

Scilab Textbook Companion for
Concepts of Physics (Volume - 1)
by H. C. Verma¹

Created by
Patel Chaitanya Kishorbhai
B. Tech.
Others
Dharmsinh Desai University
College Teacher
Prarthan Mehta
Cross-Checked by
K. V. P. Pradeep

July 31, 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: Concepts of Physics (Volume - 1)

Author: H. C. Verma

Publisher: Bharati Bhavan,india

Edition: 2

Year: 2011

ISBN: 9788177091878

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 Physics and Mathematics	5
3 Rest and Motion Kinematics	23
4 The Forces	45
5 Newton s Laws of Motion	47
6 Friction	52
7 Circular Motion	61
8 Work and Energy	71
9 Centre of Mass Linear momentum Collision	78
10 Rotational Mechanics	88
11 Gravitation	100
12 Simple Harmonic Motion	113
13 Fluid Mechanics	128

14 Some Mechanical Properties of Matter	139
15 Wave Motion and Waves on a String	158
16 Sound Waves	174
17 Light Waves	197
18 Geometrical Optics	210
19 Optical Instruments	240
20 Dispersion and Spectra	251
22 Photometry	256

List of Scilab Codes

Exa 2.1w	calculation of magnitude and direction of vector	5
Exa 2.2	calculation of sum of vectors and difference of the vectors	6
Exa 2.2w	calculation of resultant of three vectors	6
Exa 2.3	calculation of component of force in vertical direction	8
Exa 2.3w	calculation of resultant of the vectors	8
Exa 2.4	calculation of work done by the force during displacement	9
Exa 2.4w	calculation of direction of resultant vector	10
Exa 2.5	calculation of angle between two vectors from known value of their cross product	11
Exa 2.6	calculation of the slope of curve at a given point	11
Exa 2.6w	calculation of angle	12
Exa 2.7w	calculation of unit vector	13
Exa 2.9	evaluation of a integral	13
Exa 2.9w	calculation of angle between two vectors	14
Exa 2.10	calculation of round off value upto three digits	15
Exa 2.10w	calculation of value of the given scalar	16
Exa 2.11	calculation of value	17
Exa 2.11w	calculation of change in volume of sphere as radius is changed	17
Exa 2.12	calculation of value	18
Exa 2.13	calculation of average focal length of concave mirror considering uncertainty	19

Exa 2.13w	calculation of maximum and minimum value of a given function	20
Exa 2.14w	calculation of the area under curve	20
Exa 2.18w	calculation of value	21
Exa 3.1	calculation of distance and displacement	23
Exa 3.1w	calculation of average speed of the walk	23
Exa 3.2	calculation of average speed and instantaneous speed	24
Exa 3.2w	calculation of average speed and average velocity	25
Exa 3.3	calculation of distance from speed versus time graph	26
Exa 3.3w	calculation of average velocity and average acceleration	26
Exa 3.4	calculation of average velocity of the tip of minute hand in a table clock	28
Exa 3.4w	calculation of distance travelled displacement and acceleration	28
Exa 3.5	calculation of distance travelled in given time time taken to reach a particular velocity and distance covered to reach particular velocity	29
Exa 3.5w	calculation of acceleration and distance travelled	30
Exa 3.6	calculation of displacement of particle in last 1 second	31
Exa 3.6w	calculation of acceleration	32
Exa 3.7	calculation of maximum height reached by the ball	32
Exa 3.8	calculation of velocity and position of the particle	33
Exa 3.8w	calculation of total distance and number of trips	34
Exa 3.9	calculation of horizontal range of the projectile	35
Exa 3.9w	drawing graph of x versus t v versus t and a versus t	35
Exa 3.10	calculation of velocity of the swimmer with respect to ground	37

Exa 3.10w	calculation of height of balloon when stone reaches ground	37
Exa 3.11	calculation of velocity of the raindrops with respect to the man	39
Exa 3.11w	calculation of time of flight horizontal range and vertical range	39
Exa 3.16w	calculation of angle of the swim and time to cross the river	40
Exa 3.17w	calculation of time taken and position of the arrival on opposite bank	41
Exa 3.18w	calculation of speed of raindrops with respect to road and the moving man	42
Exa 3.19w	calculation of speed and direction of rain with respect to the road	43
Exa 4.1	calculation of coulomb force	45
Exa 4.3w	calculation of ratio of the electrical force to the gravitational force between two electrons	46
Exa 5.1	calculation of force exerted by the string on a particle	47
Exa 5.3w	calculation of the force exerted by the tree limb on the bullet	48
Exa 5.4w	calculation of the position of a particle	48
Exa 5.7w	calculation of acceleration with which ring starts moving if released from rest at an angle θ	49
Exa 5.8w	calculation of the maximum acceleration of the man for safe climbing	50
Exa 6.1	calculation of the angle made by the contact force with the vertical and the magnitude of contact force	52
Exa 6.1w	calculation of the maximum angle to prevent slipping	53
Exa 6.2	calculation of the force of friction exerted by the horizontal surface on the box	53
Exa 6.2w	calculation of frictional force and minimum value of coefficient of static friction	54
Exa 6.3	calculation of the force of friction exerted by the horse and condition of boy for sliding back	55

Exa 6.3w	calculation of the maximum value of mass of the block	55
Exa 6.4	calculation of coefficient of static friction and kinetic friction between the block and the plank	56
Exa 6.5w	calculation of the coefficient of kinetic friction	57
Exa 6.6w	calculation of the values of coefficient of static and kinetic friction	58
Exa 6.10w	calculation of minimum and maximum values of mass and the acceleration if given a gentle push	59
Exa 7.1	calculation of the angular velocity	61
Exa 7.1w	calculation of the maximum speed the car can take on the turn without skidding	61
Exa 7.2	calculation of the angular acceleration	62
Exa 7.2w	calculation of the value of angle of banking	63
Exa 7.3	calculation of the magnitude of linear acceleration	64
Exa 7.4	calculation of the value of radial and tangential acceleration	64
Exa 7.4w	calculation of the value of elongation of the spring	65
Exa 7.5	calculation of the normal contact force by the side wall of the groove	66
Exa 7.6	calculation of the speed of vehicle on the turn	66
Exa 7.7	calculation of the weight of the body if spring balance is shifted to the equator	67
Exa 7.7w	calculation of the value of force exerted by the air on the plane	68
Exa 7.8w	calculation of the angular speed of rotation	68
Exa 7.10w	calculation of the minimum speed at which floor may be removed	69
Exa 8.1	calculation of the work done by the spring force	71
Exa 8.1w	calculation of the work done by the porter on the suitcase	71
Exa 8.2	calculation of the work done by force of gravity	72
Exa 8.2w	calculation of the value of minimum horsepower of the motor to be used	73

Exa 8.3w	calculation of the power delivered by the pulling force and average power	73
Exa 8.4w	calculation of the work done by the given force	74
Exa 8.5	calculation of the speed of the pendulum of bob when it makes an angle of 60 degree with the vertical	75
Exa 8.11w	calculation of the speed of the particle at a given point	76
Exa 8.12w	calculation of the maximum compression of the spring	76
Exa 9.1w	Locating the centre of maass of the system .	78
Exa 9.4	calculation of the maximum compression of the string	79
Exa 9.5	calculation of the speed of combined mass .	79
Exa 9.6w	calculation of the acceleration of the centre of mass	80
Exa 9.8w	calculation of the distance from launching point	81
Exa 9.9w	calculation of the distance moved by the bigger block	82
Exa 9.10w	calculation of the average force exerted by the hero on the machine gun	83
Exa 9.11w	calculation of the fractional change in kinetic energy	83
Exa 9.13w	calculation of the final velocity of the shuttle	84
Exa 9.14w	calculation of the velocity with which the board recoils	85
Exa 9.17w	calculation of the speed of the bullet	85
Exa 9.22w	calculation of the loss of kinetic energy due to the collision	86
Exa 10.1	calculation of the number of revolutions made	88
Exa 10.1w	calculation of the number of revolutions made by the wheel	89
Exa 10.2	calculation of the time taken by the fan to attain half of the maximum speed	89
Exa 10.2w	calculation of the angle rotated during the next second	90
Exa 10.3	calculation of the angular velocity and angular acceleration of the pulley	91

Exa 10.3w	calculation of the torque required to stop the wheel in one minute	91
Exa 10.4w	calculation of the angular velocity of the wheel	92
Exa 10.5	calculation of the moment of inertia of the wheel	93
Exa 10.7w	calculation of the position of second kid on a balanced seesaw	93
Exa 10.8w	calculation of the normal force and the frictional force that the floor exerts on the ladder	94
Exa 10.9w	calculation of the contact force exerted by the floor on each leg of ladder	95
Exa 10.12	calculation of the kinetic energy of the sphere	96
Exa 10.13w	calculation of the kinetic energy and angular momentum of the disc	96
Exa 10.14w	calculation of the work done by the torque in first two seconds	97
Exa 10.19w	calculation of the moment of inertia of the system about the axis perpendicular to the rod passing through its middle point	98
Exa 10.22w	calculation of the number of revolutions made by the wheel per second	99
Exa 11.1	calculation of the initial acceleration of the particles	100
Exa 11.2	calculation of the work done in bringing three particles together	101
Exa 11.2w	calculation of the distance from the earth surface where resultant gravitational field due to the earth and the moon is zero	101
Exa 11.4	calculation of the gravitational field	102
Exa 11.4w	calculation of the separation between the particles under mutual attraction	103
Exa 11.5w	calculation of the work done by an external agent	104
Exa 11.7	calculation of the gravitational field due to the moon at its surface	105
Exa 11.8	calculation of the value of acceleration due to gravity	105

