron.");
14 else
15     printf("\nThe X-ray is not Compton-scattered by
// the electron.");
16 end
17 printf("\nThe largest wavelength of the scattered
// photon = %5.3f nm", lambda_prime);
18 printf("\nThe kinetic energy of the most energetic
// recoil electron = %3.1f keV", K);
19 printf("\nThe angle at which the recoil electron

```

```
energy is the largest = %d degrees", theta);
20
21 // Result
22 // The X-ray is Compton-scattered by the electron.
23 // The largest wavelength of the scattered photon =
    0.055 nm
24 // The kinetic energy of the most energetic recoil
    electron = 2.2 keV
25 // The angle at which the recoil electron energy is
    the largest = 180 degrees
```

Chapter 4

Structure of the Atom

Scilab code Exa 4.1 Maximum scattering angle in Geiger and Marsden experiment

```
1 // Scilab Code Ex4.1: Page-129 (2013)
2 clc; clear
3 m_e = 0.000549; // Rest mass of an electron , u
4 m_He = 4.002603; // Rest mass of a helium , u
5 M_alpha = m_He - 2*m_e; // Mass of alpha particle
   , u
6 theta_max = 2*m_e/M_alpha; // Maximum scattering
   angle for alpha particle , rad
7 printf("\\nThe maximum scattering angle for alpha
   particle = %5.3f degrees", theta_max*180/%pi);
8
9 // Result
10 // The maximum scattering angle for alpha particle =
   0.016 degrees
```

Scilab code Exa 4.2 Fraction of alpha particles deflected from a gold foil

```
1 // Scilab Code Ex4.2: Page-137 (2013)
```

```

2  clc; clear
3  rho = 19.3;      // Density of gold , g/cc
4  N_A = 6.02e+023; // Avogadro's number
5  N_M = 1;        // Number of atoms per molecule
6  M_g = 197;      // Gram atomic mass of gold , g/mol
7  n = rho*N_A*N_M/(M_g*1e-006); // Number density
    of gold atoms , atoms/metre-cube
8  Z1 = 79;        // Atomic number of gold
9  Z2 = 2;         // Atomic number of He nucleus
10 t = 1e-006;     // Thickness of the gold foil , m
11 e = 1.602e-019; // Charge on an electron , C
12 k = 9e+009;     // Coulomb constant , N-Sq.m/C^2
13 theta = 90;     // Angle of deflection of alpha
    particle , degrees
14 K = 7.7;        // Kinetic energy of alpha particles ,
    MeV
15 f = %pi*n*t*(Z1*Z2*e^2*k/(2*1.6e-013*K))^2*cotd(
    theta/2)^2; // The fraction of alpha particles
    deflected
16 printf("\\nThe fraction of alpha particles deflected
    = %1.0e" , f);
17
18 // Result
19 // The fraction of alpha particles deflected = 4e
    -005

```

Scilab code Exa 4.3 Fraction of alpha particles deflected from goil foil at a give

```

1 // Scilab Code Ex4.3: Page-138 (2013)
2 clc; clear
3 rho = 19.3;      // Density of gold , g/cc
4 N_A = 6.02e+023; // Avogadro's number
5 N_M = 1;        // Number of atoms per molecule
6 M_g = 197;      // Gram atomic mass of gold , g/mol
7 n = rho*N_A*N_M/(M_g*1e-006); // Number density

```

```

        of gold atoms, atoms/metre-cube
8  Z1 = 79;      // Atomic number of gold
9  Z2 = 2;      // Atomic number of He nucleus
10 t = 2.1e-007; // Thickness of the gold foil, m
11 e = 1.602e-019; // Charge on an electron, C
12 k = 9e+009;   // Coulomb constant, N-Sq.m/C^2
13 r = 1e-002;   // Distance of the alpha particles
        from the target, m
14 theta = 45;   // Angle of deflection of alpha
        particle, degrees
15 K = 7.7;      // Kinetic energy of alpha particles,
        MeV
16 f = n*t*(Z1*Z2*e^2*k)^2/((r*1.6e-013*K)^2*sind(theta
        /2)^4*16); // The fraction of alpha particles
        deflected
17 printf("\nThe fraction of alpha particles deflected
        at %d degrees = %3.1e per mm square", theta, f/1e
        +006);
18
19 // Result
20 // The fraction of alpha particles deflected at 45
        degrees = 3.2e-007 per mm square

```

Scilab code Exa 4.5 Size of the nucleus

```

1 // Scilab Code Ex4.5:Page-139 (2013)
2 clc; clear
3 Z1 = 2;      // Atomic number of He nucleus
4 Z2 = 13;     // Atomic number of aluminium
5 e = 1.602e-019; // Charge on an electron, C
6 k = 9e+009;   // Coulomb constant, N-Sq.m/C^2
7 K = 7.7;     // Kinetic energy of alpha particles,
        MeV
8 r_min = Z1*Z2*e^2*k/(K*1.6e-013); // Size of the
        aluminium nucleus, m

```

```

9 printf("\nThe size of the aluminium nucleus = %3.1e
    m", r_min);
10
11 // Result
12 // The size of the aluminium nucleus = 4.9e-015 m
13
14
15 // Result
16 // The maximum scattering angle for alpha particle =
    0.016 degrees

```

Scilab code Exa 4.6 Nonrelativistic justification for the speed of the electron

```

1 // Scilab Code Ex4.6: Page-140 (2013)
2 clc; clear
3 c = 3.00e+008; // Speed of light, m/s
4 r = 0.5e-010; // Radius of the atom, m
5 e = 1.6e-019; // Charge on an electron, C
6 m_e = 9.11e-031; // Mass of the electron, kg
7 k = 9e+009; // Coulomb constant, N-Sq.m/C^2
8 v = e*k^(1/2)/(m_e*r)^(1/2); // Speed of the
    electron, m/s
9 if v < 0.01*c then
10     printf("\nThe nonrelativistic treatment for
        calculating speed of the electron = %3.1e m/s
        is justified", v);
11 end
12
13 // Result
14 // The nonrelativistic treatment for calculating
    speed of the electron = 2.2e+006 m/s is justified

```

Scilab code Exa 4.7 Longest and shortest wavelengths observed in Paschen series for

```

1 // Scilab Code Ex4.7: Page-146(2013)
2 clc; clear
3 function region = check_region(lambda)
4     if lambda >= 400 & lambda < 700 then
5         region = "visible";
6         else region = "infrared";
7     end
8 endfunction
9 n_l = 3; // Lower electron orbit in Paschen
    series
10 n_u = [4, %inf]; // First and limiting upper
    orbits in Paschen series
11 R_inf = 1.0974e+007; // Rydberg constant, per
    metre
12 lambda_max = 1/(R_inf*(1/n_l^2-1/n_u(1)^2)*1e-009);
    // The longest wavelength in Paschen series,
    nm
13 region = check_region(lambda_max); // Check for
    the region
14 printf("\nThe maximum wavelength is %d nm and is in
    the %s region", ceil(lambda_max), region);
15 lambda_min = 1/(R_inf*(1/n_l^2-1/n_u(2)^2)*1e-009);
    // The shortest wavelength in Paschen series,
    nm
16 region = check_region(lambda_min); // Check for
    the region
17 printf("\nThe minimum wavelength is %d nm and is
    also in the %s region", lambda_min, region);
18
19 // Result
20 // The maximum wavelength is 1875 nm and is in the
    infrared region
21 // The minimum wavelength is 820 nm and is also in
    the infrared region

```

Scilab code Exa 4.8 Wavelengths of H alpha lines for three isotopes of hydrogen

```
1 // Scilab Code Ex4.8: Page-149(2013)
2 clc; clear
3 m_e = 0.0005486; // Mass of an electron , u
4 m_p = 1.007276; // Mass of a proton , u
5 m_d = 2.013553; // Mass of a deuteron , u
6 m_t = 3.015500; // Mass of a triton , u
7 R_inf = 1.0974e+007; // Rydberg constant , per
  metre
8 R_H = 1/(1+m_e/m_p)*R_inf; // Rydberg constant
  for hydrogen
9 R_D = 1/(1+m_e/m_d)*R_inf; // Rydberg constant
  for deuterium
10 R_T = 1/(1+m_e/m_t)*R_inf; // Rydberg constant
  for tritium
11 lambda_H = 1/(R_H*(1/2^2-1/3^2)*1e-009); //
  Wavelength of H_alpha line for hydrogen , nm
12 lambda_D = 1/(R_D*(1/2^2-1/3^2)*1e-009); //
  Wavelength of H_alpha line for deuterium , nm
13 lambda_T = 1/(R_T*(1/2^2-1/3^2)*1e-009); //
  Wavelength of H_alpha line for tritium , nm
14 printf("\nThe wavelength of H_alpha line for
  hydrogen = %6.2f nm", lambda_H);
15 printf("\nThe wavelength of H_alpha line for
  deuterium = %6.2f nm", lambda_D);
16 printf("\nThe wavelength of H_alpha line for tritium
  = %6.2f nm", lambda_T);
17
18 // Result
19 // The wavelength of H_alpha line for hydrogen =
  656.45 nm
20 // The wavelength of H_alpha line for deuterium =
  656.27 nm
21 // The wavelength of H_alpha line for tritium =
  656.22 nm
```

Scilab code Exa 4.9 Shortest wavelength emitted by doubly positive Li ion

```
1 // Scilab Code Ex4.9: Page-150 (2013)
2 clc; clear
3 R = 1.0974e+007; // Rydberg constant, per metre
4 Z = 3; // Atomic number of Li
5 n_l = 1; // Lower orbit of Li++ ion
6 n_u = %inf; // Limiting orbit of Li++ ion
7 lambda = 1/(Z^2*R*(1/n_l^2-1/n_u^2)*1e-009); //
   The shortest wavelength emitted by Li++ ion, nm
8 printf("\\nThe shortest wavelength emitted by Li++
   ion = %4.1f nm", lambda);
9
10 // Result
11 // The shortest wavelength emitted by Li++ ion =
   10.1 nm
```

Chapter 5

Wave Properties of Matter and Quantum Mechanics I

Scilab code Exa 5.1 The wavelength of the X rays incident on rock salt

```
1 // Scilab Code Ex5.1 :Page-167 (2013)
2 clc; clear;
3 N_A = 6.022e+23; // Avogadro's No., per mole
4 n = 1; // Order of diffraction
5 M = 58.5; // Molecular mass of NaCl, g/mol
6 rho = 2.16; // Density of rock salt, g/cc
7 two_theta = 20; // Scattering angle, degree
8 theta = two_theta/2; // Diffraction angle, degree
9 N = N_A*rho*2/(M*1e-006); // Number of atoms per
    unit volume, per metre cube
10 d = (1/N)^(1/3); // Interplanar spacing of NaCl
    crystal, m
11 lambda = 2*d*sind(theta)/n; // Wavelength of X-
    rays using Bragg's law, m
12 printf("\nThe wavelength of the incident X rays =
    %5.3f nm", lambda/1e-009);
13
14 // Result
15 // The wavelength of the incident X rays = 0.098 nm
```

Scilab code Exa 5.2 de Broglie wavelength of a tennis ball and an electron

```
1 // Scilab Code Ex5.2 : Page-168 (2013)
2 clc; clear;
3 h = 6.63e-034; // Planck's constant, Js
4 c = 3e+008; // Speed of light, m/s
5 // For a moving ball
6 m = 0.057; // Mass of the ball, kg
7 v = 25; // Velocity of ball, m/s
8 p = m*v; // Momentum of the ball, kgm/s
9 lambda = h/p; // Lambda is the wavelength of ball
, nm
10 printf("\\nThe wavelength of ball = %3.1e m", lambda)
;
11 // For a moving electron
12 m = 0.511e+006; // Rest mass of an electron, eV
13 K = 50; // Kinetic energy of the electron, eV
14 p = sqrt(2*m*K); // Momentum of the electron, kgm
/s
15 lambda = h*c/(1.602e-019*p*1e-009); // Wavelength
of the electron, nm
16 printf("\\nThe wavelength of the electron = %4.2f nm"
, lambda);
17
18 // Result
19 // The wavelength of ball = 4.7e-034 m
20 // The wavelength of the electron = 0.17 nm
```

Scilab code Exa 5.3 de Broglie wavelength of an electron used by Davisson and Germ

```
1 // Scilab Code Ex5.3 : Page-173 (2013)
```

```

2  clc; clear;
3  m = 9.1e-31;    // Mass of the electron , kg
4  h = 6.63e-34;  // Planck's constant , Js
5  c = 3e+008;    // Speed of light , m/s
6  e = 1.6e-19;   // Energy equivalent of 1 eV, J/eV
7  V0 = 54;       // Potential difference between
                   electrodes , V
8  lambda = h*c/(sqrt(2*m*c^2/e*V0)*e*1e-009);    // de
                   Broglie wavelength of the electron , nm
9  printf("\nThe de Broglie wavelength of the electron
           used by Davisson and Germer = %5.3f nm", lambda);
10
11 // Result
12 // The de Broglie wavelength of the electron used by
           Davisson and Germer = 0.167 nm

```

Scilab code Exa 5.4 Wavelength of a neutron at different temperatures

```

1  // Scilab Code Ex5.4 : Page-174 (2013)
2  clc; clear;
3  h = 6.63e-34;  // Planck's constant , Js
4  c = 3e+008;    // Speed of light , m/s
5  e = 1.6e-19;   // Energy equivalent of 1 eV, J/eV
6  m = 1.67e-27;  // Mass of a neutron , kg
7  k = 1.38e-23;  // Boltzmann constant , J/mol/K
8  T = [300 77];  // Temperatures , K
9  lambda = h*c/(sqrt(3*m*c^2/e*k/e*T(1))*e);    // The
                   wavelength of the neutron at 300 K, nm
10 printf("\nThe wavelength of the neutron at %d K = %5
           .3f nm", T(1), lambda/1e-09);
11 lambda = h*c/(sqrt(3*m*c^2/e*k/e*T(2))*e);    // The
                   wavelength of the neutron at 77 K, nm
12 printf("\nThe wavelength of the neutron at %d K = %5
           .3f nm", T(2), lambda/1e-09);
13

```

```

14 // Result
15 // The wavelength of the neutron at 300 K = 0.146 nm
16 // The wavelength of the neutron at 77 K = 0.287 nm

```

Scilab code Exa 5.7 Distance between first two maxima in fringe pattern

```

1 // Scilab Code Ex5.7 :Page-184 (2013)
2 clc; clear;
3 h = 6.626e-34; // Planck's constant, Js
4 c = 3e+008; // Speed of light, m/s
5 e = 1.602e-019; // Energy equivalent of 1 eV, J/
   ev
6 d = 2000; // Distance between slit centres, nm
7 K = 50e+003; // Kinetic energy of electrons, eV
8 l = 350e+006; // Distance of screen from the
   slits, nm
9 lambda = 1.226/sqrt(K); // Non-relativistic value
   of de Broglie wavelength of the electrons, nm
10 E0 = 0.511e+006; // Rest energy of the electron,
   J
11 E = K + E0; // Total energy of the electron, J
12 p_c = sqrt(E^2 - E0^2); // Relativistic mass
   energy relation, eV
13 lambda_r = h*c/(p_c*e*1e-009); // Relativistic
   value of de Broglie wavelength, nm
14 percent_d = (lambda - lambda_r)/lambda*100; //
   Percentage decrease in relativistic value
   relative to non-relativistic value
15 sin_theta = lambda_r/d; // Bragg's law
16 y = l*sin_theta; // The distance of first maximum
   from the screen, nm
17 printf("\\nThe percentage decrease in relativistic
   value relative to non-relativistic value = %1.0f
   percent", percent_d);
18 printf("\\nThe distance between first two maxima = %3

```

```

    .0 f nm", y);
19
20 // Result
21 // The percentage decrease in relativistic value
    relative to non-relativistic value = 2 percent
22 // The distance between first two maxima = 938 nm

```

Scilab code Exa 5.8 Momentum uncertainty fo a tennis ball and an electron

```

1 // Scilab Code Ex5.8 : Page-187 (2013)
2 clc; clear;
3 dx = 17.5; // The uncertainty in position , m
4 h = 1.05e-034; // Reduced Planck 's constant , Js
5 dp_x = h/(2*dx); // The uncertainty in momentum,
    kgm/s
6 printf("\\nThe unecrtainty in momentum of the ball =
    %1.0e kg-m/s", dp_x);
7 dx = 0.529e-010; // The uncertainty in position ,
    m
8 dp_x = h/(2*dx); // The uncertainty in momentum,
    kgm/s
9 printf("\\nThe uncertainty in momentum of the
    electron = %1.0e kg-m/s", dp_x);
10
11 // Result
12 // The unecrtainty in momentum of the ball = 3e-036
    kg-m/s
13 // The uncertainty in momentum of the electron = 1e
    -024 kg-m/s

```

Scilab code Exa 5.9 Minimum kinetic energy of an electron in hydrogen atom

```

1 // Scilab Code Ex5.9 : Page-188 (2013)

```

```

2 clc; clear;
3 a_0 = 5.29e-11;    // Radius of H-atom, m
4 l = 2*a_0;       // Length, m
5 h = 6.63e-34;    // Planck's constant, Js
6 m = 9.1e-31;     // Mass of electron, kg
7 K_min = h^2/(8*(%pi)^2*m*l^2);    // Minimum kinetic
    energy possessed, J
8 printf("\\nThe minimum kinetic energy of the electron
    = %3.1f eV", K_min/1.6e-19);
9
10 // Result
11 // The minimum kinetic energy of the electron = 3.4
    eV

```

Scilab code Exa 5.10 Minimum kinetic energy of electron localized within a typical

```

1 // Scilab Code 5.10: : Page-190 (2013)
2 clc; clear;
3 dx = 6e-015;    // The uncertainty in position of
    the electron, m
4 h_bar = 1.054e-034;    // PReduced Planck's constant
    , Js
5 e = 1.602e-019;    // Energy equivalent of 1 eV, J/
    eV
6 c = 3e+008;    // Speed of light, m/s
7 E0 = 0.511e+006;    // Rest mass energy of the
    electron, J
8 dp = h_bar*c/(2*dx*e);    // Minimum electron
    momentum, eV/c
9 p = dp;    // Momentum of the electron at least
    equal to the uncertainty in momentum, eV/c
10 E = sqrt(p^2+E0^2)/1e+006;    // Relativistic energy
    of the electron, MeV
11 K = E - E0/1e+006;    // Minimum kinetic energy of
    the electron, MeV

```

```

12 printf("\nThe minimum kinetic energy of the electron
    = %4.1f MeV", K);
13
14 // Result
15 // The minimum kinetic energy of the electron =
    15.9 MeV

```

Scilab code Exa 5.12 Energy width of excited state of atom and uncertainty ratio o

```

1 // Scilab Code Ex5.12 : Page-190 (2014)
2 clc; clear;
3 c = 3e+8; // Speed of light, m/s
4 dt = 1e-08; // Relaxation time of atom, s
5 h = 6.6e-34; // Planck's constant, Js
6 dE = h/(4*pi*dt); // Energy width of excited
    state of atom, J
7 lambda = 300e-009; // Wavelegh of emitted photon
    , m
8 f = c/lambda; // Frequency of emitted photon, per
    sec
9 printf("\nThe energy width of excited state of the
    atom = %3.1e eV", dE/1.6e-019);
10 df = dE/h; // Uncertainty in frequency, per sec
11 printf("\nThe uncertainty ratio of the frequency =
    %1.0e", df/f);
12
13 // Result
14 // The energy width of excited state of the atom =
    3.3e-008 eV
15 // The uncertainty ratio of the frequency = 8e-009

```

Scilab code Exa 5.13 Energy at different levels

```

1 // Scilab Code Ex5.13 : Page-195 (2014)
2 clc; clear;
3 n = [1 2 3]; // First three energy levels
4 e = 1.6e-019; // Energy equivalent of 1 eV, J/eV
5 c = 3e008; // Speed of light, m/s
6 h = 6.63e-034; // Planck's constant, Js
7 m = 9.1e-031; // Mass of the proton, kg
8 l = 0.1; // Length of one-dimensional box, nm
9 E_n = n^2*(h*c/(e*1e-009))^2/(8*m*c^2/e*l^2); //
    Energy of nth level, eV
10 printf("\n\nThe first three energy level are:\nE1 = %2
    .0f eV, E2 = %3.0f eV and E3 = %3.0f eV", E_n(1),
    E_n(2), E_n(3));
11
12 // Result
13 // The first three energy level are:
14 // E1 = 38 eV, E2 = 151 eV and E3 = 340 eV

```

Chapter 6

Quantum Mechanics II

Scilab code Exa 6.4 Probabilities of a particle in the given regions

```
1 // Scilab Code Ex6.4 : Page-205 (2014)
2 clc; clear;
3 alpha = 1; // For simplicity assume alpha to be
  unity
4 P = integrate('sqrt(alpha)*exp(-2*alpha*x)', 'x', 0,
  1/alpha); // Probability that the particle
  lies between 0 and 1/alpha
5 printf("\nThe probability that the particle lies
  between 0 and 1/alpha = %5.3f", P);
6 P = integrate('sqrt(alpha)*exp(-2*alpha*x)', 'x', 1/
  alpha, 2/alpha); // Probability that the
  particle lies between 1/alpha and 2/alpha
7 printf("\nThe probability that the particle lies
  between 1/alpha and 2/alpha = %5.3f", P);
8
9 // Result
10 // The probability that the particle lies between 0
  and 1/alpha = 0.432
11 // The probability that the particle lies between 1/
  alpha and 2/alpha = 0.059
```

Scilab code Exa 6.9 Transition energy for a proton confined to a nucleus

```
1 // Scilab Code Ex6.9: Page-215 (2014)
2 clc; clear;
3 h = 6.62e-034; // Planck's constant, Js
4 h_bar = h/(2*%pi); // Reduced Planck's constant,
   Js
5 c = 3.00e+008; // Speed of light, m/s
6 e = 1.602e-019; // Energy equivalent of 1 eV, J
7 m = 938.3e+006; // Energy equivalent of proton
   mass, eV
8 L = 1e-005; // Diameter of the nucleus, nm
9 E1 = %pi^2*(h_bar*c/(e*1e-009))^2/(2*L^2*m*1e+006);
   // Energy of the ground state of proton, MeV
10 E2 = 4*E1; // Energy of first excited state of
   proton, MeV
11 delta_E = E2 -E1; // Transition energy of the
   proton inside the nucleus, MeV
12 printf("\nThe transition energy of the proton inside
   the nucleus = %1d MeV", delta_E);
13
14 // Result
15 // The transition energy of the proton inside the
   nucleus = 6 MeV
```

Scilab code Exa 6.14 Fraction of electrons tunneling through a barrier

```
1 // Scilab Code Ex6.14: Page-229 (2014)
2 clc; clear;
3 h = 6.62e-034; // Planck's constant, Js
4 h_bar = h/(2*%pi); // Reduced Planck's constant,
   Js
```

```

5 c = 3.00e+008;    // Speed of light , m/s
6 e = 1.602e-019;  // Energy equivalent of 1 eV, J
7 m = 0.511e+006;  // Energy equivalent of electron
    rest mass, eV
8 V0 = 10;         // Height of potential barrier , eV
9 E = 5;          // Energy of the incident electrons , eV
10 L = 0.8e-009;   // Width of the potential barrier ,
    m
11 k = sqrt(2*m*(V0 - E))*e/(h_bar*c);    //
    Schrodinger 's constant , per m
12 T = (1 + V0^2*sinh(k*L)^2/(4*E*(V0 - E)))^(-1);
    // Transmission electron probability
13 printf("\nThe fraction of electrons tunneled through
    the barrier = %3.1e", T);
14
15 // Result
16 // The fraction of electrons tunneled through the
    barrier = 4.4e-008

```

Scilab code Exa 6.15 Probability of electron tunneling through the barrier

```

1 // Scilab Code Ex6.15: Page-229 (2014)
2 clc; clear;
3 h = 6.62e-034;    // Planck 's constant , Js
4 h_bar = h/(2*%pi); // Reduced Planck 's constant ,
    Js
5 c = 3.00e+008;    // Speed of light , m/s
6 e = 1.602e-019;  // Energy equivalent of 1 eV, J
7 m = 0.511e+006;  // Energy equivalent of electron
    rest mass, eV
8 V0 = 10;         // Height of potential barrier , eV
9 Sum_M = 0;
10 i = 1;
11 for E = 5:-1:1    // Range of energies of the
    incident electrons , eV

```

```

12     M = 16*E/V0*(1-E/V0);    // All the factors
        multiplying the exponential term
13     Sum_M = Sum_M + M;      // Accumulator
14     i = i + 1;
15 end
16 E = 5;    // Given energy of the incident electrons ,
        eV
17 M = int(Sum_M/i);    // AVERAGE value of M
18 L = 0.8e-009;    // Width of the potential barrier ,
        m
19 k = sqrt(2*m*(V0 - E))*e/(h_bar*c);    //
        Schrodinger 's constant , per m
20 T = M*exp(-2*k*L);    // Transmission electron
        probability
21 printf("\nThe fraction of electrons tunneled through
        the barrier = %3.1e", T);
22
23 // Result
24 // The fraction of electrons tunneled through the
        barrier = 2.2e-008

```

Scilab code Exa 6.16 A particle penetrating through a potential step

```

1 // Scilab Code Ex6.16: Page-230 (2014)
2 clc; clear;
3 h = 6.62e-034;    // Planck 's constant , Js
4 h_bar = h/(2*%pi);    // Reduced Planck 's constant ,
        Js
5 c = 3.00e+008;    // Speed of light , m/s
6 e = 1.602e-019;    // Energy equivalent of 1 eV, J
7 m = 0.511e+006;    // Energy equivalent of electron
        rest mass , eV
8 V0 = 10;    // Height of potential barrier , eV
9 E = 5;    // Given energy of the incident electrons ,
        eV

```

```

10 l = h_bar*c/(2*sqrt(2*m*(V0 - E))*1e-009*e);    //
    Penetration distance into the barrier when the
    probability of the particle penetration drops to
    1/e, nm
11 printf("\nThe penetration distance for a %d eV
    electron approaching a step barrier of %d eV = %5
    .3f nm", E, V0, l);
12
13 // Result
14 // The penetration distance for a 5 eV electron
    approaching a step barrier of 10 eV = 0.044 nm

```

Scilab code Exa 6.17 An alpha particle tunnelling through a nucleus