Exa 11.9	calculation of the speed and time period of the satellite	106
Exa 11.9w	calculation of the maximum height attained by the particle	107
Exa 11.10	calculation of the escape velocity from the moon	108
Exa 11.10w	calculation of the stretch produced in the spring	108
Exa 11.11w	calculation of time period of the pendulum if used at the equator	109
Exa 11.12w	calculation of the speed of projection of the satellite into an orbit	110
Exa 11.13w	calculation of the speed and the angular speed of the satellite S2 relative to the satellite S1	111
Exa 12.1	calculation of the spring constant	113
Exa 12.1w	calculation of the amplitude time period maximum speed and velocity at time t	113
Exa 12.2	calculation of the amplitude of the motion .	114
Exa 12.2w	calculation of the maximum force exerted by the spring on the block	115
Exa 12.3	calculation of the time period of oscillation of the particle	115
Exa 12.3w	calculation of the maximum time period maximum speed maximum acceleration speed for a given displacement speed at a given time .	116
Exa 12.4	calculation of the value of phase constant . .	117
Exa 12.5	calculation of the total mechanical energy of the system	118
Exa 12.6	writing the equation giving angular displacement as a function of time	118
Exa 12.6w	calculation of the maximum speed of the block and the speed when the spring is stretched	119
Exa 12.7	calculation of the time period of a pendulum	120
Exa 12.8	calculation of the value of the acceleration due to gravity	121
Exa 12.9	calculation of the time period of oscillation .	121
Exa 12.10	calculation of the value of torsional constant of the wire	122

Exa 12.11	calculation of the amplitude of the simple harmonic motion	123
Exa 12.14w	calculation of the time period linear amplitude speed and angular acceleration	123
Exa 12.16w	calculation of the time period of small oscillations	124
Exa 12.18w	calculation of the time period of small oscillation about the point of suspension	125
Exa 12.19w	calculation of the moment of inertia of the second disc about the wire	126
Exa 12.22w	calculation of the phase difference between the individual motions	127
Exa 13.1	calculation of the force exerted by the water on the bottom	128
Exa 13.1w	calculation of the force exerted by the mercury on the bottom of the beaker	129
Exa 13.2	calculation of the volume of the cube outside the water	129
Exa 13.2w	calculation of the height of the atmosphere to exert the same pressure as at the surface of the earth	130
Exa 13.3	calculation of the speed of the outgoing liquid	131
Exa 13.3w	calculation of the height of the water column	131
Exa 13.4	calculation of the difference in the pressures at A and B point	132
Exa 13.5	calculation of the speed of the water coming out of the tap	133
Exa 13.5w	calculation of the force applied on the water in the thicker arm	133
Exa 13.6w	calculation of the elongation of the spring .	134
Exa 13.7w	calculation of the maximum weight that can be put on the block without wetting it	135
Exa 13.8w	calculation of the angle that the plank makes with the vertical in equilibrium	135
Exa 13.10w	calculation of the rate of water flow through the tube	136
Exa 13.11w	calculation of the velocity of the water coming out of the opening	137

Exa 14.1	calculation of the tensile stress developed in the wire	139
Exa 14.1w	calculation of the extension of the wire . . .	140
Exa 14.2	calculation of the value of Young modulus .	140
Exa 14.2w	calculation of the elongation of the rope and corresponding change in the diameter	141
Exa 14.3	calculation of the elastic potential energy stored in the stretched steel wire	142
Exa 14.3w	calculation of the minimum radius of the wire used if it is not to break	142
Exa 14.4	calculation of the force by which the surface on one side of the diameter pulls the surface on the other side	143
Exa 14.4w	calculation of the ratio of the lengths of the two wire	144
Exa 14.5	calculation of the gain in the surface energy	145
Exa 14.5w	calculation of the decrease in the volume of the sample of water	145
Exa 14.6	calculation of the excess pressure inside a mercury drop	146
Exa 14.6w	calculation of the longitudinal strain in two wires	147
Exa 14.7	calculation of the density of the liquid . . .	147
Exa 14.7w	calculation of the longitudinal strain developed in each wire	148
Exa 14.8	calculation of the height of the water in the column	149
Exa 14.8w	calculation of the elastic potential energy stored in the wire	150
Exa 14.9	calculation of the value of the coefficient of viscosity of the solution	151
Exa 14.9w	calculation of the elongation of the wire . .	151
Exa 14.11w	calculation of the amount by which the pressure inside the bubble is greater than the atmospheric pressure	152
Exa 14.12w	calculation of the load W suspended from wire to keep it in equilibrium	153
Exa 14.13w	calculation of the radius of the capillary tube	154

Exa 14.15w	calculation of the tangential force needed to keep the plate moving	154
Exa 14.16w	calculation of the shearing stress between the horizontal layers of water	155
Exa 14.17w	calculation of the terminal velocity of the rain drop	156
Exa 15.1	calculation of the velocity function v giving displacement function g_x giving shape . . .	158
Exa 15.1w	calculation of the amplitude wavelength frequency speed of the wave	159
Exa 15.2	calculation of the amplitude wave number wavelength frequency time period wave velocity .	160
Exa 15.2w	calculation of the maximum velocity and acceleration of the particle	160
Exa 15.3	calculation of the time taken by the pulse in travelling through a distance	161
Exa 15.3w	calculation of the speed and displacement of the particle	162
Exa 15.4	calculation of the power transmitted through a given point	163
Exa 15.4w	calculation of the extension of the wire over its natural length	164
Exa 15.5	calculation of the phase difference between the waves and amplitude of the resultant wave	165
Exa 15.5w	calculation of the wavelength of the pulse when it reaches the top of the rope	165
Exa 15.6	calculation of the velocity node closest to origin antinode closest to origin amplitude at x	166
Exa 15.6w	calculation of the displacement of the particle	167
Exa 15.7	calculation of the fundamental frequency of the portion of the string between the wall and the pulley	168
Exa 15.7w	calculation of the maximum displacement wavelengths and wave speed velocity nodes and antinodes number of loops	169
Exa 15.8	calculation of the length of the experimental wire to get the resonance	170

Exa 15.8w	calculation of the pressing in the guitar to produce required fundamental frequency . . .	171
Exa 15.9w	calculation of the position of bridges in sonometer wire	171
Exa 15.10w	calculation of the length of the wire	172
Exa 16.1	calculation of the audibility of a wave	174
Exa 16.1w	calculation of the depth of the sea and wavelength of the signal in the water	175
Exa 16.2	calculation of the amplitude of vibration of the particles of the medium	175
Exa 16.2w	calculation of the location of the plane	176
Exa 16.3	calculation of the intensity of the sound wave	177
Exa 16.3w	calculation of the frequency wavelength speed maximum and minimum pressures of the sound wave	177
Exa 16.4	calculation of the increase in the sound level in decibels	178
Exa 16.4w	calculation of the minimum separation between the two points for a given phase difference	179
Exa 16.5	calculation of the nature of interference	179
Exa 16.5w	calculation of the atmospheric temperature	180
Exa 16.6	calculation of the distance of the piston from the open end for tube to vibrate in its first overtone	181
Exa 16.6w	calculation of the speed of sound wave in hydrogen	181
Exa 16.7	calculation of the tuning frequency of fork B	182
Exa 16.7w	calculation of the energy delivered to the microphone	183
Exa 16.8	calculation of the most dominant frequency	183
Exa 16.8w	calculation of the amplitude of vibration of the particles of the air	184
Exa 16.9w	calculation of the factor by which the pressure amplitude increases	185
Exa 16.10w	calculation of the frequency at which the maxima of intensity are detected	185

Exa 16.11w	calculation of the minimum distance between the source and the detector for maximum sound detection	186
Exa 16.12w	calculation of the length of the shortest closed organ pipe that will resonate with the tuning fork	187
Exa 16.13w	calculation of the length of the closed pipe .	188
Exa 16.14w	calculation of the speed of the sound in air .	188
Exa 16.15w	calculation of the fundamental frequency if the air is replaced by hydrogen	189
Exa 16.16w	calculation of the speed wavelength in the rod frequency wavelength in the air	190
Exa 16.17w	calculation of the frequency of the note emitted by the taut string	191
Exa 16.18w	calculation of the apparent frequency	191
Exa 16.19w	calculation of the frequency of the whistle of the train	192
Exa 16.20w	calculation of the main frequency heard by the person	193
Exa 16.21w	calculation of the original frequency of the source	193
Exa 16.22w	calculation of the speed of the car	194
Exa 16.23w	calculation of the frequency of train whistle heard by the person standing on the road perpendicular to the track	195
Exa 17.1	calculation of the speed of light in glass	197
Exa 17.1w	calculation of the limits of wavelengths in the water	197
Exa 17.2	calculation of the separation between successive bright fringes	198
Exa 17.2w	calculation of the refractive index of the glass .	199
Exa 17.3	calculation of the wavelength of light in the water	199
Exa 17.3w	calculation of the wavelengths of the violet and the red light calculation of the wavelengths of the violet and the red light	200
Exa 17.4	calculation of the minimum thickness of the film	201