```

1
2 // Scilab Code Ex6.17: Page-234 (2014)
3 clc; clear;
4 h = 6.62e-034;    // Planck's constant, Js
5 h_bar = h/(2*pi);    // Reduced Planck's constant,
    Js
6 c = 3.00e+008;    // Speed of light, m/s
7 e = 1.602e-019;    // Charge on an electron, C
8 k = 9e+009;    // Coulomb constant, N-Sq.m./C^2
9 m = 3727;    // Energy equivalent of alpha particle
    rest mass, MeV
10 E = 5;    // Given energy of the incident electrons,
    eV
11 Z1 = 2;    // Atomic number of an alpha particle
12 Z2 = 92;    // Atomic number of the U-238 nucleus
13 r_N = 7e-015;    // Nuclear radius, m
14 K = 4.2;    // Kinetic energy of alpha particle, MeV
15 V_C = Z1*Z2*e^2*k/(r_N*e*1e+006);    // Coulomb
    Potential, MeV
16 r_prime = V_C*r_N/(K*1e-015);    // Distance through
    which the alpha particle must tunnel, fm

```

```

17 kapa = sqrt(2*m*(V_C-E))*e/(h_bar*c*1e-006);    //
    Schronginger's Constant, per m
18 L = r_prime - r_N/1e-015;    // Barrier width, fm
19 T = 16*(K/V_C)*(1-K/V_C)*exp(-2*kapa*L*1e-015);
    // Tunnelling probability of alpha particle
20 V_C_new = V_C/2;    // Potential equal to half the
    Coulomb potential, MeV
21 L = L/2;    // Width equal to half the barrier width
    , fm
22 kapa = sqrt(2*m*(V_C_new-E))*e/(h_bar*c*1e-006);
    // Schronginger's Constant, per m
23 T_a = 16*(K/V_C_new)*(1-K/V_C_new)*exp(-2*kapa*L*1e
    -015);    // Approximated tunnelling probability
    of alpha particle
24 v = sqrt(2*K/m)*c;    // Speed of the alpha particle
    , m/s
25 t = r_N/v;    // Time taken by alpha particle to
    cross the U-238 nucleus, s
26 printf("\nThe barrier height = %2d MeV", ceil(V_C));
27 printf("\nThe distance that alpha particle must
    tunnel = %2d fm", r_prime);
28 printf("\nThe tunneling potential assuming square-
    top potential = %1.0e", T);
29 printf("\nThe approximated tunneling potential = %3
    .1e", T_a);
30 printf("\nThe speed of the alpha particle = %3.1e m/
    s", v);
31 printf("\nThe time taken by alpha particle to cross
    the U-238 nucleus = %1.0e s", t);
32
33 // Result
34 // The barrier height = 38 MeV
35 // The distance that alpha particle must tunnel = 63
    fm
36 // The answers are given wrongly in the textbook
37 // The tunneling potential assuming square-top
    potential = 6e-123
38 // The approximated tunneling potential = 3.8e-040

```

- 39 // The speed of the alpha particle = 1.4×10^7 m/s
40 // The time taken by alpha particle to cross the U
-238 nucleus = 5×10^{-22} s
41 // Some answers are given wrong in the textbook for
this problem
-

Chapter 7

The Hydrogen Atom

Scilab code Exa 7.2 Normalization of hydrogen wave function

```
1 // Scilab Code Ex7.2: Page-248 (2014)
2 clc; clear;
3 a0 = 1; // For simplicity assume Bohr radius to
   be unity, m
4 NE = 1/(64*%pi*a0^5)*integrate('r^4*exp(-r/a0)', 'r',
   , 0, 15)*integrate('sin(t)^3', 't', 0, %pi)*
   integrate('p^0', 'p', 0, 2*%pi);
5 if round(NE) == 1 then
6     printf("\nThe hydrogen wave function <211| is
   normalized");
7 else
8     printf("\nThe hydrogen wave function <211| is
   not normalized");
9 end
10
11 // Result
12 // The hydrogen wave function <211| is normalized
```

Scilab code Exa 7.4 Degeneracy of M level in hydrogen atom

```

1 // Scilab Code Ex7.4: Page-252 (2014)
2 clc; clear;
3 n = 3; // Principal quantum number
4 Total = 0;
5 printf("\nn      l      m_l      2(l + 1)");
6 printf("\n-----");
7 for l = 0:1:n-1
8     printf("\n%d", n);
9     printf("      %d      ", l);
10    if l > 0 then
11        count = 0;
12        for m_l = -l:1:l
13            printf("%2d ", m_l);
14            count = count + 1;
15        end
16        if l == 1 then
17            printf("      ");
18        else
19            printf("");
20        end
21    else
22        m_l = 0;
23        count = 0;
24        printf("%2d      ", m_l);
25        count = count + 1;
26    end
27    printf("      %d", count);
28    Total = Total + count;
29 end
30 printf("\n      Total = %d", Total);
31
32 // Result
33 // n      l      m_l      2(l + 1)
34 // -----
35 // 3      0      0      1
36 // 3      1      -1  0  1      3
37 // 3      2      -2 -1  0  1  2      5
38 //                               Total = 9

```

Scilab code Exa 7.7 Energy difference between components of p states of atomic hydrogen

```
1 // Scilab Code Ex7.7: Page-255 (2014)
2 clc; clear;
3 e = 1.602e-019; // Charge on an electron, C
4 h = 6.62e-034; // Planck's constant, Js
5 h_bar = h/(2*%pi); // Reduced Planck's constant,
   Js
6 m = 9.11e-031; // Electron mass, kg
7 B = 2.00; // External magnetic field, T
8 m_l1 = 0; // Lower orbital magnetic quantum number
9 m_l2 = 1; // Upper orbital magnetic quantum number
10 delta_m_l = m_l2 - m_l1; // Change in m_l
11 mu_B = e*h_bar/(2*m); // Bohr's magneton, J/T
12 delta_E = mu_B*B*delta_m_l/e; // Energy
   difference between components of p states of
   atomic hydrogen placed in the external field, eV
13 printf("\n\nThe value of Bohr magneton = %4.2e J/T",
   mu_B);
14 printf("\n\nThe energy difference between components
   of p states of atomic hydrogen placed in the
   external field = %4.2e eV", delta_E);
15
16 // Result
17 // The value of Bohr magneton = 9.26e-024 J/T
18 // The energy difference between components of p
   states of atomic hydrogen placed in the external
   field = 1.16e-004 eV
```

Scilab code Exa 7.8 Separation of the atomic beam

```

1 // Scilab Code Ex7.8: Page-257 (2014)
2 clc; clear;
3 m = 1.67e-027; // Mass of the proton, kg
4 k = 1.38e-023; // Boltzmann constant, J/K
5 T = 663; // Temperature of the discharge tube, K
6 v_x = sqrt(3*k*T/m); // Average speed of the
    hydrogen atom
7 mu_z = 9.27e-024; // Bohr's magneton, J/T
8 B_grad = 1240; // Magnetic field gradient, T/m
9 delta_x = 0.03; // Length of the homogeneous
    magnetic field, m
10 d = 1/(2*m)*(mu_z*B_grad)*(delta_x/v_x)^2; //
    Separation of the atomic beam, m
11 printf("\nThe separation of the atomic beam = %4.2 f
    mm", d/1e-003);
12
13 // Result
14 // The separation of the atomic beam = 0.19 mm

```

Scilab code Exa 7.9 Number of distinct states for the 4d level of atomic hydrogen

```

1 // Scilab Code Ex7.9: Page-259 (2014)
2 clc; clear;
3 n = 4; // Principal quantum number
4 l = 2; // For 4d-state
5 printf("\nn      l      m_l      m_s");
6 printf("\n-----");
7     count = 0;
8     for m_l = -1:1:1
9         if (m_l == 0) then
10             printf("\n%d", n);
11             printf("      %d      ", l);
12             printf("      %2d", m_l);
13             printf("      +1/2, -1/2");
14             else

```

```

15         printf("\n                %2d", m_l);
16         printf("        +1/2, -1/2");
17     end
18     count = count + 2;
19 end
20 printf("\nTotal No. of different states for 4d level
        of atomic hydrogen = %d", count);
21
22 // Result
23 // n      l          m_l      m_s
24 // -----
25 //                -2      +1/2, -1/2
26 //                -1      +1/2, -1/2
27 // 4      2          0      +1/2, -1/2
28 //                1      +1/2, -1/2
29 //                2      +1/2, -1/2
30 // Total No. of different states for 4d level of
        atomic hydrogen = 10

```

Scilab code Exa 7.10 Energy of allowed transitions for the hydrogen atom

```

1 // Scilab Code Ex7.10: Page-263 (2014)
2 clc; clear;
3 function flag = check_allowance(dn, dl, dml, dms)
4     if (dl == -1 | dl == 1 | dml == -1 | dml == 0 |
        dml == 1 | dms == -1 | dms == 0 | dms == 1) &
        dl <> 0 then
5         flag = 1;
6     else
7         flag = 0;
8     end
9 endfunction
10 state = [2 0 0 1/2; 3 1 1 1/2; 2 0 0 1/2; 3 0 0 1/2;
        4 2 -1 -1/2; 2 1 0 1/2];
11 for i = 1:2:5

```

```

12     flag = 0;
13     d_n = state(i,1) - state(i+1,1);
14     d_l = state(i,2) - state(i+1,2);
15     d_m_l = state(i,3) - state(i+1,3);
16     d_m_s = state(i,4) - state(i+1,4);
17     flag = check_allowance(d_n, d_l, d_m_l, d_m_s);
18     if flag == 1 then
19         printf("\n\nThe transition (%d,%d,%d,1/%d)
                --> (%d,%d,%d,1/%d) is allowed", state(i
                ,1), state(i,2), state(i,3), state(i,4)
                *4, state(i+1,1), state(i+1,2), state(i
                +1,3), state(i+1,4)*4);
20         delta_E = -13.6*(1/state(i+1)^2-1/state(i)
                ^2);
21         printf("\nThe energy of this transition is
                %4.2f eV", delta_E);
22     else
23         printf("\n\nThe transition (%d,%d,%d, %d)-->
                (%d,%d,%d,%d) is not allowed", state(i
                ,1), state(i,2), state(i,3), state(i,4),
                state(i+1,1), state(i+1,2), state(i+1,3),
                state(i+1,4));
24     end
25 end
26
27 // Result
28 // The transition (2,0,0,1/2) --> (3,1,1,1/2) is
    allowed
29 // The energy of this transition is 1.89 eV
30
31 // The transition (2,0,0, 0)--> (3,0,0,0) is not
    allowed
32
33 // The transition (4,2,-1,1/-2) --> (2,1,0,1/2) is
    allowed
34 // The energy of this transition is -2.55 eV

```

Scilab code Exa 7.13 Probability of the electron in the 1s state of the hydrogen a

```
1 // Scilab Code Ex7.13: Page-265 (2014)
2 clc; clear;
3 a0 = 1; // For simplicity assume bohr radius to
   be unity
4 P = integrate('4/a0^3*exp(-2*r/a0)*r^2', 'r', a0,
   10);
5 printf("\nThe probability of the electron in the 1s
   state of the hydrogen atom = %4.2f", P);
6
7 // Result
8 // The probability of the electron in the 1s state
   of the hydrogen atom = 0.68
```

Chapter 8

Atomic Physics

Scilab code Exa 8.3 Splitting of 3p subshell of sodium

```
1 // Scilab Code Ex8.3: Page-285 (2014)
2 clc; clear;
3 delta_E = 2e-003; // Energy difference for the 3p
  subshell of sodium, eV
4 h = 6.62e-034; // Planck's constant, Js
5 e = 1.602e-019; // Energy equivalent of 1 eV, J
6 c = 3.00e+008; // Speed of light in vacuum, m/s
7 lambda = 589.3; // Wavelength of spectral line,
  nm
8 // As  $\Delta E = hc/\lambda^2 \Delta \lambda$ , solving
  for  $\Delta \lambda$ 
9 delta_lambda = lambda^2*e/(h*c*1e+009)*delta_E;
  // Splitting of 3p subshell of sodium, nm
10 printf("\nThe splitting of 3p subshell of sodium =
  %3.1f nm", delta_lambda);
11
12 // Result
13 // The splitting of 3p subshell of sodium = 0.6 nm
```

Scilab code Exa 8.5 LS coupling of two electrons in an atom

```
1 // Scilab Code Ex8.5: Page-289 (2014)
2 clc; clear;
3 l1 = 1; // Orbital angular momentum quantum
    number for first electron
4 l2 = 2; // Orbital angular momentum quantum
    number for second electron
5 s1 = 1/2; // Spin angular momentum quantum number
    for first electron
6 s2 = 1/2; // Spin angular momentum quantum number
    for second electron
7 temp_j = zeros(15);
8 cnt = 1;
9 printf("\nThe all possible values of the total
    angular momentum quantum number of J are:\n");
10 for L = abs(l1 - l2):1:abs(l1 + l2)
11     for S = abs(s1 - s2):1:abs(s1 + s2)
12         for j = abs(L - S):1:abs(L + S)
13             temp_j(cnt) = j;
14             cnt = cnt + 1;
15         end
16     end
17 end
18 J = -1;
19 temp_J = gsort(temp_j, 'g', 'i');
20 for i = 1:1:cnt-1
21     if temp_J(i) > J then
22         J = temp_J(i);
23         printf("%d ", J);
24     end
25 end
26
27 // Result
28 // The all possible values of the total angular
    momentum quantum number of J are:
29 // 0 1 2 3 4
```

Scilab code Exa 8.8 Internal magnetic field causing spin orbit splitting

```
1 // Scilab Code Ex8.8: Page-291 (2014)
2 clc; clear;
3 delta_E = 0.0021; // Energy difference for the 3p
    subshell of sodium, eV
4 h = 6.62e-034; // Planck's constant, Js
5 h_bar = h/(2*%pi); // Reduced Planck's constant,
    Js
6 e = 1.602e-019; // Energy equivalent of 1 eV, J
7 m = 9.11e-031; // Rest of an an electron, kg
8 g_s = 2; // Gyromagnetic ratio due to spin
    splitting
9 // As delta_E = g_s*e*h_bar/(2*m)*B, solving for B
10 B = m*delta_E/h_bar; // Internal magnetic field
    causing the LS splitting, T
11 printf("\\nThe internal magnetic field causing the LS
    splitting = %2d T", B);
12
13 // Result
14 // The internal magnetic field causing the LS
    splitting = 18 T
```

Chapter 9

Statistical Physics

Scilab code Exa 9.1 Mean translational kinetic energy of gas molecules at room tem

```
1 // Scilab Code Ex9.1: Page-303 (2014)
2 clc; clear;
3 k = 1.38e-023; // Boltzmann constant, J/K
4 N_A = 6.023e+023; // Avogadro's number
5 T = 293; // Room temperature, K
6 e = 1.6e-019; // Energy equivalent of 1 eV, J
7 // For a single molecule
8 K_bar_single = 3/2*k*T/e; // Mean translational
   kinetic energy of a single gas molecule, J
9 // For a 1 mole of molecules
10 K_bar_mole = K_bar_single*N_A*e; // Mean
   translational kinetic energy of 1 mole of gas
   molecules, J
11 printf("\nThe mean translational kinetic energy of
   the single idela gas molecule = %5.3f eV",
   K_bar_single);
12 printf("\nThe mean translational kinetic energy of
   the one mole of ideal gas molecules = %4d J",
   ceil(K_bar_mole));
13
14 // Result
```

```

15 // The mean translational kinetic energy of the
    single idela gas molecule = 0.038 eV
16 // The mean translational kinetic energy of the one
    mole of ideal gas molecules = 3654 J

```

Scilab code Exa 9.3 Mean molecular speed in light gas hydrogen and heavy radon gas

```

1 // Scilab Code Ex9.3: Page-310 (2014)
2 clc; clear;
3 k = 1.38e-023; // Boltzmann constant, J/K
4 u = 1.67e-027; // Mass equivalent of one atomic
    mass unit, kg
5 T = 293; // Room temperature, K
6 m_H = 1.008*u; // Gram atomic mass of hydrogen,
    kg
7 m = 2*m_H; // Gram molecular mass of hydrogen
    molecule, kg
8 v_bar = 4/sqrt(2*%pi)*sqrt(k*T/m); // Mean
    molecular speed in the light gas hydrogen, m/s
9 printf("\n\nThe mean molecular speed in the light gas
    hydrogen = %4d m/s", ceil(v_bar));
10 m = 222*u; // Gram atomic mass of Radon, kg
11 v_bar = 4/sqrt(2*%pi)*sqrt(k*T/m); // Mean
    molecular speed in the heavy radon gas, m/s
12 printf("\n\nThe mean molecular speed in the heavy
    radon gas = %3d m/s", ceil(v_bar));
13
14 // Result
15 // The mean molecular speed in the light gas
    hydrogen = 1749 m/s
16 // The mean molecular speed in the heavy radon gas =
    167 m/s

```

Scilab code Exa 9.4 Fraction of molecules in an ideal gas having speed near to the

```
1 // Scilab Code Ex9.4: Page-310 (2014)
2 clc; clear;
3 m = 1; // For simplicity assume mass of gas
        molecule to be unity, kg
4 k = 1.38e-023; // Boltzmann constant, J/K
5 T = 293; // Room temperature, K
6 bita = k*T; // Energy associated with three
        degrees of freedom, J
7 v_mps = sqrt(2/(bita*m)); // For simplicity assume
        most probable speed to be unity, m/s
8 C = (bita*m/(2*pi))^(3/2); // Constant in the
        distribution function
9 P = integrate('4*pi*C*exp(-1/2*bita*m*v^2)*v^2', 'v
        ', 0.99*v_mps, 1.01*v_mps);
10 printf("\nThe fraction of molecules in an ideal gas
        in equilibrium which have speeds within 1 percent
        above and below the most probable speed = %5.3f"
        , P);
11
12 // Result
13 // The fraction of molecules in an ideal gas in
        equilibrium which have speeds within 1 percent
        above and below the most probable speed = 0.017
```

Scilab code Exa 9.6 Relative number of atoms in the ground and first excited state

```
1 // Scilab Code Ex9.6: Page-315 (2014)
2 clc; clear;
3 k = 1.38e-023; // Boltzmann constant, J/K
4 T = [293 5000 1e+006]; // Room temperature,
        temperature at the surface of the star and
        temperature at the star interior respectively, K
5 e = 1.6e-019; // Energy equivalent of 1 eV, J
```

```

6 g_E1 = 2;    // Possible configuration of the
  electrons in ground state of H-atom
7 g_E2 = 8;    // Possible configuration of the
  electrons in the first excited state of H-atom
8 E1 = -13.6;  // Energy of the ground state, eV
9 E2 = -3.4;  // Energy of the first excited state
  state, eV
10 n_ratio = zeros(3);
11 for i = 1:1:3
12     n_ratio(i) = g_E2/g_E1*exp(1/(k*T(i))*(E1 - E2)*
  e);
13     printf("\nFor T = %4.2e K, n_E2/n_E1 = %4.2e", T
  (i), n_ratio(i));
14 end
15
16
17 // Result
18 // For T = 2.93e+002 K, n_E2/n_E1 = 2.05e-175
19 // For T = 5.00e+003 K, n_E2/n_E1 = 2.14e-010
20 // For T = 1.00e+006 K, n_E2/n_E1 = 3.55e+000

```

Scilab code Exa 9.7 Fermi energy and Fermi temperature for copper

```

1 // Scilab Code Ex9.7: Page-320 (2014)
2 clc; clear;
3 e = 1.6e-019;    // Energy equivalent of 1 eV, J
4 n = 8.47e+028;   // Number density of conduction
  electrons in copper, per metre cube
5 k = 1.38e-023;   // Boltzmann constant, J/K
6 h = 6.626e-034; // Planck's constant, Js
7 m = 9.11e-031;   // Mass of an electron, kg
8 E_F = h^2/(8*m*e)*(3*n/%pi)^(2/3); // Fermi
  energy for copper, eV
9 T_F = E_F*e/k;   // Fermi temperature for copper, K
10 printf("\nThe Fermi energy for copper = %4.2f eV",

```

```

    E_F);
11 printf("\nThe Fermi temperature for copper = %4.2e K
    ", T_F);
12
13 // Result
14 // The Fermi energy for copper = 7.04 eV
15 // The Fermi temperature for copper = 8.16e+004 K

```

Scilab code Exa 9.8 Electronic contribution to the molar heat capacity of metals

```

1 // Scilab Code Ex9.8: Page-322 (2014)
2 clc; clear;
3 R = 1; // For simplicity assume the molar gas
    constant to be unity, J/mol/K
4 T = 293; // Room temperature, K
5 T_F = 8.16e+004; // The Fermi temperature for
    copper
6 C_V = %pi^2*T/(2*T_F)*R; // Electronic
    contribution to the molar heat capacity for
    copper, J/mol/K
7 printf("\nThe electronic contribution to the molar
    heat capacity for copper = %6.4fR", C_V);
8 T_F = 6.38e+004; // The Fermi temperature for
    silver
9 C_V = %pi^2*T/(2*T_F)*R; // Electronic
    contribution to the molar heat capacity for
    silver, J/mol/K
10 printf("\nThe electronic contribution to the molar
    heat capacity for silver = %6.4fR", C_V);
11
12 // Result
13 // The electronic contribution to the molar heat
    capacity for copper = 0.0177R
14 // The electronic contribution to the molar heat
    capacity for silver = 0.0227R

```


Chapter 10

Molecules Lasers and Solids

Scilab code Exa 10.1 Energy of lowest rotational state of nitrogen gas

```
1 // Scilab Code Ex10.1 : Page-342 (2014)
2 clc; clear;
3 m = 2.33e-026; // Mass of a nitrogen atom, kg
4 R = 1.1e-010; // Interatomic separation between
    two nitrogen atoms, m
5 h = 6.626e-034; // Planck's constant, Js
6 e = 1.6e-019; // Energy equivalent of 1 eV, J
7 h_bar = h/(2*%pi); // Reduced Planck's constant,
    Js
8 I = m*R^2/2; // Moment of rotational inertia of
    nitrogen gas molecule, kg-Sq.m
9 l = 1; // Rotational angular momentum quantum
    number
10 E_rot = h_bar^2*l*(l+1)/(2*I); // The energy for
    lowest rotational state of the nitrogen molecule,
    J
11 printf("\\nThe energy for lowest rotational state of
    the nitrogen molecule = %4.2e eV", E_rot/e);
12
13 // Result
14 // The energy for lowest rotational state of the
```

nitrogen molecule = 4.93×10^{-4} eV

Scilab code Exa 10.2 Vibrational energy levels of HCl molecule

```
1
2 // Scilab Code Ex10.2 : Page-343 (2014)
3 clc; clear;
4 h = 6.626e-034; // Planck's constant, Js
5 e = 1.6e-019; // Energy equivalent of 1 eV, J
6 h_bar = h/(2*pi); // Reduced Planck's constant,
   Js
7 k = 1.38e-023; // Boltzmann constant, J/K
8 u = 1.67e-027; // Mass equivalent of 1 amu, kg
9 m1 = 34.97*u; // Atomic mass of chlorine atom, kg
10 m2 = 1.008*u; // Atomic mass of hydrogen atom, kg
11 mu = m1*m2/(m1 + m2); // Reduced mass of the HCl
   system, kg
12 delta_E = 0.36; // Spacing between vibrational
   energy levels of the HCl molecule, eV
13 omega = delta_E*e/h_bar; // Angular frequency of
   vibration, rad/s
14 kapa = mu*omega^2; // Effective force constant
   for HCl molecule, N/m
15 T = delta_E*e/k; // Classical temperature
   associated with the rotational energy spacing, K
16
17 printf("\nThe effective force constant for HCl
   molecule = %3d N/m", ceil(kapa));
18 printf("\nThe classical temperature associated with
   the rotational energy spacing = %4d K", ceil(T));
19
20 // Result
21 // The effective force constant for HCl molecule =
   489 N/m
22 // The classical temperature associated with the
```


23 rotational energy spacing = 4174 K
// The answers are given wrong in the textbook

Scilab code Exa 10.4 Range parameter for NaCl

```
1 // Scilab Code Ex10.4 : Page-358 (2014)
2 clc; clear;
3 e = 1.602e-019; // Charge on an electron , C
4 N_A = 6.023e+023; // Avogadro's number
5 alpha = 1.7476; // Madelung constant
6 E = -764.4e+003; // Dissociation energy of NaCl
   molecule , J/mol
7 V = E/N_A; // Repulsive potential energy , J
8 k = 8.988e+009; // Coulomb's constant , N-Sq.m/C^2
9 r0 = 0.282e-009; // Equilibrium separation for
   nearest neighbour in NaCl, m
10 rho = r0*(1+r0*V/(k*alpha*e^2)); // Range
   parameter for NaCl, nm
11 printf("\nThe range parameter for NaCl = %6.4 f nm",
   rho/1e-009);
12
13 // Result
14 // The range parameter for NaCl = 0.0316 nm
```

Scilab code Exa 10.5 Induced diamagnetism in an atom

```
1 // Scilab Code Ex10.5 :Page-364 (2014)
2 clc; clear;
3 e = 1.602e-019; // Charge on an electron , C
4 r = 5.29e-011; // Orbital radius of electron
   equal to Bohr radius , m
5 B = 2.0; // Applied magnetic field , T
6 m = 9.11e-031; // Mass of an electron , kg
```

```

7 delta_mu = e^2*r^2*B/(4*m);    // Induced
    diamagnetic moment in the atom, J/T
8 printf("\nThe induced diamagnetic moment in the atom
    = %3.1e J/T", delta_mu);
9
10 // Result
11 // The induced diamagnetic moment in the atom = 3.9e
    -029 J/T

```

Scilab code Exa 10.6 Paramagnetism in a typical material

```

1 // Scilab Code Ex10.6 : Page-366 (2014)
2 clc; clear;
3 mu_B = 9.27e-024;    // Bohr's magneton, J/T
4 B = 0.50;    // Applied magnetic field, T
5 k = 1.38e-023;    // Boltzmann constant, J/K
6 T = 10*mu_B*B/k;    // Temperature at which mu*B =
    0.1k*T, K
7 b_muB = mu_B*B/(k*T);
8 ratio = b_muB/tanh(b_muB);    // Ratio of b_muB and
    tanh(b_muB)
9 if (ratio - 1) < 0.01 then
10     printf("\nThe value of T = %4.2f K is suitable as
        a classical temperature.", T);
11 else
12     printf("\nThe value of T = %4.2f K is not
        suitable as a classical temperature.", T);
13 end
14 // For higher temperature
15 T = 100;    // Given temperature
16 b_muB = mu_B*B/(k*T);
17 ratio = b_muB/tanh(b_muB);    // Ratio of b_muB and
    tanh(b_muB)
18 if (ratio - 1) < 0.001 then
19     printf("\nAt the value of T = %4.2f K, the

```