Exa 17.4w	calculation of the separation between the slits	201
Exa 17.5	calculation of the angular divergence for most of the light getting diffracted	202
Exa 17.5w	calculation of the ratio of maximum intensity to the minimum intensity in the interference fringe pattern	203
Exa 17.6	calculation of the diameter of the disc image	203
Exa 17.6w	calculation of the ratio of maximum intensity to the minimum intensity in the interference pattern	204
Exa 17.7w	calculation of the maximum and the minimum path difference at the detector	205
Exa 17.8w	calculation of the distance of bright fringe from the central maximum	206
Exa 17.9w	calculation of the number of fringes that will shift due to introduction of the sheet	207
Exa 17.10w	calculation of the wavelengths in the visible region that are strongly reflected	207
Exa 17.11w	calculation of the distance between the two first order minima	208
Exa 18.1	calculation of position of the image of an object placed at a distance from the mirror	210
Exa 18.1w	calculation of position and nature of the image of an object placed at a distance from a concave mirror	211
Exa 18.2	calculation of length of the image of an object placed at a distance from a concave mirror	211
Exa 18.2w	calculation of length of the image of an object placed horizontal at a distance from the mirror	212
Exa 18.3	calculation of shift in the position of printed letters by a glass cube	213
Exa 18.3w	calculation of object distance for half image height as compared to original height in case of reflection by convex mirror	213
Exa 18.4	calculation of refractive index of material from known critical angle	214
Exa 18.4w	calculation of image distance and focal length of concave mirror	215

Exa 18.5	calculation of refractive index of material from known value of angle of minimum deviation by prism	216
Exa 18.5w	calculation of maximum angle of reflection for a surface	216
Exa 18.6	calculation of position of the image of an object placed at a distance from spherical convex surface	217
Exa 18.6aw	calculation of minimum refractive index for parallel emergence for given condition in right prism	218
Exa 18.6bw	verification of total internal reflection for given conditions of right prism	218
Exa 18.7	calculation of the size of the image of an object placed at a distance from the spherical concave surface	219
Exa 18.8	calculation of focal length of a biconvex lens from known value of radii of curvature of refracting surfaces	220
Exa 18.9	calculation of size of the image of an object placed at a distance from a convex lens	221
Exa 18.11w	locating image of a dust particle on the surface of water filled in a concave mirror as observed from top	222
Exa 18.12w	calculation of position of final image formed due to a system of glass slab and a concave mirror	223
Exa 18.13w	calculation angle of minimum deviation for equilateral prism of silicate flint glass from known value of wavelength	223
Exa 18.14w	calculation of angle of rotation of the mirror in given setup	224
Exa 18.15w	calculation of location of the image of an object placed at a distance from the spherical convex surface	225
Exa 18.16w	calculation of height of the image of an object placed along axis at a distance from a horizontal cylindrical glass rod	226

Exa 18.17w	calculation of apparent depth of a air bubble inside a glass sphere	227
Exa 18.18w	calculation of position of image due to refraction at the first surface and position of final image	228
Exa 18.19w	calculation of focal length of thin lens	229
Exa 18.20w	calculation of position of diverging mirror to obtain real image at the source itself for given system	230
Exa 18.21w	calculation of separation between mirror and the lens for parallel emergence of the final beam	230
Exa 18.22w	calculation of object distance from the lens with one side silvered	232
Exa 18.23w	calculation of location of image of an object placed in front of a concavo convex lens made of glass	233
Exa 18.24w	calculation of new focal length of a lens on immersing in water	234
Exa 18.25w	calculation of location of final image for an object on the axis of a cylindrical tube containing water closed by an equiconvex lens .	235
Exa 18.26w	calculation of new position of the slide of projector if the position of the screen is changed	236
Exa 18.27w	calculation of position of the object to get a focused image	237
Exa 18.28aw	finding the image of a distant object formed by combination of two convex lens by using thin lens formula	237
Exa 18.28bw	finding the image of a distant object formed by combination of two convex lens by using equivalent lens method	238
Exa 19.1	determining which boy appears taller	240
Exa 19.1w	calculation of the angular magnification	241
Exa 19.2	calculation of the angular magnification and the length of the microscope tube	241

Exa 19.2w	calculation of the object distance to obtain maximum angular magnification for a normal eye	242
Exa 19.3	calculation of the power of lens for the spectacles	243
Exa 19.3w	calculation of the position of the image linear magnification and the angular magnification	243
Exa 19.4w	calculation of the object distance and the angular magnification	244
Exa 19.5w	calculation of the object distance and the angular magnification for the least strain in the eyes	245
Exa 19.6w	calculation of the length of the tube and the angular magnification produced by the telescope	246
Exa 19.7w	calculation of the tube length magnifying power and angular magnification	246
Exa 19.8w	calculation of the angular magnification due to the converging lens	247
Exa 19.9w	calculation of the power of lens and maximum distance that can be seen clearly	248
Exa 19.10w	calculation of the near point and the distance of the retina from the lens	249
Exa 20.1	calculation of the dispersive power of the flint glass	251
Exa 20.1w	calculation of the angular dispersion produced by a thin prism of the flint glass	252
Exa 20.2	calculation of the dispersive power of the material of the lens	252
Exa 20.2w	calculation of the angle of flint glass prism and angular dispersion produced by the combination	253
Exa 20.3w	calculation of the refracting angles of the two prisms	254
Exa 22.1	calculation of the luminous flux	256
Exa 22.1w	calculation of the total radiant flux total luminous flux and the luminous efficiency	256

Exa 22.2w	calculation of the total luminous flux emitted by the source and the total luminous intensity of the source	257
Exa 22.3w	calculation of the luminous flux falling on a plane	258
Exa 22.4w	calculation of the illuminance at a small surface area of the table top	259
Exa 22.5w	calculation of the luminous flux emitted into a cone of given solid angle	260

Chapter 2

Physics and Mathematics

Scilab code Exa 2.1w calculation of magnitude and direction of vector

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.1w
5 //calculation of magnitude and direction of vector
6
7 //given data
8 xcomp=25; //value of component along X axis
9 ycomp=60; //value of component along Y axis
10 theta=90; //angle between X and Y axis
11
12 //calculation
13 A=sqrt((xcomp*xcomp)+(ycomp*ycomp)+(2*xcomp*ycomp*
      cosd(theta)));
14 alpha=atand(ycomp/xcomp);
15
16 disp(A,'magnitude of the vector is');
17 disp(alpha,'direction of the vector is');
```

Scilab code Exa 2.2 calculation of sum of vectors and difference of the vectors

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 2.2
5 //calculation of sum of vectors and difference of
   the vectors
6
7 //given data
8 A=5; //magnitude(in unit) of A vector
9 B=5; //magnitude(in unit) of B vector
10 theta=60; // angle(in degree) between both vectors
11
12
13 //calculation
14 C=sqrt((A*A)+(B*B)+(2*A*B*cosd(theta))); //C=|A+B|
   sum of two vectors
15 thetas=180-theta; //for difference(subtraction)
   reverse direction of a vector and add it to other
16 D=sqrt((A*A)+(B*B)+(2*A*B*cosd(thetas))); //D=|A-B|
   difference of two vectors
17
18 disp(C, 'the sum of two vectors(in unit) is ');
19 disp(D, 'the difference of two vectors(in unit) is ');
```

Scilab code Exa 2.2w calculation of resultant of three vectors

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 2.2w
5 //calculation of resultant of three vectors
6
7 //given data
```



```

8 theta1=37; //value of angle(in degree) of first
   vector with X axis
9 theta2=0; //value of angle(in degree) of second
   vector with X axis
10 theta3=90; //value of angle(in degree) of third
   vector with X axis
11 x=5; //magnitude(in m) of first vector
12 y=3; //magnitude(in m) of second vector
13 z=2; //magnitude(in m) of third vector
14
15 //calculation
16 xcomp1=x*cosd(theta1); //xcomponent(in m) of first
   vector
17 ycomp1=x*sind(theta1); //ycomponent(in m) of first
   vector
18 xcomp2=y*cosd(theta2); //xcomponent(in m) of second
   vector
19 ycomp2=y*sind(theta2); //ycomponent(in m) of second
   vector
20 xcomp3=z*cosd(theta3); //xcomponent(in m) of third
   vector
21 ycomp3=z*sind(theta3); //ycomponent(in m) of third
   vector
22
23 xcompr=xcomp1+xcomp2+xcomp3; //xcomponent(in m) of
   resultant vector
24 ycompr=ycomp1+ycomp2+ycomp3; //ycomponent(in m) of
   resultant vector
25
26 r=sqrt((xcompr*xcompr)+(ycompr*ycompr)); //magnitude
   (in m) of resultant vector
27 theta=atand(ycompr/xcompr); //value of angle(in
   degree) of resultant vector with X axis
28
29 disp(r, 'magnitude(in m) of resultant vector is ');
30 disp(theta, 'value of angle(in degree) of resultant
   vector with X axis ');

```

Scilab code Exa 2.3 calculation of component of force in vertical direction

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.3
5 //calculation of component of force in vertical
   direction
6
7 //given data
8 F=10.5 //force(in newton) acting on the particle
9 theta=37 //angle(in degree) at which force acts
10
11 //calculation
12 Fp=F*cosd(theta); //component of force in vertical
   direction
13
14 disp(Fp,'component of force(in newton) in vertical
   direction is');
```

Scilab code Exa 2.3w calculation of resultant of the vectors

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.3w
5 //calculation of resultant of the vectors
6
7 //given data
8 //theta1=90; value of angle(in degree) of OA vector
9 //theta2=0; value of angle(in degree) of OB vector
10 //theta3=135; value of angle(in degree) of OC vector
```

```

11 OA=5; //magnitude(in m) of OA vector
12 //OB=magnitude(in m) of OB vector
13 //OC=magnitude(in m) of OC vector
14
15 //calculation
16 //xcomp1=0; xcomponent(in m) of OA vector
17 //ycomp1=-OA; ycomponent(in m) of OA vector
18 //xcomp2=OB; xcomponent(in m) of OB vector
19 //ycomp2=0; ycomponent(in m) of OB vector
20 //xcomp3=(-1/sqrt(2))*OC; xcomponent(in m) of OC
    vector
21 //ycomp3=(1/sqrt(2))*OC; ycomponent(in m) of OC
    vector
22
23 //xcompr=OB-((1/sqrt(2))*OC); xcomponent(in m) of
    resultant vector=0(given)      (1)
24 //therefore OB=((1/sqrt(2))*OC)
    (2)
25 //ycompr=((1/sqrt(2))*OC)-OA; ycomponent(in m) of
    resultant vector
26 //((1/sqrt(2))*OC)=OA
    (3)
27
28 OC=sqrt(2)*OA; //from equation (3)
29 OB=((1/sqrt(2))*OC) //from equation(2)
30
31 disp(OC, 'magnitude(in m) of OC vector is ');
32 disp(OB, 'magnitude(in m) of OB vector is ');

```

Scilab code Exa 2.4 calculation of work done by the force during displacement

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1

```

```

3  clc;clear;
4  //example 2.4
5  //calculation of work done by the force during
    displacement
6
7  //given data
8  F=12; //force(in newton) acting on the particle
9  r=2; //displacement(in m) of the particle
10 theta=180; //angle(in degree) between force and
    displacement
11
12 //calculation
13 W=F*r*cosd(theta); //formula of work done
14
15 disp(W, 'work done(in J) by the force ,during the
    given displacement is ');

```

Scilab code Exa 2.4w calculation of direction of resultant vector

```

1  //developed in windows XP operating system 32bit
2  //platform Scilab 5.4.1
3  clc;clear;
4  //example 2.4w
5  //calculation of direction of resultant vector
6
7  //given data
8  //OA=OB=OC=F all the three vectors have same
    magnitude
9  //xcompOA=F*cos30=(F*(sqrt(3)))/2
10 //xcompOB=F*cos360=F/2
11 //xcompOC=F*cos135=-F/(sqrt(2))
12 //xcompr=xcompOA + xcompOB + xcompOC
13
14 //ycompOA=F*cos60=F/2
15 //ycompOB=F*cos360=-(F*(sqrt(3)))/2

```

```

16 //ycompOC=F*cos135=F/(sqrt(2))
17 //ycompr=ycompOA + ycompOB + ycompOC
18
19 // calculation
20 theta=atand((1-sqrt(3)-sqrt(2))/(1+sqrt(3)+sqrt(2)))
    ;
21
22 disp(theta,'the angle(in degree) made by OA+OB-OC
    vector with X axis is');

```

Scilab code Exa 2.5 calculation of angle between two vectors from known value of t

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.5
5 //calculation of angle between two vectors from
    known value of their cross product
6
7 //given data
8 C=15; //magnitude(in unit) of cross product of two
    vectors ,C=|A*B|
9 A=5; //magnitude(in unit) of A vector
10 B=6; //magnitude(in unit) of B vector
11 //calculation
12 theta=asind(C/(A*B)); //formula for cross product
13
14 printf("angle(in degree) between the given two
    vectors is %d or %d",theta,180-theta);

```

Scilab code Exa 2.6 calculation of the slope of curve at a given point

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.6
5 //calculation of the slope of curve at a given point
6
7 //given data
8 AB=5; //length of AB line segment
9 BC=4; //length of BC line segment
10 DE=5; //length of DE line segment
11 EF=-4; //length of EF line segment
12
13 //calculation
14 m1=AB/BC; //formula of slope ,m1=dy/dx at x=2
15 //m2=0 since tangent to curve at x=6 is parallel to
    x axis
16 m2=0;
17 m3=DE/EF; //formula of slope ,m2=dy/dx at x= 10
18
19 disp(m1,'the slope of the curve at x=2 is ');
20 disp(m2,'the slope of the curve at x=6 is ');
21 disp(m3,'the slope of the curve at x=10 is ');

```

Scilab code Exa 2.6w calculation of angle

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.6w
5 //calculation of angle
6
7 //given data
8 xcompOA=4; //magnitude(in m) of x component of OA
    vector
9 //xcompOB=6*cos(theta) magnitude(in m) of x
    component of OB vector

```

```

10
11 //calculation
12 theta=acosd(-xcompOA/6); //since xcompOA + xcompOB=0
      where xcompOB=6*cos(theta)
13
14 disp(theta, 'the value of angleAOB(in degree) is ');

```

Scilab code Exa 2.7w calculation of unit vector

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.7w
5 //calculation of unit vector
6
7 //given data
8 ax=5; //x component of A vector
9 ay=1; //y component of A vector
10 az=-2; //z component of A vector
11
12 //calculation
13 A=sqrt((ax*ax)+(ay*ay)+(az*az));
14 uax=ax/A; //x component of unit vector of A vector
15 uay=ay/A; //y component of unit vector of A vector
16 uaz=az/A; //z component of unit vector of A vector
17
18 disp(uax, 'x component of unit vector of A vector');
19 disp(uay, 'y component of unit vector of A vector');
20 disp(uaz, 'z component of unit vector of A vector');

```

Scilab code Exa 2.9 evaluation of a integral

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.9
5 //evaluation of a integral
6
7 //given data
8 //function of  $x=(2*x^2)+(3*x)+5$ 
9 //limit=3 to 6
10
11 //calculation
12 y=integrate('((2*x^2)+(3*x)+5)', 'x', 3, 6)
13
14 disp(y, 'value of the given integral is ')

```

Scilab code Exa 2.9w calculation of angle between two vectors

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.9w
5 //calculation of angle between two vectors
6
7 //given data
8 ax=2; //xcomponent of A vector
9 ay=3; //ycomponent of A vector
10 az=4; //zcomponent of A vector
11
12 bx=4; //xcomponent of B vector
13 by=3; //ycomponent of B vector
14 bz=2; //zcomponent of B vector
15
16 //calculation
17 adotb=(ax*bx)+(ay*by)+(az*bz);
18 a=sqrt((ax*ax)+(ay*ay)+(az*az));
19 b=sqrt((bx*bx)+(by*by)+(bz*bz));

```



```

20 theta=acosd(adotb/(a*b)); //formula of dot product
21
22 disp(theta,' angle(in degree) between given two
    vectors is ')

```

Scilab code Exa 2.10 calculation of round off value upto three digits

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.10
5 //calculation of round off value upto three digits.
6
7 //given data
8 a=15462
9 b=14.745
10 c=14.750
11 d=14.650*10^12
12
13 //calculation
14 //since round off upto three digit is required, we
    have to sort the numerics with the number of
    significant figures i.e. 3
15 na=15500
16 nb=14.7
17 nc=14.8
18 nd=14.6*10^12
19
20 printf('the value of %f rounded upto three
    significant digits is %d',a,na);
21 printf('\nthe value of %f rounded upto three
    significant digits is %3.2f',b,nb);
22 printf('\nthe value of %f rounded upto three
    significant digits is %3.2f',c,nc);
23 printf('\nthe value of %3.4e rounded upto three

```

```
significant digits is %3.2e',d,nd);
```

Scilab code Exa 2.10w calculation of value of the given scalar

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.10w
5 //calculation of value of the given scalar
6
7 //given data
8 ax=2; //xcomponent of A vector
9 ay=-3; //ycomponent of A vector
10 az=7; //zcomponent of A vector
11
12 bx=1; //xcomponent of B vector
13 by=0; //ycomponent of B vector
14 bz=2; //zcomponent of B vector
15
16 cx=1; //xcomponent of C vector
17 cy=0; //ycomponent of C vector
18 cz=2; //zcomponent of C vector
19
20 //calculation
21 //D=B*C
22 dx=(by*cz)-(cy*bz);
23 dy=-((bx*cz)-(cx*bz));
24 dz=(bx*cy)-(cx*by);
25
26 //R=A.(B*C)
27 R=(ax*dx)+(ay*dy)+(az*dz);
28
29 disp(R,'value of the given scalar is');
```

Scilab code Exa 2.11 calculation of value

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.11
5 //calculation of value
6
7 //given data
8 x=25.2;
9 y=1374;
10 z=33.3;
11
12 //calculation
13 temp=(x*y)/z
14 //since x,z has three significant figures and y has
    four significant figures
15 //we have to sort the answer with the minimum number
    of significant figures i.e. 3
16 //results into temp=1039.7838 we need to consider
    only 3 significant figures , hence
17
18 ntemp=1040
19
20 printf('value is %f,considering only 2 significant
    figures value is %d',temp,ntemp);
```

Scilab code Exa 2.11w calculation of change in volume of sphere as radius is chang

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
```

```

4 //example 2.11w
5 //calculation of change in volume of sphere as
   radius is changed
6
7 //given data
8 R=20; //initial radius(in cm) of sphere
9 Rdash=20.1; //final radius(in cm) of sphere
10 function v=f(R)
11     v=(4*%pi*R^3)/3;
12 endfunction
13
14 //calculation
15 function v=f(R)
16     v=(4*%pi*R^3)/3;
17 endfunction
18
19 deltaR=Rdash-R;
20 deltav=(derivative(f,R))*deltaR
21
22 disp(deltav,'the change in volume(in cm cube) of
   sphere is ')

```

Scilab code Exa 2.12 calculation of value

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.18w
5 //calculation of value
6
7 //given data
8 x=24.36;
9 y=0.0623;
10 z=256.2;
11

```