```

        approximation is an excellent one.", T);
20 else
21     printf("\nAt the value of T = %4.2f K, the
        approximation is not an excellent.", T);
22 end
23
24 // Result
25 // The value of T = 3.36 K is suitable as a classical
    temperature.
26 // At the value of T = 100.00 K, the approximation
    is an excellent one.

```

Scilab code Exa 10.7 Superconductivity in niobium

```

1 // Scilab Code Ex10.7 : Page-374 (2014)
2 clc; clear;
3 k = 1.38e-023; // Boltzmann constant, J/K
4 e = 1.602e-019; // Energy equivalent of 1 eV, J
5 h = 6.62e-034; // Planck's constant, Js
6 c = 3.00e+008; // Speed of light, m/s
7 T_c = 9.25; // Critical temperature for niobium,
    K
8 E_g = 3.54*k*T_c; // Energy gap for niobium from
    BCS theory, J
9 lambda = h*c/E_g; // Minimum photon wavelength
    needed to break the Cooper pair, m
10 printf("\nThe energy gap for niobium = %4.2f meV",
    E_g/(e*1e-003));
11 printf("\nThe minimum photon wavelength needed to
    break the Cooper pair = %4.2e m", lambda);
12
13 // Result
14 // The energy gap for niobium from BCS theory = 2.82
    meV
15 // The minimum photon wavelength needed to break the

```

Cooper pair = $4.39e-004$ m

Scilab code Exa 10.8 Magnetic field perpendicular to the loop

```
1 // Scilab Code Ex10.8 : Page-382 (2014)
2 clc; clear;
3 r = 1e-002; // Radius of the loop, m
4 phi0 = 2.068e-015; // Magnetic flux penetrating
   to the loop, T-Sq.m
5 A = %pi*r^2; // Area of the loop, Sq.m
6 B = phi0/A; // Magnetic field perpendicular to
   the loop, T
7 printf("\\nThe magnetic field perpendicular to the
   loop = %4.2e T", B);
8
9 // Result
10 // The magnetic field perpendicular to the loop =
   6.58e-012 T
```

Chapter 11

Semiconductor Theory and Devices

Scilab code Exa 11.1 Relative number of electrons with given energies above the va

```
1 // Scilab Code Ex11.1 : Page-399 (2014)
2 clc; clear;
3 e = 1.6e-019; // Energy equivalent of 1 eV, J
4 k = 1.38e-023; // Boltzmann constant, J/K
5 T = 293; // Room temperature, K
6 dE = [0.10, 1.0, 10.0]; // Energies above the
    valence band, eV
7 F_FD = zeros(3);
8 for i = 1:1:3
9     F_FD(i) = 1/(exp(dE(i)*e/(k*T)) + 1);
10    printf("\nFor E - E_F = %4.2f eV, F_FD = %4.2e",
    dE(i), F_FD(i));
11 end
12
13 // Result
14 // For E - E_F = 0.10 eV, F_FD = 1.88e-002
15 // For E - E_F = 1.00 eV, F_FD = 6.53e-018
16 // For E - E_F = 10.00 eV, F_FD = 1.40e-172
```

Scilab code Exa 11.3 Hall effect in zinc strip

```
1
2 // Scilab Code Ex11.3: Page-402 (2014)
3 clc; clear;
4 e = 1.6e-019; // Energy equivalent of 1 eV, J
5 rho = 5.92e-008; // Resistivity of the zinc at
    room temperature, ohm-m
6 B = 0.25; // Magnetic field applied perpendicular
    to the strip, T
7 x = 10.0e-002; // Length of the zinc strip, m
8 y = 2.0e-002; // Width of the zinc strip, m
9 V = 20e-003; // Potential difference applied
    across the strip, V
10 I = 0.400; // Current through the strip, A
11 V_H = 0.56e-006; // Hall voltage that appeared
    across the strip, V
12 z = rho*x*I/(y*V); // Thickness of the strip, m
13 n = I*B/(e*V_H*z); // Number density of the
    charge carriers, per metre cube
14 printf("\\nThe thickness of the zinc strip = %4.2e m"
    , z);
15 printf("\\nThe number density of the charge carriers
    = %4.2e per metre cube", n);
16 printf("\\nThe charge carries in zinc are positive.")
    ;
17
18 // Result
19 // The thickness of the zinc strip = 5.92e-006 m
20 // The number density of the charge carriers = 1.89e
    +029 per metre cube
21 // The charge carries in zinc are positive.
```

Scilab code Exa 11.4 Current through a reverse bias pn junction diode

```
1 // Scilab Code Ex11.4: Page-408 (2014)
2 clc; clear;
3 e = 1.602e-019; // Energy equivalent of 1 eV, J
4 k = 1.38e-023; // Boltzmann constant, J/K
5 T = 293; // Room temperature, K
6 V_f = 0.200; // Forward voltage, V
7 I_f = 50e-003; // Forward current, A
8 V_r = -0.200; // Reverse voltage, V
9 I_r = I_f*(exp(e*V_r/(k*T))-1)/(exp(e*V_f/(k*T)) -
10 1); // Reverse current from diode equation, A
11 printf("\nThe reverse current through pn-junction
12 diode = %2d micro-ampere", I_r/1e-006);
13 // Result
14 // The reverse current through pn-junction diode =
15 -18 micro-ampere
```

Scilab code Exa 11.5 Energy produced by a solar cell per day

```
1 // Scilab Code Ex11.5: Page-412 (2014)
2 clc; clear;
3 A = 100*100; // Area of solar cell, Sq.m
4 t = 12*60*60; // Time for which the solar cell
5 operates, s
6 phi = 680; // Solar flux received by the solar
7 cell, W/Sq.m
8 eta = 0.30 // Efficiency of the solar array
9 E_array = eta*phi*A*t; // Energy produced by
10 solar cell in one 12-hour day, J
```

```

8 printf("\nThe energy produced by solar cell in one
   12-hour day : %3.1e J", E_array);
9 P = 100e+006;    // Power output of power plant, W
10 t = 24*60*60;   // Time for which power plant
   operates, s
11 E_plant = P*t;   // Energy produced by power plant,
   J
12 printf("\nThe energy produced by power plant in one
   day : %3.1e J which is about %d times more than
   that produced by solar cell array..!", E_plant,
   ceil(E_plant/E_array));
13
14 // Result
15 // The energy produced by solar cell in one 12-hour
   day : 8.8e+010 J
16 // The energy produced by power plant in one day :
   8.6e+012 J which is about 99 times more than that
   produced by solar cell array..!

```

Scilab code Exa 11.6 Data bits stored into CD ROM

```

1
2 // Scilab Code Ex11.6: Page-418 (2014)
3 clc; clear;
4 r1 = 2.30e-002;    // Radius of inner edge of
   storing region of CD-ROM, m
5 r2 = 5.80e-002;   // Radius of outer edge of storing
   region of CD-ROM, m
6 A = %pi*(r2^2 - r1^2); // Area of the usable
   region of CD-ROM, Sq.m
7 N = 700e+006*8;   // Total number of bits in CD-ROM
8 APB = A/N;        // Area per bit of CD-ROM, Sq.m/bit
9 t = 1.6e-006;     // Track width of CD-ROM, m
10 l = APB/t;       // Bit length, m
11 printf("\nThe surface area of CD-ROM allowed for

```



```
    each data bit = %3.1e Sq.m/bit", APB);
12 printf("\nThe approx. dimensions of each bit along
    the track = %1.0f micro-metre", 1/1e-006);
13
14 // Result
15 // The surface area of CD-ROM allowed for each data
    bit = 1.6e-012 Sq.m/bit
16 // The approx. dimensions of each bit along the
    track = 1 micro-metre
```

Chapter 12

The Atomic Nucleus

Scilab code Exa 12.1 Minimum kinetic energy of a proton in a medium sized nucleus

```
1 // Scilab Code Ex12.1: Page-432 (2014)
2 clc; clear;
3 h = 6.62e-034; // Planck's constant, Js
4 h_bar = h/(2*%pi); // Reduced Planck's constant,
   Js
5 m = 1.67e-027; // Rest mass of proton, kg
6 e = 1.602e-019; // Energy equivalent of 1 eV, J
7 c = 3.00e+008; // Speed of light, m/s
8 delta_x = 8.0e-015; // Size of the nucleus, m
9 delta_p = h_bar/(2*delta_x*e); // Uncertainty in
   momentum of proton from Heisenberg Uncertainty
   Principle, eV-s/m
10 p_min = delta_p; // Minimum momentum of the
   proton inside the nucleus, eV-s/m
11 K = (p_min*c)^2*e/(2*m*c^2*1e+006); // The
   minimum kinetic energy of the proton in a medium
   sized nucleus, MeV
12 printf("\\nThe minimum kinetic energy of the proton
   in a medium sized nucleus = %4.2f MeV", K);
13
14 // Result
```

```
15 // The minimum kinetic energy of the proton in a
    medium sized nucleus = 0.08 MeV
```

Scilab code Exa 12.2 Nuclear radius of calcium

```
1 // Scilab Code Ex12.2: Page-436 (2014)
2 clc; clear;
3 c = 3.00e+008; // Speed of light, m/s
4 e = 1.602e-019; // Energy equivalent of 1 eV, J
5 m_e = 0.511; // Rest mass energy of electron, MeV
6 m_p = 938.3; // Rest mass energy of proton, MeV
7 h = 6.62e-034; // Planck's constant, Js
8 A = 40; // Mass number of Ca-40
9 r0 = 1.2; // Nuclear radius constant, fm
10 R = r0*A^(1/3); // Radius of Ca-40 nucleus, fm
11 printf("\nThe radius of Ca-40 nucleus = %3.1f fm", R
    );
12 lambda = 2.0; // de-Broglie wavelength to
    distinguish a distance at least half the radius,
    fm
13 // Electron energy
14 E = ceil(sqrt(m_e^2+(h*c/(lambda*e*1e+006*1e-015))
    ^2)); // Total energy of the probing electron,
    MeV
15 K = E - m_e; // Kinetic energy of probing
    electron, MeV
16 printf("\nThe kinetic energy of probing electron =
    %3d MeV", ceil(K));
17 // Proton energy
18 E = ceil(sqrt(m_p^2+(h*c/(lambda*e*1e+006*1e-015))
    ^2)); // Total energy of the probing electron,
    MeV
19 K = E - m_p; // Kinetic energy of probing
    electron, MeV
20 printf("\nThe kinetic energy of probing proton = %3d
```

```

    MeV", ceil(K));
21
22 // Result
23 //The radius of Ca-40 nucleus = 4.1 fm
24 // The kinetic energy of probing electron = 620 MeV
25 // The kinetic energy of probing proton = 187 MeV
26 // The answers are deviated due to the
    approximations used in the textbook

```

Scilab code Exa 12.3 Radii of U238 and He4 nuclei

```

1 // Scilab Code Ex12.3: Page-437 (2014)
2 clc; clear;
3 A_U238 = 238; // Mass number of U-238
4 A_He4 = 4; // Mass number of He-4
5 r0 = 1.2; // Nuclear radius constant, nm
6 R_U238 = r0*A_U238^(1/3); // Radius of U-238
    nucleus, fm
7 R_He4 = r0*A_He4^(1/3); // Radius of He-4 nucleus
    , fm
8 printf("\nThe radii of U-238 and He-4 nuclei are %3
    .1f fm and %3.1f fm repectively", R_U238, R_He4);
9 ratio = R_U238/R_He4; // Ratio of the two radii
10 printf("\nThe ratio of radius to U-238 to that of He
    -4 = %3.1f", ratio);
11
12 // Result
13 // The radii of U-238 and He-4 nuclei are 7.4 fm and
    1.9 fm repectively
14 // The ratio of radius to U-238 to that of He-4 =
    3.9

```

Scilab code Exa 12.4 A proton subjected to the magnetic field

```

1 // Scilab Code Ex12.4: Page-438 (2014)
2 clc; clear;
3 h = 6.62e-034; // Planck's constant, Js
4 c = 3.00e+008; // Speed of light, m/s
5 e = 1.602e-019; // Energy equivalent of 1 eV, J
6 B = 2.0; // Applied magnetic field, T
7 mu_N = 3.15e-008; // Nucleon magnetic moment, eV/
  T
8 mu_p = 2.79*mu_N; // Proton magnetic moment, eV/T
9 delta_E = 2*mu_p*B; // Energy difference between
  the up and down proton states, eV
10 f = delta_E*e/h; // Frequency of electromagnetic
  radiation that flips the proton spins, Hz
11 lambda = c/f; // Wavelength of electromagnetic
  radiation that flips the proton spins, m
12 printf("\\nThe energy difference between the up and
  down proton states = %3.1e eV", delta_E);
13 printf("\\nThe frequency of electromagnetic radiation
  that flips the proton spins = %2d MHz", f/1e
  +006);
14 printf("\\nThe wavelength of electromagnetic
  radiation that flips the proton spins = %3.1f m",
  lambda);
15
16 // Result
17 // The energy difference between the up and down
  proton states = 3.5e-007 eV
18 // The frequency of electromagnetic radiation that
  flips the proton spins = 85 MHz
19 // The wavelength of electromagnetic radiation that
  flips the proton spins = 3.5 m

```

Scilab code Exa 12.5 Binding energy of Be