```

12 //calculation
13
14 //since after point the value of z is in one digit ,
    thus consider only one digit after point.
15 //the other values can be thus written as
16 x=24.4;
17 y=.1;
18 z=256.2;
19
20 temp=x+y+z
21 printf('the value is %3.1f',temp);

```

Scilab code Exa 2.13 calculation of average focal length of concave mirror consider

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.13
5 //calculation of average focal length of concave
    mirror considering uncertainty
6
7 //given data
8 fi=[25.4 25.2 25.6 25.1 25.3 25.2 25.5 25.4 25.3
    25.7]; //focal length(in cm)
9 N=length(fi);
10
11 //calculation
12 fbar=mean(fi) //average of fi
13 fnew=fi-fbar;
14 sfnew=sum(fnew.*fnew)
15 sigma=sqrt(sfnew/N) //uncertainty(in cm) in focal
    length
16
17 printf("the focal length of the given concave mirror
    (in cm) is %f or %f",fbar+sigma,fbar-sigma);

```

Scilab code Exa 2.13w calculation of maximum and minimum value of a given function

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.13w
5 //calculation of maximum and minimum value of a
   given function
6
7 //given data
8 function y=f(x)
9     y=x+(1/x);
10 endfunction
11
12 //calculation
13 //dy/dx=1-(1/x^2)=0 for maximum or minimum
14 //x=1 or -1
15 //at x=0 y=infinite is maximum value
16 //minimum value of y at x=1
17 ymin=f(1);
18
19 disp(ymin,'maximum value of given function is
   infinite and minimum value is ')
```

Scilab code Exa 2.14w calculation of the area under curve

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.14w
5 //calculation of the area under curve
```

```

6
7 //given data
8 function y=f(x)
9     y=x*x;
10 endfunction
11
12 //calculation
13 A=integrate('f','x',0,6)
14
15 disp(A,'the area under curve is ')

```

Scilab code Exa 2.18w calculation of value

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 2.18w
5 //calculation of value
6
7 //given data
8 x=21.6003;
9 y=234;
10 z=2732.10;
11 a=13;
12
13 //calculation
14 //since a has least significant figures that is 2,
15     we have to sort the other numerics with the same
16     number of significant figures i.e. 2
17
18 x=22;
19 y=234;
20 z=2732;
21 a=13;
22 temp=(x+y+z)*13
23 //results into temp=38844. Again we need to consider

```

```
    only 2 significant figures , hence
21 ntemp=39000
22
23 printf('value is %d,considering only 2 significant
    figures value is %d',temp,ntemp);
```

Chapter 3

Rest and Motion Kinematics

Scilab code Exa 3.1 calculation of distance and displacement

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.1
5 //calculation of distance and displacement
6
7 //given data
8 r=40; //radius(in m) of the circle
9
10 //calculation
11 dist=%pi*r; //distance travelled(in m)
12 displ=2*r; //displacement(in m)
13
14 disp(dist,'distance travelled(in m) by the person is
    ');
15 disp(displ,'displacement(in m) of the person from
    initial to final point is');
```

Scilab code Exa 3.1w calculation of average speed of the walk

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.1w
5 //calculation of average speed of the walk
6
7 //given data
8 v1=6//speed(in km/h) of the man
9 v2=8//speed(in km/h) of the man
10 d1=1//distance(in km) travelled at v1 speed
11 d2=1//distance(in km) travelled at v2 speed
12 d=2//given distance(in km)
13
14 //calculation
15 t=(v1/d1)+(v2/d2);//total time(in s) taken
16 vavg=d/t;//formula for average velocity
17
18 disp(vavg,'the average velocity(in km/h) of the man
    is ');

```

Scilab code Exa 3.2 calculation of average speed and instantaneous speed

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.2
5 //calculation of average speed and instantaneous
    speed
6
7 //given data
8 function s=f(t)
9     s=2.5*t^2;
10 endfunction
11 t=5; //time (in s)
12

```

```

13 //calculation
14 vav=f(t)/t; //average speed(in m/s)
15 vinst=derivative(f,t); //instantaneous speed(in m/s)
16
17 disp(vav,'the average speed(in m/s) of the particle
    is ');
18 disp(vinst,'the instantaneous speed(in m/s) of the
    particle is ');

```

Scilab code Exa 3.2w calculation of average speed and average velocity

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.2w
5 //calculation of average speed and average velocity
6
7 //given data
8 w=40//length(in ft)of the wall
9 t=50//time(in min) taken
10 rnd=10//number of rounds taken
11
12 //calculation
13 dist=2*w*rnd;
14 avgspeed=dist/t;
15 avgvelocity=0//average velocity(in ft/min) since
    displacement=0 as he is at the same door from
    where he has started
16
17 printf('the average speed of the teacher is %3.2f ft
    /min and the average velocity is %3.2f ft/min',
    avgspeed,avgvelocity);

```

Scilab code Exa 3.3 calculation of distance from speed versus time graph

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.3
5 //calculation of distance from speed versus time
  graph
6
7 //given data
8 base=3; //time(in s) representing the base of graph(
  triangle)
9 height=6; //speed(in m/s) representing the height of
  the graph(triangle)
10 //calculation
11 dist=(1/2)*base*height; //distance travelled is the
  area of the graph(triangle)
12
13 disp(dist,'the distance(in m) travelled by the
  particle is');
```

Scilab code Exa 3.3w calculation of average velocity and average acceleration

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.3w
5 //calculation of average velocity and average
  acceleration
6
7 //given data
8 A=1//given value of constant A
9 B=4//given value of constant B
10 C=-2//given value of constant C
11 D=5//given value of constant D
```

```

12 t=4//time(in s)
13 t1=0//initial time(in s) for calculation of average
    velocity and average acceleration
14 t2=4//final time(in s) for calculation of average
    velocity and average acceleration
15
16 function x=f(t)
17     x=(A*(t^3))+(B*(t^2))+(C*t)+D
18 endfunction
19
20 function a=f1(t)
21     a=(6*A*t)+(2*B)
22 endfunction
23
24 //calculation
25 v=derivative(f,t)//formula of velocity
26 na=f1(t)//formula of acceleration
27
28 x1=f(t1);//formula of position of the particle at t1
    time
29 x2=f(t2);//formula of position of the particle at t2
    time
30 vavg=(x2-x1)/(t2-t1);//formula of average velocity
31
32 v1=derivative(f,t1);//formula of velocity of the
    particle at t1 time
33 v2=derivative(f,t2);//formula of velocity of the
    particle at t2 time
34 aavg=(v2-v1)/(t2-t1);//formula of average
    acceleration
35
36 printf('\nthe velocity of particle at t=4 s is %3.2f
    m/s',v);
37 printf('\nthe acceleration of particle at t=4 s is
    %3.2f m/s^2',na)
38 printf('\nthe average velocity of the particle
    between t=0 s and t=4 s is %3.2f m/s',vavg);
39 printf('\nthe average acceleration of the particle

```

between $t=0$ s and $t=4$ s is $\%3.2f$ m/s²', aavg);

Scilab code Exa 3.4 calculation of average velocity of the tip of minute hand in a

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 3.4
5 //calculation of average velocity of the tip of
   minute hand in a table clock
6
7 //given data
8 R=4; //length(in cm) of the minute hand = radius(in
   cm) of the circle representing the clock
9 t1=1800; //time(in second) elapsed between 6.00 a.m
   and 6.30 a.m          30*60
10 t2=45000; //time(in second) elapsed between 6.00 a.m
   and 6.30 p.m          (12*60*60) + (30*60)
11
12 //calculation
13 vav1=(2*R)/t1; //average velocity(in cm/s) in first
   case
14 vav2=(2*R)/t2; //average velocity(in cm/s) in second
   case
15
16 disp(vav1, 'average velocity(in cm/s) of the tip of
   minute hand in time elapsed between 6.00 a.m and
   6.30 a.m is ');
17 disp(vav2, 'average velocity(in cm/s) of the tip of
   minute hand in time elapsed between 6.00 a.m and
   6.30 p.m is ');
```

Scilab code Exa 3.4w calculation of distance travelled displacement and accelerat.

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.4w
5 //calculation of distance travelled ,displacement and
   acceleration
6
7 //given data
8 //graph of velocity(in m/s) versus time(in s)
9
10 //calculation
11 d1=(2*10)/2;//distance(in m) travelled during t=0 s
   to t=2 s = area of OAB
12 d2=(2*10)/2;//distance(in m) travelled during t=2 s
   to t=4 s = area of BCD
13 d=d1+d2;//distance(in m) travelled during t=0 s to t
   =4 s
14 dis=d1+(-d2);//displacement(in m) during t=0 s to t
   =4 s
15 a1=(10-0)/(1-0);//acceleration(in m/s^2) at t=1/2 s
   = slope of OA
16 a2=(-10-0)/(3-2);//acceleration(in m/s^2) at t=2 s =
   slope of BC
17
18 disp(d1,'distance(in m) travelled during t=0 s to t
   =2 s is ');
19 disp(d2,'distance(in m) travelled during t=2 s to t
   =4 s is ');
20 disp(d,'distance(in m) travelled during t=0 s to t=4
   s ');
21 disp(dis,'displacement(in m) during t=0 s to t=4 s')
   ;
22 disp(a1,'acceleration(in m/s^2) at t=1/2 s ');
23 disp(a2,'acceleration(in m/s^2) at t=2 s ');

```

Scilab code Exa 3.5 calculation of distance travelled in given time time taken to

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 3.5
5 //calculation of distance travelled in given time,
   time taken to reach a particular velocity and
   distance covered to reach particular velocity
6
7 //given data
8 u=2.5; //initial velocity(in m/s) of the particle
9 t=2; //time(in s) for which the particle has
   travelled
10 v=7.5; //final velocity(in m/s) of the particle
11 a=.5; //acceleration(in m/s^2) of the particle
12
13 //calculation
14 x=(u*t)+((1/2)*a*t*t); //Equation of motion with
   constant acceleration
15 t1=(v-u)/a; //Equation of motion with constant
   acceleration
16 x1=((v*v)-(u*u))/(2*a); //Equation of motion with
   constant acceleration
17
18 disp(x, 'distance(in m) travelled by the particle in
   the first two seconds is ');
19 disp(t1, 'time(in s) taken by particle to reach 7.5 m
   /s velocity is ');
20 disp(x1, 'distance(in m) covered by particle to reach
   7.5 m/s velocity is ');
```