```

1 // Scilab Code Ex12.5: Page-443 (2014)

```

```

2  clc; clear;
3  c = 3.00e+008;    // Speed of light , m/s
4  e = 1.602e-019;  // Energy equivalent of 1 eV, J
5  u = 931.5;       // Energy equivalent of 1 amu, MeV
6  m_n = 1.008665;  // Mass of a neutron, u
7  M_H = 1.007825;  // Mass of a proton, u
8  M_He = 4.002603; // Mass of helium nucleus
9  M_Be = 8.005305; // Mass of Be-8, u
10 B_Be = (4*m_n+4*M_H - M_Be)*u;
11 B_Be_2alpha = (2*M_He - M_Be)*u;
12 printf("\nThe binding energy of Be-8 = %4.1f MeV and
        is positive", B_Be);
13 if (B_Be_2alpha > 0) then
14     printf("\nThe Be-4 is stable w.r.t. decay into
        two alpha particles.");
15 else
16     printf("\nThe Be-4 is unstable w.r.t. decay into
        two alpha particles since the decay has
        binding energy of %5.3f MeV", B_Be_2alpha);
17 end
18
19 // Result
20 // The binding energy of Be-8 = 56.5 MeV and is
    positive
21 // The Be-4 is unstable w.r.t. decay into two alpha
    particles since the decay has binding energy of
    -0.092 MeV

```

Scilab code Exa 12.6 Total Coulomb energy of U238

```

1  // Scilab Code Ex12.6: Page-444 (2014)
2  clc; clear;
3  Z = 92;    // Atomic number of U-238
4  A = 238;   // Mass number of U-238
5  E_Coul = 0.72*Z*(Z-1)*A^(-1/3); // Total Coulomb

```

```

    energy of U-238, MeV
6 printf("\nThe total Coulomb energy of U-238 = %3d
    MeV", E_Coul);
7
8 // Result
9 // The total Coulomb energy of U-238 = 972 MeV

```

Scilab code Exa 12.8 Binding energy per nucleon

```

1 // Scilab Code Ex12.8: Page-447 (2014)
2 clc; clear;
3 u = 931.5; // Energy equivalent of 1 amu, MeV
4 m_p = 1.007825; // Mass of proton, amu
5 m_n = 1.008665; // Mass of neutron, amu
6 M_Ne = 19.992440; // Mass of Ne-20 nucleus, amu
7 M_Fe = 55.934942; // Mass of Fe-56 nucleus, amu
8 M_U = 238.050783; // Mass of U-238 nucleus, amu
9 A_Ne = 20; // Mass number of Ne-20 nucleus
10 A_Fe = 56; // Mass number of Fe-56 nucleus
11 A_U = 238; // Mass number of U-238 nucleus
12 BE_Ne = [10*m_p+10*m_n-M_Ne]*u; // Binding energy
    of Ne-20 nucleus, MeV
13 BE_Fe = [26*m_p+30*m_n-M_Fe]*u; // Binding energy
    of Fe-56 nucleus, MeV
14 BE_U = [92*m_p+146*m_n-M_U]*u; // Binding energy
    of U-238 nucleus, MeV
15 printf("\nThe binding energy per nucleon for Ne-20 :
    %4.2f MeV/nucleon", BE_Ne/A_Ne);
16 printf("\nThe binding energy per nucleon for Fe-56 :
    %4.2f MeV/nucleon", BE_Fe/A_Fe);
17 printf("\nThe binding energy per nucleon for U-238 :
    %4.2f MeV/nucleon", BE_U/A_U);
18
19 // Result
20 // The binding energy per nucleon for Ne-20 : 8.03

```

```

MeV/nucleon
21 // The binding energy per nucleon for Fe-56 : 8.79
MeV/nucleon
22 // The binding energy per nucleon for U-238 : 7.57
MeV/nucleon

```

Scilab code Exa 12.10 Radiative decay of Po210

```

1 // Scilab Code Ex12.10: Page-451 (2014)
2 clc; clear;
3 N_A = 6.023e+023; // Avogadro's number
4 T = 138*24*3600; // Half life of Po-210, s
5 R = 2000; // Activity of Po-210, disintegrations/
s
6 M = 0.210; // Gram molecular mass of Po-210, kg
7 f = 1/3.7e+010; // Conversion factor to convert
from decays/s to Ci
8 A = R*f/1e-006; // Activity of Po-210, micro-Ci
9 N = R*T/log(2); // Number of radioactive nuclei of
Po-210, nuclei
10 m = N*M/N_A; // Mass of Po-210 sample, kg
11 printf("\nThe activity of Po-210 = %5.3f micro-Ci",
A);
12 printf("\nThe mass of Po-210 sample = %3.1e kg", m);
13
14 // Result
15 // The activity of Po-210 = 0.054 micro-Ci
16 // The mass of Po-210 sample = 1.2e-014 kg

```

Scilab code Exa 12.11 Time of decay of F18 isotope

```

1 // Scilab Code Ex12.11: Page-452 (2014)
2 clc; clear;

```



```

3 T = 110;      // Half life of F-18, min
4 f_remain = 0.01;    // Fraction of the F-18 sample
    remained
5 t = -log(0.01)/(log(2)*60)*T;    // Time taken by
    the F-18 sample to decay to 1 percent of its
    initial value, h
6 printf("\nThe time taken for 99 percent of the F-18
    sample to decay = %4.1f h", t);
7
8 // Result
9 // The time taken for 99 percent of the F-18 sample
    to decay = 12.2 h

```

Scilab code Exa 12.12 Alpha activity of 10 kg sample of U235

```

1 // Scilab Code Ex12.12: Page-452 (2014)
2 clc; clear;
3 N_A = 6.023e+023;    // Avogadro's number
4 M = 10e+03;    // Mass of the U-235, g
5 M_U235 = 235;    // Molecular mass of U-235, g
6 t_half = 7.04e+008;    // Half life of U-235, y
7 N = M*N_A/M_U235;    // Number of U-235 atoms in 10
    kg sample
8 R = log(2)*N/t_half;    // The alpha activity of 10
    kg sample of U-235, decays/y
9 printf("\nThe alpha activity of 10 kg sample of U
    -235 = %3.1e Bq", R/(365.25*24*60*60));
10
11 // Result
12 // The alpha activity of 10 kg sample of U-235 = 8.0
    e+008 Bq

```

Scilab code Exa 12.13 Non emission of a neutron by U230

```

1 // Scilab Code Ex12.13: Page-453 (2014)
2 clc; clear;
3 u = 931.5; // Energy equivalent of 1 u, MeV
4 M_U230 = 230.033927; // Atomic mass of U-230, u
5 m_n = 1.008665; // Mass of a neutron, u
6 M_H = 1.007825; // Mass of hydrogen, u
7 M_U229 = 229.033496; // Gram atomic mass of U-230
8 Q = (M_U230 - M_U229 - m_n)*u; // Q-value of the
   reaction emitting a neutron
9 if (Q < 0) then
10     printf("\nThe neutron decay in this reaction is
   not possible.");
11 else
12     printf("\nThe neutron decay in this reaction is
   possible.");
13 end
14 Q = (M_U230 - M_U229 - M_H)*u; // Q-value of the
   reaction emitting a proton
15 if (Q < 0) then
16     printf("\nThe proton decay in this reaction is
   not possible.");
17 else
18     printf("\nThe proton decay in this reaction is
   possible.");
19 end
20
21 // Result
22 // The neutron decay in this reaction is not
   possible.
23 // The proton decay in this reaction is not possible
   .

```

Scilab code Exa 12.15 Possible reaction with Fe55 isotope

```

1 // Scilab Code Ex12.15: Page-461 (2014)

```

```

2  clc; clear;
3  u = 931.5;      // Energy equivalent of 1 u, MeV
4  M_Fe55 = 54.938298; // Atomic mass of Fe-55, u
5  M_Mn55 = 54.938050; // Atomic mass of Mn-55, u
6  m_e = 0.000549; // Mass of the electron, u
7  Q = (M_Fe55 - M_Mn55 - 2*m_e)*u; // Q-value of
   the reaction undergoing beta+ decay, MeV
8  if (Q < 0) then
9      printf("\nThe beta+ decay is not allowed for Fe
   -55");
10 else
11     printf("\nThe beta+ decay is allowed for Fe-55")
   ;
12 end
13 Q = (M_Fe55 - M_Mn55)*u; // Q-value of the
   reaction undergoing electron capture, MeV
14 if (Q < 0) then
15     printf("\nFe-55 may not undergo electron capture
   ");
16 else
17     printf("\nFe-55 may undergo electron capture");
18 end
19
20 // Result
21 // The beta+ decay is not allowed for Fe-55
22 // Fe-55 may undergo electron capture

```

Scilab code Exa 12.16 Allowed decay modes for Ac226

```

1  // Scilab Code Ex12.16: : Page-461 (2014)
2  clc; clear;
3  function [] = check_allowance(Q, decay_type)
4      if (Q < 0) then
5          printf("\nThe %s is not allowed for Ac-226",
   decay_type);

```

```

6 else
7     printf("\nThe %s is allowed for Ac-226",
            decay_type);
8 end
9 endfunction
10 u = 931.5; // Energy equivalent of 1 u, MeV
11 M_Ac226 = 226.026090; // Atomic mass of Ac-226, u
12 M_Fr222 = 222.017544; // Atomic mass of Fr-222, u
13 M_He4 = 4.002603; // Atomic mass of He-4, u
14 M_Th226 = 226.024891; // Atomic mass of M_Th226,
    u
15 M_Ra226 = 226.025403; // Atomic mass of M_Ra226,
    u
16 m_e = 0.000549; // Mass of the electron, u
17 // Alpha Decay
18 Q = (M_Ac226 - M_Fr222 - M_He4)*u; // Q-value of
    the reaction undergoing alpha decay, MeV
19 check_allowance(Q, "alpha decay");
20 // Beta- Decay
21 Q = (M_Ac226 - M_Th226)*u; // Q-value of the
    reaction undergoing beta- decay, MeV
22 check_allowance(Q, "beta- decay");
23 // Beta+ Decay
24 Q = (M_Ac226 - M_Ra226 - 2*m_e)*u; // Q-value of
    the reaction undergoing beta+ decay, MeV
25 check_allowance(Q, "beta+ decay");
26 // Electron Capture
27 Q = (M_Ac226 - M_Ra226)*u; // Q-value of the
    reaction undergoing electron capture, MeV
28 check_allowance(Q, "electron capture");
29
30 // Result
31 // The alpha decay is allowed for Ac-226
32 // The beta- decay is allowed for Ac-226
33 // The beta+ decay is not allowed for Ac-226
34 // The electron capture is allowed for Ac-226

```

Scilab code Exa 12.17 Error introduced in the gamma ray energy

```
1 // Scilab Code Ex12.17: Page-463(2014)
2 clc; clear;
3 u = 931.5; // Energy equivalent of 1 u, MeV
4 E_ex = 0.072; // Energy of the excited state, MeV
5 M = 226*u*1e; // Energy equivalent of atomic mass
    of Th-226, MeV
6 x = E_ex/(2*M); // The error introduced in the
    gamma ray energy by approximation
7 printf("\nThe error introduced in the gamma ray
    energy by approximation = %3.1e", x);
8
9 // Result
10 // The error introduced in the gamma ray energy by
    approximation = 1.7e-007
```

Scilab code Exa 12.18 Age of the uranium ore

```
1 // Scilab Code Ex12.18: Page-467(2014)
2 clc; clear;
3 t_half = 4.47e+009; // The half life of uranium
    ore, years
4 R_prime = 0.60; // The ratio of Pb206 abundance
    to that of U238
5 t = t_half/log(2)*log(R_prime + 1); // Age of the
    uranium ore, years
6 printf("\nThe age of U-238 ore = %3.1e years", t);
7
8 // Result
9 // The age of U-238 ore = 3.0e+009 years
```

Scilab code Exa 12.19 C14 dating to determine age of bone

```
1 // Scilab Code Ex12.19: Page-469(2014)
2 clc; clear;
3 t_half = 5730; // The half life of uranium ore ,
   years
4 R0 = 1.2e-012; // The initial ratio of C14 to C12
   at the time of death
5 R = 1.10e-012; // The final ratio of C14 to C12 t
   years after death
6 // As  $R = R0 \cdot \exp(-\lambda \cdot t)$ , solving for t
7 t = -log(R/R0)*t_half/log(2); // Age of the bone,
   years
8 printf("\\nThe %3d years age of bone does not date
   from the Roman Empire.", ceil(t));
9
10 // Result
11 // The 720 years age of bone does not date from the
   Roman Empire.
```

Chapter 13

Nuclear Interactions and Applications

Scilab code Exa 13.1 Number of neutrons produced in collision of alpha particle and

```
1 // Scilab Code Ex13.1: Page-479(2014)
2 clc; clear;
3 N_A = 6.02e+023; // Avogadro's number
4 e = 1.6e-019; // Charge on an electron, C
5 q = 2*e; // Charge on the alpha particle, C
6 rho = 1.9; // Density of carbon target, atoms/cc
7 N_M = 1; // Number of atoms per molecule
8 M_g = 12; // Gram atomic mass of C12 isotope, g/
   mol
9 sigma = 25e-031; // Total cross section for the
   reaction, Sq.m
10 t = 1e-006; // Thickness of carbon target, m
11 I_beam = 1e-006; // Beam current of alpha
   particle, ampere
12 time = 3600; // Time for which the alpha particle
   beam is incident on the target, s
13 n = rho*N_A*N_M/M_g; // Number of nuclei per unit
   volume, per cc
14 P = n*t*sigma*1e+006; // Probability of
```

```

    scattering of alpha particles
15 N_I = I_beam*time/q;    // Number of incident alpha
    particles
16 N_n = N_I*P;    // Number of neutrons produced in
    the reaction
17 printf("\nThe number of neutrons produced in the
    reaction = %3.1e neutrons", N_n);
18
19 // Result
20 // The number of neutrons produced in the reaction =
    2.7e+009 neutrons

```

Scilab code Exa 13.2 Likelihood of a neutron production than a proton

```

1 // Scilab Code Ex13.2: Page-480(2014)
2 clc; clear;
3 sigma_n = 3;    // Differential cross setion of
    production of the neutron, mb/sr
4 sigma_p = 0.2;    // Differential cross setion of
    production of the proton, mb/sr
5 // As P_n = sigma_n and P_p = sigma_p, so
6 P_ratio = sigma_n/sigma_p;    // The likelihood of a
    neutron production than a proton
7 printf("\nThe likelihood of the neutron production
    than the proton = %2d", P_ratio);
8
9 // Result
10 // The likelihood of the neutron production than the
    proton = 15

```

Scilab code Exa 13.3 Nuclear reaction observed by Rutherford

```

1 // Scilab Code Ex13.3: Page-481(2014)

```



```

2  clc; clear;
3  u = 931.5;      // Energy equivalent of 1 amu, MeV
4  M_He = 4.002603; // Mass of He-4 nucleus, u
5  M_N = 14.003074; // Mass of N-14 nucleus, u
6  M_H = 1.007825; // Mass of hydrogen nucleus, u
7  M_O = 16.999132; // Mass of O-16 nucleus, u
8  K_alpha = 7.7; // The kinetic energy of alpha
    particle, MeV
9  Q = (M_He + M_N - M_H - M_O)*u; // The ground
    state Q-value of the nuclear reaction, MeV
10 // As Q = K_p + K_O - K_alpha, solving for K_p + K_O
11 K = Q + K_alpha; // The sum of kinetic energy of
    the products, MeV
12 printf("\\nThe ground state Q-value of the endoergic
    nuclear reaction = %5.3f MeV", Q);
13 printf("\\nThe sum of kinetic energy of the products
    = %3.1f MeV", K);
14
15 // Result
16 // The ground state Q-value of the endoergic nuclear
    reaction = -1.192 MeV
17 // The sum of kinetic energy of the products = 6.5
    MeV

```

Scilab code Exa 13.4 Final energy of excitation of product nucleus in the nuclear

```

1 // Scilab Code Ex13.4: Page-484(2014)
2 clc; clear;
3 u = 931.5; // Energy equivalent of 1 amu, MeV
4 K_lab = 14.6; // Kinetic energy of incident alpha
    particle, MeV
5 Mx = 4; // Mass number of projectile nucleus
6 MX = 12; // Mass number of target nucleus
7 M_He = 4.002603; // Mass of He nucleus, u
8 M_C = 12.0 // Mass of carbon nucleus, u

```

```

9 M_0 = 15.994915;    // Mass of oxygen nucleus , u
10 K_cm = MX/(Mx + MX)*K_lab;    // Kinetic energy
    available in the centre of mass, MeV
11 E_0 = (M_He + M_C - M_0)*u;    // The ground state
    excitation energy of O-16, MeV
12 E_final_0 = K_cm + E_0;    // The final excitation
    energy of O-16, MeV
13 printf("\nThe final excitation energy of O-16 = %4.2
    f MeV", E_final_0);
14
15 // Result
16 // The final excitation energy of O-16 = 18.11 MeV

```

Scilab code Exa 13.5 Ground state Q value of the induced fission reaction

```

1
2 // Scilab Code Ex13.5: Page-487(2014)
3 clc; clear;
4 u = 931.5;    // Energy equivalent of 1 amu, MeV
5 M_U235 = 235.0439;    // Mass of U-235 nucleus , u
6 m_n = 1.0087;    // Mass of a neutron , u
7 M_Zr99 = 98.9165;    // Mass of Zr-99 nucleus , u
8 M_Te134 = 133.9115;    // Mass of Te-134 nucleus , u
9 Q = (M_U235 + m_n - M_Zr99 - M_Te134 - 3*m_n)*u;
    // Q-value of the reaction , MeV
10 printf("\nThe ground state Q-value of the induced
    fission reaction = %3d MeV", ceil(Q));
11
12 // Result
13 // The ground state Q-value of the induced fission
    reaction = 185 MeV

```

Scilab code Exa 13.6 Excitation energy of the compound nuclei

```

1 // Scilab Code Ex13.6: Page-488(2014)
2 clc; clear;
3 u = 931.5; // Energy equivalent of 1 amu, MeV
4 m_n = 1.0087; // Mass of a neutron, u
5 M_U235 = 235.0439; // Mass of U-235 nucleus, u
6 M_U236 = 236.0456; // Mass of U-236 nucleus, u
7 M_U238 = 238.0508; // Mass of U-238 nucleus, u
8 M_U239 = 239.0543; // Mass of U-239 nucleus, u
9 E_U236 = (m_n + M_U235 - M_U236)*u; // Excitation
    energy of U-236 nucleus, MeV
10 E_U239 = (m_n + M_U238 - M_U239)*u; // Excitation
    energy of U-239 nucleus, MeV
11 printf("\\nThe excitation energy of U-236 nucleus =
    %3.1f MeV", E_U236);
12 printf("\\nThe excitation energy of U-239 nucleus =
    %3.1f MeV", E_U239);
13
14 // Result
15 // The excitation energy of U-236 nucleus = 6.5 MeV
16 // The excitation energy of U-239 nucleus = 4.8 MeV

```

Scilab code Exa 13.7 Nuclear fission through neutron capture

```

1
2 // Scilab Code Ex13.7: Page-490(2014)
3 clc; clear;
4 t = 30e-003; // Time during which the number of
    fissions is to be calculated, s
5 E = 185; // Energy produced for each fission, MeV
6 delta_t = 5e-006; // Average time during which a
    neutron is captured, s
7 fs = t/delta_t; // Number of fission cycles
    within 30 ms
8 N = (1.01)^fs; // Number of fissions that occur
    in 30 ms

```

```

9 E_total = N*E;    // Total energy produced in 30 ms,
    MeV
10 printf("\nThe total number of fissions that occur in
    %d ms = %3.1e", t/1e-003, N);
11 printf("\nThe total energy produced = %3.1e MeV",
    E_total);
12
13 // Result
14 // The total number of fissions that occur in 30 ms
    = 8.5e+025
15 // The total energy produced = 1.6e+028 MeV
16 // The answers are given wrong in the textbook

```

Scilab code Exa 13.8 Fusion reaction in supergiant stars

```

1 // Scilab Code Ex13.8: Page-500(2014)
2 clc; clear;
3 u = 931.5;    // Energy equivalent of 1 amu, MeV
4 M_He = 4.002603;    // Mass of He nucleus, u
5 M_C = 12.0 // Mass of carbon nucleus, u
6 M_O = 15.994915;    // Mass of oxygen nucleus, u
7 Q = (M_He + M_C - M_O)*u;    // Q-value of the
    reaction, MeV
8 printf("\nThe energy expended in the fusion reaction
    inside supergiant star = %3.1f MeV", Q);
9
10 // Result
11 //The energy expended in the fusion reaction inside
    supergiant star = 7.2 MeV

```

Scilab code Exa 13.9 Ignition temperature needed for the fusion reaction between a

```

1 // Scilab Code Ex13.9: Page-502(2014)

```

```

2  clc; clear;
3  k = 1.38e-023;    // Boltzmann constant, J/K
4  r = 3e-015;      // Distance at whcih the nuclear
                    // force becomes effective, m
5  e = 1.6e-019;    // Charge on an electron, C
6  K = 9e+009;      // Coulomb's constant, N-Sq.m/C^2
7  V = K*e^2/r;     // Coulomb potential energy, J
8  // As  $V = 3/2*k*T$ , solving for T
9  T = 2/3*V/k;     // The ignition temperature needed
                    // for the fusion reaction between deuterium and a
                    // tritium, K
10 printf("\\nThe ignition temperature needed for the
        fusion reaction between a deuterium and a tritium
        = %3.1e K", T);
11
12 // Result
13 // The ignition temperature needed for the fusion
    reaction between a deuterium and a tritium = 3.7e
    +009 K

```

Scilab code Exa 13.10 Neutron beam study of atomic structures

```

1  // Scilab Code Ex13.10: Page-509(2014)
2  clc; clear;
3  k = 1.38e-023;    // Boltzmann constant, J/K
4  e = 1.6e-019;    // Energy equivalent of 1 eV, J
5  h = 6.62e-034;    // Planck's oconstant, Js
6  m = 1.67e-027;    // Mass of the neutron, kg
7  lambda = 0.060e-009; // Wavelength of the neutron
                        // beam, m
8  p = h/lambda;     // Momentum of the neutron from de-
                    // Broglie relation, kg-m/s
9  K = p^2/(2*m*e);  // Kinetic energy of the neutron
                    // needed to study atomic structure, eV
10 // As  $K = 3/2*k*T$ , solving for T

```

```
11 T = 2/3*K*e/k;    // The temperature of the neutron
    needed to study atomic structure , K
12 printf("\nThe energy and temperature of the neutron
    needed to study the atomic structure of solids =
    %4.2f eV and %4d K respectively", K, T);
13
14 // Result
15 // The energy and temperature of the neutron needed
    to study the atomic structure of solids = 0.23 eV
    and 1760 K respectively
```

Chapter 14

Particle Physics

Scilab code Exa 14.1 Mass of the meson from Heisenberg uncertainty principle

```
1 // Scilab Code Ex14.1: Page-522(2014)
2 clc; clear;
3 e = 1.6e-019; // Energy equivalent of 1 eV, J
4 h = 6.62e-034; // Planck's constant, Js
5 c = 3.00e+008; // Speed of light in vacuum, m/s
6 h_bar = h/(2*%pi); // Reduced Planck's constant,
   Js
7 R_N = 1e-015; // Range of nuclear force, m
8 // As  $\Delta E \Delta t = \hbar/2$  and  $\Delta E = m_{\text{pion}} c^2$ , solving for  $m_{\text{pion}}$ 
9  $m_{\text{pion}} = \hbar c / (2 R_N e \cdot 10^6)$ ; // Mass of the
   meson, MeV/c2
10 printf("\n\nThe estimated mass of meson from
   Heisenberg uncertainty principle = %d MeV/c2",
   round( $m_{\text{pion}}$ ));
11
12 // Result
13 // The estimated mass of meson from Heisenberg
   uncertainty principle = 99 MeV/c2
14 // The answer is rounded off in the textbook
```

Scilab code Exa 14.2 Range of the weak interaction

```
1 // Scilab Code Ex14.2: Page-525(2014)
2 clc; clear;
3 e = 1.6e-019; // Energy equivalent of 1 eV, J
4 h = 6.62e-034; // Planck's constant, Js
5 c = 3.00e+008; // For simplicity assume speed of
  light to be unity
6 h_bar = h/(2*%pi); // Reduced Planck's constant,
  Js
7 m_W = 80.4; // Energy equivalent of mass of W-
  particle, MeV
8 R_W = h_bar*c/(2*m_W*e*1e+009); // Range of W-
  particle, m
9 delta_t = h_bar/(2*m_W*e*1e+009); // Time during
  which the energy conservation is violated, s
10 printf("\nThe range of W- particle = %3.1e m", R_W);
11 printf("\nThe time during which the energy
  conservation is violated = %1.0e s", delta_t);
12
13 // Result
14 // The range of W- particle = 1.2e-018 m
15 // The time during which the energy conservation is
  violated = 4e-027 s
```

Scilab code Exa 14.10 Fixed target accelerators

```
1 // Scilab Code Ex14.10: Page-548(2014)
2 clc; clear;
3 m_p = 0.938; // Rest mass energy of the proton,
  GeV
```



```

4 K = 6.4;      // Kinetic energy of the proton
   projectile , GeV
5 E_cm = sqrt(2*m_p^2+2*m_p*K);    // Centre of mass
   energy of proton collsion with the fixed proton
   target , GeV
6 Q = 2*m_p - 4*m_p;      // Q value of the reaction ,
   GeV
7 K_th = -3*Q;    // Threshold kinetic energy required
   to produce the antiprotons , GeV
8 K = 1000;      // Kinetic energy of the protons in
   Tevatron , GeV
9 E_cm_T = sqrt(2*m_p^2+2*m_p*K);    // Centre-of-mass
   energy available for the reaction for the
   Tevatron , GeV
10 printf("\nThe available energy in the center on mass
   = %4.2f GeV" , E_cm);
11 printf("\nThe threshold kinetic energy required to
   produce the antiprotons = %3.1f GeV" , K_th);
12 printf("\nThe centre-of-mass energy available for
   the reaction for the Tevatron = %d GeV" , E_cm_T);
13
14 // Result
15 // The available energy in the center on mass = 3.71
   GeV
16 // The threshold kinetic energy required to produce
   the antiprotons = 5.6 GeV
17 // The centre-of-mass energy available for the
   reaction for the Tevatron = 43 GeV

```

Scilab code Exa 14.11 Energy required by a fixed target accelerator to match with

```

1 // Scilab Code Ex14.11: Page-550(2014)
2 clc; clear;
3 m_p = 0.938;    // Rest mass energy of the proton ,
   GeV

```

```
4 E_cm = 14000;    // Centre of mass energy of
    colliding proton beams at LHC, GeV
5 // As  $E_{cm} = \sqrt{2*m_p^2+2*m_p*K}$ , solving for K
6 K = E_cm^2*1e+009/(2*m_p);    // Approx. kinetic
    energy of the protons needed for fixed-target
    experiment, eV
7 printf("\nThe kinetic energy of the protons needed
    for fixed-target experiment = %3.1e eV", K);
8
9 // Result
10 // The kinetic energy of the protons needed for
    fixed-target experiment = 1.0e+017 eV
```

Chapter 15

General Relativity

Scilab code Exa 15.1 Gravitational time dilation effect