Scilab code Exa 3.5w calculation of acceleration and distance travelled

```
1 //developed in windows XP operating system 32bit
```



```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.5w
5 //calculation of acceleration and distance travelled
6
7 //given data
8 v1=100//speed1(in m/s)
9 v2=150//speed2(in m/s)
10 t=1//change in time (in s)
11
12 //calculation
13 a=(v2-v1)/t;//formula of acceleration
14 x=((v2*v2)-(v1*v1))/(2*a);//distance travelled in (t
    +1)th second
15
16 printf('acceleration of the particle is %3.2f m/s^2'
    ,a)
17 printf('\\ndistance travelled in (t+1)th second is %3
    .2f m',x)

```

Scilab code Exa 3.6 calculation of displacement of particle in last 1 second

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.6
5 //calculation of displacement of particle in last 1
    second
6
7 //given data
8 u=5; //initial velocity(in m/s) of the particle
9 a=2; //constant acceleration(in m/s^2) of the
    particle
10 t=10; //time(in s)
11

```

```

12 // calculation
13 //s = u*t+((1/2)*a*t^2)....equation of motion
14 //sdash = u*(t-1)+((1/2)*a*(t-1)^2)
15 //st = s-sdash =u+((a/2)*(2*t-1));
16
17 st=u+((a/2)*(2*t-1)); //formula of displacement in
    last one second
18 disp(st,'displacement(in m) of particle in last 1
    second ');

```

Scilab code Exa 3.6w calculation of acceleration

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.6w
5 //calculation of acceleration
6
7 //given data
8 u=0//initial velocity(in m/s)
9 v=2.2//final velocity(in m/s)
10 d=.24//distance(in m) travelled
11
12 //calculation
13 a=((v*v)-(u*u))/(2*d);//formula of acceleration
14
15 printf('the acceleration of the stone is %3.3f m/s^2
    ',a)

```

Scilab code Exa 3.7 calculation of maximum height reached by the ball

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1

```

```

3  clc;clear;
4  //example 3.7
5  //calculation of maximum height reached by the ball
6
7  //given data
8  u=4; //initial velocity(in m/s) of the ball
9  a=-10; //acceleration(in m/s^2) of the ball
10
11 //calculation
12 y=-((u*u)/(2*a)); //formula for vertical height(in m
    )
13
14 disp(y, 'maximum height(in m) reached by the ball is '
    );

```

Scilab code Exa 3.8 calculation of velocity and position of the particle

```

1  //developed in windows XP operating system 32bit
2  //platform Scilab 5.4.1
3  clc;clear;
4  //example 3.8
5  //calculation of velocity and position of the
    particle
6
7  //given data
8  a=1.5; //acceleration(in m/s^2) of the particle
9  theta=37; //angle(in degree) made by particle with X
    axis
10 ux=8; //x component of initial velocity(in m/s) of
    the particle
11 uy=0; //y component of initial velocity(in m/s) of
    the particle
12 t=4; //time(in s)
13
14 //calculation

```

```

15 ax=a*cosd(theta);
16 ay=a*sind(theta);
17
18 vx=ux+(ax*t); //formula of x component of final
    velocity
19 vy=uy+(ay*t); //formula of y component of final
    velocity
20 v=sqrt((vx*vx)+(vy*vy));
21 thetav=atand(vy/vx);
22
23 x=(ux*t)+((ax*t*t)/2); //formula for x coordinate of
    particle at time t
24 y=(uy*t)+((ay*t*t)/2); //formula for y coordinate of
    particle at time t
25
26 printf('the velocity of the particle at t=4 s is %f
    m/s and angle made with X axis is %f degree',v,
    thetav)
27 printf('the particle is at(%f,%f)m at time t=4 s',x
    ,y)

```

Scilab code Exa 3.8w calculation of total distance and number of trips

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.8w
5 //calculation of total distance and number of trips
6
7 //given data
8 dcar=20//distance(in km) travelled by the car
9 vcar=40//speed(in km/h) of the car
10 vfly=100//speed(in km/h) of the fly
11
12 //calculation

```

```

13 tcar=dcar/vcar;//time(in h) taken by the car to
    cover given distance
14 tfly=tcar;
15 dfly=tfly*vfly;//distance(in m) travelled by the fly
16 //number of trips made by fly can be infinite
17
18 printf('total distance travelled by the fly is %3.2f
    km and number of trips made by fly can be
    infinite ',dfly);

```

Scilab code Exa 3.9 calculation of horizontal range of the projectile

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.9
5 //calculation of horizontal range of the projectile
6
7 //given data
8 u=12//initial velocity(in m/s) of the projectile
9 theta=45//angle(in degree) made by the projectile
    with X axis
10 g=10//gravitational acceleration(in m/s^2)
11
12 //calculation
13 h=(u*u*sind(2*theta))/g;//formula for horizontal
    range of a projectile
14
15 printf('the ball hits the field at %f m from the
    point of projection ',h);

```

Scilab code Exa 3.9w drawing graph of x versus t v versus t and a versus t

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.9w
5 //drawing graph of x versus t, v versus t and a
   versus t
6
7 //given data
8 h=19.6//height(in m) from where the ball is dropped
9 //evaluating value for equation  $x=(u*t)+((1/2)*a*$ 
    $t^2)$ 
10
11 //calculation
12 t=[0 1 2 2 3 4]
13 x=[0 4.9 19.6 19.6 4.9 0]//values of x(in m)
   obtained on evaluating equation  $x=(u*t)+((1/2)$ 
    $*a*t^2)$  along with direction of motion
14 v=[0 9.8 19.6 -19.6 -9.8 0]//values of v(in m)
   obtained on evaluating equation  $v=u+(a*t)$ 
   along with direction of motion
15 a=9.8//constant acceleration(m/s2)
16
17 subplot(221);
18 plot(t,x);
19 xlabel('time(in s)');
20 ylabel('distance(in m)');
21
22 subplot(222);
23 plot(t,v);
24 xlabel('time(in s)');
25 ylabel('velocity(in m/s)');
26
27 subplot(223);
28 plot(t,a*ones(1,length(t)));
29 xlabel('time(in s)');
30 ylabel('acceleration (in m/s2)');

```

Scilab code Exa 3.10 calculation of velocity of the swimmer with respect to ground

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.10
5 //calculation of velocity of the swimmer with
   respect to ground
6
7 //given data
8 vsr=4//velocity(in km/h) of the swimmer with respect
   to water
9 vrg=3//velocity(in km/h) of the river water with
   respect to ground
10
11 //calculation
12 vsg=sqrt((vsr*vsr)+(vrg*vrg));//formula for relative
   velocity      vsg = vsr + vrg
13 theta=atand(4/3);
14
15 printf('the velocity of the swimmer with respect to
   ground is %f km/h and angle made by him with X
   axis is %f degree',vsg,theta);
```

Scilab code Exa 3.10w calculation of height of balloon when stone reaches ground

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.10w
5 //calculation of height of balloon when stone
   reaches ground
```

```

6
7 //given data
8 x=-50//height(in m) of the ballon when the stone was
      dropped
9 u=5//velocity(in m/s) of the ballon
10 a=-10//acceleration(in m/s^2) of the ballon
11
12 //calculation
13 //from  $x=(u*t)+((1/2)*a*t*t)$  we have  $-5*t^2 + 5*t +$ 
      50 = 0
14 a=-5//coefficient of  $t^2$ 
15 b=5//coefficient of  $t$ 
16 c=50//constant
17
18 t1=(-b+sqrt((b*b)-(4*a*c)))/(2*a)//value of t
19 t2=(-b-sqrt((b*b)-(4*a*c)))/(2*a)//value of t
20
21 if(t1>0)
22     t=t1;
23 end
24
25 if(t2>0)
26     t=t2;
27 end
28
29 if(t1>0 & t2>0)
30     tn1=t1;
31     tn2=t2;
32 end
33
34 tballoon=t;//during this time baloon has uniformly
      moved upwards
35 dballoon=u*t;
36 dtotal=dballoon+(-x);
37
38 printf('height of the ballon when the stone reaches
      ground is %3.2f m',dtotal);

```

Scilab code Exa 3.11 calculation of velocity of the raindrops with respect to the

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.11
5 //calculation of velocity of the raindrops with
   respect to the man
6
7 //given data
8 vmanstreet=3//velocity(in km/h) of man with respect
   to the street
9 vrainstreet=4//velocity(in km/h) of rain with
   respect to the street
10
11 //calculation
12 vrainman=sqrt((vrainstreet*vrainstreet)+(vmanstreet*
   vmanstreet));//velocity(in km/h) of rain with
   respect to the man
13 theta=atand(vmanstreet/vrainstreet);//angle(in
   degree) made by rain drops with Y axis
14
15 printf('velocity of the raindrops with respect to
   the man is %3.2f km/h and angle made by rain
   drops with Y axis is %3.3f degree',vrainman,theta
   )
```

Scilab code Exa 3.11w calculation of time of flight horizontal range and vertical

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
```

```

4 //example 3.11w
5 //calculation of time of flight ,horizontal range and
   vertical range
6
7 //given data
8 u=20//initial velocity(in m/s) of the football
9 theta=45//angle(in degree) made by the football with
   ground
10 g=10//gravitational acceleration(in m/s^2)
11
12 //calculation
13 ux=u*cosd(theta);
14 uy=u*sind(theta);
15
16 t=(2*uy)/g;// from equation y=(uy*t)+((1/2)*g*t*t)
   ..... taking y=0
17 H=((uy*uy)/(2*g));//from equation (vy*vy)=(uy*uy)
   -(2*g*y) taking vy=0
18 x=ux*t;//horizontal distance travelled at ux
   velocity
19
20 printf('the time taken by the ball to strike the
   ground is %3.2f s',t);
21 printf('\nthe maximum height reached by the ball is
   %3.2f m',H);
22 printf('\nthe horizontal distance travelled by the
   ball before reaching the ground is %3.2f m',x);

```

Scilab code Exa 3.16w calculation of angle of the swim and time to cross the river

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.16w
5 //calculation of angle of the swim and time to cross

```

```

        the river
6
7 //given data
8 vrg=2//velocity(in km/h) of the river with respect
    to ground
9 vmr=3////velocity(in km/h) of the man with respect
    to river
10 d=.5//width(in km) of the river
11
12 //calculation
13 theta=asind(vrg/vmr);//from equation of relative
    velocity   vmg=vmr+vrg...taking components along
    X axis
14 vmg=vmr*cosd(theta);//taking component along Y axis
15 time=d/vmg;
16
17 printf('swimmer should try to swim,making an angle
    of %3.2f degree with Y axis ',theta);
18 printf('\ntime taken by the swimmer to cross the
    river is %3.2f h',time);

```

Scilab code Exa 3.17w calculation of time taken and position of the arrival on opposite bank

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.17w
5 //calculation of time taken and position of the
    arrival on opposite bank
6
7 //given data
8 dyaxis=.5//displacement(in km) along Y axis
9 vrg=2//velocity(in km/h) of the river with respect
    to ground
10 vmr=3////velocity(in km/h) of the man with respect

```

```

    to river
11 theta1=30//angle(in degree) of vmr with Y axis
12 theta2=90//angle(in degree) of vrg with Y axis
13
14 //calculation
15 vyaxis=(vmr*cosd(theta1))+(vrg*cosd(theta2));//
    velocity along Y axis i.e taking y component in
    equation    vmg=vmr+vrg
16 t=dyaxis/vyaxis;
17 vxaxis=(-vmr*sind(theta1))+(vrg*sind(theta2));//
    velocity along X axis i.e taking x component in
    equation    vmg=vmr+vrg
18 dxaxis=vxaxis*t;
19
20 printf('time taken by the swimmer to cross the river
    is %3.2f hour',t);
21 printf('\ndisplacement of the swimmer along X axis
    is %3.4f km',dxaxis);

```

Scilab code Exa 3.18w calculation of speed of raindrops with respect to road and

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.18w
5 //calculation of speed of raindrops with respect to
    road and the moving man
6
7 //given data
8 vmg=10//velocity(in km/h) of the man with respect to
    the ground
9 theta=30//angle(in degree) made by vrg with Y axis
10
11 //calculation
12 vrg=vmg/sind(theta);// from equation of relative

```

```

    velocity    vrg=vrm+vmg...taking horizontal
    components
13 vrm=vrg*cosd(theta); // from equation of relative
    velocity    vrg=vrm+vmg...taking vertical
    components
14
15 printf('the speed of raindrops with respect to the
    ground is %3.2f km/h and with respect to the man
    is %3.2f km/h',vrg,vrm);

```

Scilab code Exa 3.19w calculation of speed and direction of rain with respect to

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 3.19w
5 //calculation of speed and direction of rain with
    respect to the road
6
7 //given data
8 vmanroad=8//velocity(in km/h) of the man with
    respect to the road
9
10 //calculation
11 //from equation of relative velocity    vrainroad =
    vrainman + vmanroad
12 //taking horizontal components    vrainroad*sind
    (alpha)=8    1
13 //taking components along line OA    vrainroad*sind
    (30+alpha)=12*cosd(30)    2
14 //from    1    and    2
15
16 alpha=acotd(sqrt(3)/2);
17 vrainroad=vmanroad/sind(alpha); //from equation 2
18

```

```
19 printf('the speed of the rain with respect to the  
road is %3.2f km/h and makes angle of %3.2f  
degree with Y axis ',vrainroad,alpha);
```

Chapter 4

The Forces

Scilab code Exa 4.1 calculation of coulomb force

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 4.1
5 //calculation of coulomb force
6
7 //given data
8 np=26//number of protops in an iron atom
9 na=6*10^26//number of atome in 58 kg iron
10 mi=58//mass(in kg) of iron
11 e=1.6*10^(-19)//charge(in coulomb) on an electron
12 perdiff=1//percentage of charge of electron less
    than that of proton
13 r=1//separation(in m) between the two blocks
14
15 //calculation
16 poschrg=(na*np*e*perdiff)/(mi*100)
17 fc=(9*10^9*poschrg*poschrg)/(r*r)
18
19 disp(fc,'the coulomb force(in newton) between the
    two blocks is newton')
```

Scilab code Exa 4.3w calculation of ratio of the electrical force to the gravitat.

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 4.3w
5 //calculation of ratio of the electrical force to
   the gravitational force between two electrons
6
7 //given data
8 me=9.1*10^-31//mass(in kg)of an electron
9 e=1.6*10^-19//charge(in coulomb)of an electron
10 k=9*10^9//value of ratio 1/(4*%pi*epsilonzero) (in
    N m^2/C^2)
11 G=6.67*10^-11//value of universal gravitational
    constant (in N m^2/kg^2)
12
13 //calculation
14 ratio=(k*e*e)/(G*me*me)//ratio = electric force /
    gravitational force
15
16 disp(ratio,'the ratio of electric to gravitational
    force between two electrons is')
```

Chapter 5

Newton s Laws of Motion

Scilab code Exa 5.1 calculation of force exerted by the string on a particle

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3   clc; clear;
4 //example 5.1
5 //calculation of force exerted by the string on a
   particle
6
7 //given data
8 m=.5//mass(in kg) of the particle
9 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
10
11 //calculation
12 T=m*g//tension in the string is equal to the
   downward force exerted by earth
13
14 printf('the force exerted by the string on particle
   in vertically upward direction is %3.2f N',T);
```

Scilab code Exa 5.3w calculation of the force exerted by the tree limb on the bullet

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 5.3w
5 //calculation of the force exerted by the tree limb
   on the bullet
6
7 //given data
8 u=250//initial velocity(in m/s) of the bullet
9 v=0//final velocity(in m/s) of the bullet
10 x=.05//penetration(in m) by the bullet in the tree
   limb
11 m=.01//mass of bullet(in kg)
12
13 //calculation
14 a=((u*u)-(v*v))/(2*x)//formula of horizontal
   acceleration in case of uniform linear motion
15 F=m*a;
16
17 printf('the force exerted by the tree limb on the
   bullet is %3.2f N',F)
```

Scilab code Exa 5.4w calculation of the position of a particle

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 5.4w
5 //calculation of the position of a particle
6
7 //given data
8 m=.01//mass(in kg) of the particle
9 Fx=10//component of force(in N) along X axis
```

```

10 Fy=5//component of force(in N) along Y axis
11 ux=0//x component of initial velocity(in m/s) of the
    particle
12 uy=0//y component initial velocity(in m/s) of the
    paticle
13 t=5//time(in s) at which position is to be
    determined
14
15 //calculation
16 ax=Fx/m;
17 x=(ux*t)+((1/2)*ax*t*t);//formula of horizontal
    position in case of uniform linear motion
18 ay=Fy/m;
19 y=(uy*t)+((1/2)*ay*t*t);//formula of vertical
    position in case of uniform linear motion
20
21 printf('at t=5 s position of the particle is (i%3.2f
    + j%3.2f)m',x,y)

```

Scilab code Exa 5.7w calculation of acceleration with which ring starts moving if

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 5.7w
5 //calculation of acceleration with which ring starts
    moving if released from rest at an angle theta
6
7 //given data
8 //m=mass of the ring
9 theta=30//angle(in degree)of the release
10 m=1//assume for obtaining the solution
11 M=2*m //mass of the block
12 g=9.8//gravitational acceleration(in m/s^2) of the
    earth

```

```

13
14 //calculation
15 //M*g-T=M*a*cosd(theta)..... equation of motion of
    the block...(1)
16 //T*cosd(theta)=m*a..... equation of motion of
    the ring....(2)
17 //solving above equations we get
18 a=(M*g*cosd(theta))/(m+M*(cosd(theta)*cosd(theta)))
19
20 printf('the acceleration with which ring starts
    moving if released from rest at an angle theta is
    %3.2f m/s^2',a)

```

Scilab code Exa 5.8w calculation of the maximum acceleration of the man for safe climbing

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 5.8w
5 //calculation of the maximum acceleration of the man
    for safe climbing
6
7 //given data
8 m=60//mass(in kg) of the man
9 theta=30//angle(in degree) made by the rope with
    ground
10 fgmax=360//maximum force(in N) that can be applied
    to the wooden clamp
11 g=10//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 T=fgmax/sind(theta)//since T*sin(theta)=upward force
15 a=(T-(m*g))/m//from equation of motion
16

```