```
1 // Scilab Code Ex15.1: Page-562(2014)
2 clc; clear;
3 g = 9.8; // Acceleration due to gravity, m/sec^2
4 H = 10000; // Altitude of the aeroplane above the
   surface of earth, m
5 c = 3.00e+008; // Speed of light in free space, m
   /s
6 T = 45*3600; // Time taken by the airplane to
   from eastward to westward trip, s
7 delta_T_G = g*H*T/(c^2*1e-009); // Time
   difference in the two clocks due to gravitational
   redshift, ns
8 C = 4e+007; // Circumference of the earth, m
9 v = 300; // Speed of the jet airplane, m/s
10 T0 = C/v; // Time of flight of jet airplane very
   near the surface of the earth, s
11 bita = v/c; // Boost parameter
12 // As from special relativity time dilation relation
   ,  $T = T_0 \sqrt{1 - \text{bita}^2}$ , solving for  $T_0 - T =$ 
   delta_T_R, we have
13 delta_T_R = T0*(1-sqrt(1-bita^2))/1e-009; // Time
```

```

    difference in the two clocks due to special
    relativity , ns
14 printf("\nThe gravitational time dilation effect of
    %d ns is larger than the approximate %4.1f ns of
    that of special relativity.", ceil(delta_T_G),
    delta_T_R);
15
16 // Result
17 // The gravitational time dilation effect of 177 ns
    is larger than the approximate 66.7 ns of that of
    special relativity.

```

Scilab code Exa 15.2 Schwarzschild radius for the sun and the earth

```

1 // Scilab Code Ex15.2: Page-567(2014)
2 clc; clear;
3 c = 3.00e+008; // Speed of light in free space , m
    /s
4 G = 6.67e-011; // Newton's gravitational constant
    , N-Sq.m/per kg square
5 M_S = 2.0e+030; // Mass of the sun , kg
6 M_E = 6.0e+024; // Mass of the earth , kg
7 r_S = 2*G*M_S/(c^2*1e+003); // Schwarzschild
    radius for sun , km
8 r_E = 2*G*M_E/(c^2*1e-003); // Schwarzschild
    radius for earth , mm
9 printf("\nThe Schwarzschild radius for sun = %d km",
    ceil(r_S));
10 printf("\nThe Schwarzschild radius for earth = %d mm
    ", ceil(r_E));
11
12 // Result
13 // The Schwarzschild radius for sun = 3 km
14 // The Schwarzschild radius for earth = 9 mm

```

Scilab code Exa 15.3 Time taken by a black hole to radiate its energy

```
1 // Scilab Code Ex15.3: Page-568(2014)
2 clc; clear;
3 c = 3.00e+008; // Speed of light in free space, m
  /s
4 G = 6.67e-011; // Newton's gravitational constant
  , N-Sq.m/per kg square
5 h = 6.62e-034; // Planck's constant, Js
6 h_bar = h/(2*%pi); // Reduced Planck's constant,
  Js
7 sigma = 5.67e-008; // Stefan-Boltzmann constant,
  W per Sq.m per K^4
8 k = 1.38e-023; // Boltzmann constant, J/K
9 M0 = 1.99e+030; // Mass of the sun, kg
10 alpha = 2*sigma*h_bar^4*c^6/((8*%pi)^3*k^4*G^2);
  // A constant, kg^3/s
11 t = integrate('1/alpha*M^2', 'M', 0, 3*M0);
12 printf("\nThe time required for the 3-solar-mass
  black hole to evaporate = %3.1e y", t
  /(365.25*24*60*60));
13
14 // Result
15 // The time required for the 3-solar-mass black hole
  to evaporate = 5.7e+068 y
```

Chapter 16

Cosmology and Modern Astrophysics

Scilab code Exa 16.1 Hubble constant determination

```
1 // Scilab Code Ex16.1: Page-581(2014)
2 clc; clear;
3 H0 = 22; // Value of Hubble constant, km/s per
  million ly
4 parsec = 3.26; // The value of 1 parsec, light
  years
5 printf("\nThe value of Hubble constant = %d km/s per
  Mpc", ceil(H0*parsec));
6
7 // Result
8 // The value of Hubble constant = 72 km/s per Mpc
```

Scilab code Exa 16.2 Current ratio of protons to neutrons in the universe

```
1 // Scilab Code Ex16.2: Page-583(2014)
2 clc; clear;
```

```

3 M = 1;      // Let the current mass of the universe be
              unity
4 m_u = 1;    // Mass equivalent of 1 amu, u
5 N_n = 2;    // Number of neutrons in helium
6 N_p = 2;    // Number of protons in helium
7 M_p = 0.75*M*m_u; // Total mass of protons
8 M_He = 0.25*M*m_u; // Total mass of helium
9 N_frp = M_p/M_He*(N_n + N_p); // Total number of
              free protons for every He-4
10 N_P = N_frp + N_p; // Total number of protons per
                    He-4
11 ratio = N_P/N_n; // Current ratio of protons to
                    the neutrons in the universe
12 printf("\nThe current ratio of protons to the
          neutrons in the universe = %d", ratio);
13
14 // Result
15 // The current ratio of protons to the neutrons in
          the universe = 7

```

Scilab code Exa 16.3 Ratio of protons to neutrons at 10 billion kelvin temperature

```

1 // Scilab Code Ex16.3: Page-587(2014)
2 clc; clear;
3 m_n = 939.566; // Rest mass of the neutron, MeV/c
                ^2
4 m_p = 938.272; // Rest mass of the proton, MeV/c
                ^2
5 e = 1.6e-019; // Energy equivalent of 1 eV, J
6 c = 1; // For simplicity assume speed of light of
          light to be unity
7 T = 1e+010; // Temperature of the universe, K
8 delta_m = m_n - m_p; // Mass difference between a
                       proton and a neutron, MeV/c^2
9 k = 1.38e-023; // Boltzmann constant, J/k

```

```

10 // As from Maxwell-Boltzmann distribution from
    thermodynamics,  $N = \exp(-m*c^2/(k*T))$ , so
11 ratio = exp(delta_m*c^2*1e+006*e/(k*T)); // Ratio
    of protons to neutrons in the universe at 10
    billion kelvin
12 printf("\nThe ratio of protons to neutrons in the
    universe at 10 billion kelvin = %3.1f", ratio);
13
14 // Result
15 // The ratio of protons to neutrons in the universe
    at 10 billion kelvin = 4.5

```

Scilab code Exa 16.4 Mean temperature of the sun

```

1 // Scilab Code Ex16.4: Page-589(2014)
2 clc; clear;
3 M = 1.99e+030; // Mass of the sun, kg
4 G = 6.67e-011; // Universal gravitational
    constant, N-Sq.m/kg^2
5 k = 1.38e-023; // Boltzmann constant, J/K
6 R = 6.96e+008; // Radius of the sun, m
7 m = 1.67e-027; // Rest mass of the proton, kg
8 PE = 3/5*(G*M^2/R); // Self potential energy of
    the sun, J
9 // As  $KE = 1/3*(M/m_p)*m_p*v^2$ , solving for v
10 v = sqrt(2*PE/M); // Velocity of a proton inside
    the sun, m/s
11 // From kinetic theory of gases,  $v = \sqrt{3*k*T/m}$ ,
    solving for T
12 T = m*v^2/(3*k); // The mean temperature of the
    sun, K
13 printf("\nThe mean temperature of the sun = %1.0e K",
    , T);
14
15 // Result

```


16 // The mean temperature of the sun = 9e+006 K

Scilab code Exa 16.5 Radius of the neutron star

```
1 // Scilab Code Ex16.5: Page-590(2014)
2 clc; clear;
3 M_sun = 1.99e+030; // Mass of the sun, kg
4 m_n = 1.675e-027; // Rest mass of the neutron, kg
5 h = 6.62e-034; // Planck's constant, Js
6 h_bar = h/(2*pi); // Planck's constant, Js
7 G = 6.67e-011; // Universal gravitational
  constant, N-Sq.m/kg^2
8 N = 2*M_sun/m_n; // Number of neutrons in the
  neutron star
9 V = (6.5*h_bar^2/(N^(1/3)*m_n^3*G))^3; // Volume
  of the neutron star, metre cube
10 R = (3/(4*pi)*V)^(1/3); // The radius of neutron
  star, m
11 printf("\nThe radius of the neutron star of 2 solar
  masses = %d km", ceil(R/1e+003));
12
13 // Result
14 // The radius of the neutron star of 2 solar masses
  = 11 km
```

Scilab code Exa 16.6 Redshift versus recession velocity

```
1 // Scilab Code Ex16.6: Page-598(2014)
2 clc; clear;
3 c = 1; // Assume speed of light to be unity
4 clf();
```

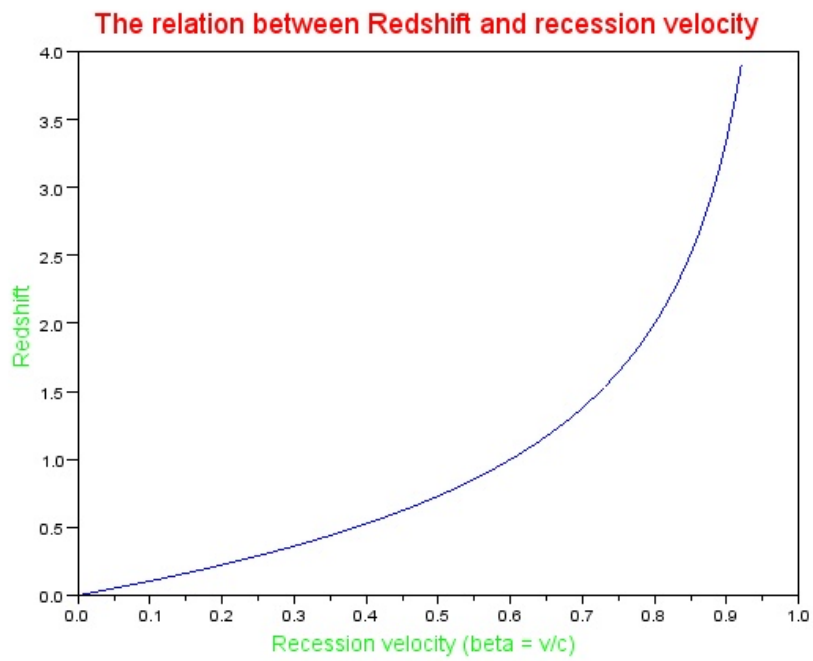


Figure 16.1: Redshift versus recession velocity

```

5 v = [0:0.01:0.92]';
6 bita = v/c;           // Recession velocity ratio
7 for i = 1:1:93
8     red_shift(i) = sqrt((1+bita(i))/(1-bita(i)))-1;
9 end
10 plot(bita, red_shift);
11 title('The relation between Redshift and recession
        velocity', 'fontsize', 4, 'color', 'red', '
        position', [0.02, 4.1]);
12 xlabel('Recession velocity (beta = v/c)', 'fontsize'
        , 3, 'color', 'green');
13 ylabel('Redshift', 'fontsize', 3, 'color', 'green');
14
15 // Result
16 // The plot between Redshift vs recession velocity
    is as shown in the Fig.

```

Scilab code Exa 16.7 Difference in the travel times of different mass neutrinos

```

1 // Scilab Code Ex16.7: Page-598(2014)
2 clc; clear;
3 c = 1; // For simplicity assume speed of light to
        be unity, m/s
4 d = 1.6e+005; // Distance of the supernova 1987A
        from the earth, ly
5 m = 16; // Mass of heavier neutrino, eV/c^2;
6 E = 20e+006; // Energy of the neutrino, eV
7 delta_t = d/(2*c)*(m/E)^2; // Difference between
        the travel times of the lighter and the massive
        neutrinos, y
8 printf("\nThe difference between the travel times of
        the lighter and the massive neutrinos = %3.1f s"
        , delta_t*(365.25*24*60*60));
9
10 // Result

```

```
11 // The difference between the travel times of the
    lighter and the massive neutrinos = 1.6 s
```

Scilab code Exa 16.8 Critical density of the universe

```
1 // Scilab Code Ex16.8: Page-602(2014)
2 clc; clear;
3 c = 3.00e+008; // Speed of light, m/s
4 H = 22; // Hubble constant, km/s per million ly
5 G = 6.67e-011; // Universal gravitational
    constant, N-Sq.m/kg^2
6 rho_c = 3/(8*%pi)*H^2/G*1e+003/(c*365.25*24*60*60*1e
    +006)^2; // The critical density of the
    universe, g/cc
7 printf("\nThe critical density of the universe = %3
    .1e g/cc", rho_c);
8
9 // Result
10 //The critical density of the universe = 9.7e-030 g/
    cc
```

Scilab code Exa 16.9 Upper limit of the age of the universe

```
1 // Scilab Code Ex16.9: Page-604(2014)
2 clc; clear;
3 H0 = 71; // Hubble constant, km/s per Mpc
4 tau = 1/H0*1e+006*3.26*9.46e+012/3.16e+007; //
    The upper limit of the age of the universe, y
5 printf("\nThe upper limit of the age of the universe
    = %4.2e y", tau);
6
7 // Result
```

8 // The upper limit of the age of the universe = 1.37
e+010 y