```
17 printf('the maximum acceleration of the man for safe  
climbing is %3.2f m/s^2',a)
```

Chapter 6

Friction

Scilab code Exa 6.1 calculation of the angle made by the contact force with the ve

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.1
5 //calculation of the angle made by the contact force
   with the vertical and the magnitude of contact
   force
6
7 //given data
8 M=.4//mass(in kg) of the body
9 f=3//frictional force(in N)
10 g=10//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation
13 N=M*g//formula of normal force
14 theta=atand(f/N)//angle made by the contact force
   with the vertical
15 F=sqrt((N*N)+(f*f))
16
17 printf('the angle made by the contact force with the
```

vertical is %3.2f degree \n the magnitude of
contact force is %3.2f N',theta,F)

Scilab code Exa 6.1w calculation of the maximum angle to prevent slipping

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.1w
5 //calculation of the maximum angle to prevent
   slipping
6
7 //given data
8 mus=.3//coefficient of static friction
9
10 //calculation
11 thetamax=atand(mus)
12
13 printf('the maximum angle to prevent slipping is %3
   .2f',thetamax)
```

Scilab code Exa 6.2 calculation of the force of friction exerted by the horizontal

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.2
5 //calculation of the force of friction exerted by
   the horizontal surface on the box
6
7 //given data
8 M=20//mass(in kg) of the box
9 muk=.25//coefficient of kinetic friction
```

```

10 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 fk=muk*M*g//formula of kinetic friction
14
15 printf('the force of friction exerted by the
    horizontal surface on the box,in opposite
    direction to the pull is %3.2f N',fk)

```

Scilab code Exa 6.2w calculation of frictional force and minimum value of coefficient

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.2w
5 //calculation of frictional force and minimum value
    of coefficient of static friction
6
7 //given data
8 m=4//mass(in kg) of the block
9 f=20//frictional force(in N)=horizontal force(in N)
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 N=m*g//normal force
14 musmin=f/N
15
16 printf('the frictional force on the block,in
    opposite direction to the applied force is %3.2f
    N',f)
17 printf('\nthe coefficient of static friction between
    the block and the table is greater than or equal
    to %3.2f',musmin)

```

Scilab code Exa 6.3 calculation of the force of friction exerted by the horse and

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.3
5 //calculation of the force of friction exerted by
   the horse and condition of boy for sliding back
6
7 //given data
8 M=30//mass(in kg) of the boy
9 a=2//average acceleration(in m/s^2) of the horse
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 fs=M*a//Newton's second law
14 musmax=fs/(M*g)//equation of static friction
15
16 printf('the force of friction exerted by the horse
    on the boy is %3.2f N',fs)
17 printf('\\nfor the boy sliding back during
    acceleration , the value of coefficient of static
    friction is less than %3.2f ',musmax)
```

Scilab code Exa 6.3w calculation of the maximum value of mass of the block

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.3w
```

```

5 //calculation of the maximum value of mass of the
   block
6
7 //given data
8 mus=.2//coefficient of static friction between the
   block and the table
9 M=2//mass(in kg) of one block
10 g=10//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation
13 N=M*g//normal force
14 //T=m*g          tension in the string
                       (1)
15 //fs=mus*N      frictional force
                       (2)
16 //f=T           from equilibrium equation of 2 kg
   block          (3)
17 //from above equations ,we get
18 m=(mus*N)/g
19
20 printf('the maximum value of mass of the block is %3
   .2f kg',m)

```

Scilab code Exa 6.4 calculation of coefficient of static friction and kinetic friction

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.4
5 //calculation of coefficient of static friction and
   kinetic friction between the block and the plank
6
7 //given data
8 theta1=18//angle of plank(in degree) with horizontal

```

```

    when block starts slipping
9  theta2=15//angle of plank(in degree) with horizontal
    when block slips with uniform speed
10
11 //calculation
12 mus=tand(theta1)//formula of coefficient of static
    friction
13 muk=tand(theta2)//formula of coefficient of kinetic
    friction
14
15 printf('the coefficient of static friction between
    the block and the plank is tan(%d)=%3.2f',theta1,
    mus)
16 printf('\n the coefficient of kinetic friction
    between the block and the plank is tan(%d)=%3.2f'
    ,theta2,muk)

```

Scilab code Exa 6.5w calculation of the coefficient of kinetic friction

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.5w
5 //calculation of the coefficient of kinetic friction
6
7 //given data
8 theta=30//angle(in degree)f the incline
9 g=10//gravitational acceleration(in m/s^2) of the
    earth
10
11 //calculation
12 a=g/4//acceleration(in m/s^2) of the block.....given
13 //f=m*g/4.....taking parallel components
    to the incline
14 //N=m*g*cosd(theta).....taking vertical components

```

```

        to the incline
15 //from above equations ,we get
16 muk=1/(4*cosd(theta))//      muk=f/N      equation of
        static friction
17
18 printf('the coefficient of kinetic friction is %3.2f
        ',muk)

```

Scilab code Exa 6.6w calculation of the values of coefficient of static and kinetic

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.6w
5 //calculation of the values of coefficient of static
        and kinetic friction
6
7 //given data
8 M=2.5//mass(in kg) of the block
9 F=15//horizontal force(in N)
10 g=10//gravitational acceleration(in m/s^2) of the
        earth
11 x=10//displacement(in m) of the block
12 t=5//time(in s) required by the block
13
14 //calculation
15 mus=F/(M*g)
16 a=(2*x)/(t*t)//acceleration of the block from
        equation of uniform linear motion
17 //F-muk*M*g=M*a..... newton's second law
18 muk=(F-(M*a))/(M*g)
19
20 printf('the coefficient of static friction between
        the block and the surface is %3.2f',mus)
21 printf('\n the coefficient of kinetic friction

```

between the block and the surface is $\%3.2f'$,muk)

Scilab code Exa 6.10w calculation of minimum and maximum values of mass and the a

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 6.10w
5 //calculation of minimum and maximum values of m(
    mass) and the acceleration if given a gentle push
6
7 //given data
8 mus=.28//the value of coefficient of static friction
    between the block and the surface
9 muk=.25//the value of coefficient of kinetic
    friction between the block and the surface
10 M=2//mass(in kg) of one block
11 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 //T=(M*g*(1-mus))/sqrt(2) ..... taking
    components along incline for block1 .....(1)
15 //T=(M*g*(1+mus))/sqrt(2) ..... taking
    components along incline for block2 .....(2)
16 //from above equations,we get
17 m1=((1-mus)*M)/(1+mus)//minimum value of m
    .....(3)
18 m2=((1+mus)*M)/(1-mus)//maximum value of m obtained
    by taking reverse direction of friction in above
    equations
19
20 //(M*g/sqrt(2)) - T = M*a ..... newton's second
    law for M block .....(4)
```

```

21 //T - (m*g/sqrt(2)) = m*a ..... newton 's second
    law for m block ..... (5)
22 //adding equations (4) and (5)
23 //((M*g*(1-muk))/sqrt(2)) - ((m*g*(1+muk))/sqrt(2))
    = (M+m)*a
24 a=(((M*(1-muk))-(m1*(1+muk)))*g)/(sqrt(2)*(M+m1))//
    calculating acceleration for minimum value of m
    if gently pushed ..... given
25
26 printf('the minimum value of m for which the system
    remains at rest is %3.2f kg',m1)
27 printf('\nthe maximum value of m for which the
    system remains at rest is %3.2f kg',m2)
28 printf('\nthe acceleration of either block for
    minimum value of m and if gently pushed up the
    incline is %3.2f m/s^2',a)

```

Chapter 7

Circular Motion

Scilab code Exa 7.1 calculation of the angular velocity

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.1
5 //calculation of the angular velocity
6
7 //given data
8 v=10//linear speed(in m/s)
9 r=20*10-2//radius(in cm) of the circle
10
11 //calculation
12 w=v/r//formula of angular velocity
13
14 printf('the angular velocity is %d rad/s',w)
```

Scilab code Exa 7.1w calculation of the maximum speed the car can take on the turn

```
1 //developed in windows XP operating system 32bit
```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.1w
5 //calculation of the maximum speed the car can take
   on the turn without skidding
6
7 //given data
8 R=45//radius(in m) of the turn
9 mus=2.0//coefficient of static friction between the
   tyre and the road
10 g=10//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation
13 //considering forces in vertical and horizontal
   directions an dplying Newton's law we get
14 //fs = M*v*v.....(1)
15 //by equation of limiting friction ,we get
16 //fs = mus*N = mus*M*g.....(2)
17 //from above equations we get
18 v=sqrt(mus*g*R)
19
20 printf('the maximum speed the car can take on the
   turn without skidding is %d m/s or %3.1f km/hr',v
   ,(v*10^-3*60*60))

```

Scilab code Exa 7.2 calculation of the angular acceleration

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.2
5 //calculation of the angular acceleration
6
7 //given data

```



```

8 v1=5//initial speed(in m/s)
9 v2=6//final speed(in m/s)
10 dt=2//change in time(in s)
11 r=20*10^-2//radius(in cm) of the circle
12
13 //calculation
14 at=(v2-v1)/dt//formula of tangential acceleration
15 alpha=at/r//formula of angular acceleration
16
17 printf('the angular accleration is %3.1f rad/s^2',
        alpha)

```

Scilab code Exa 7.2w calculation of the value of angle of banking

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.2w
5 //calculation of the value of angle of banking
6
7 //given data
8 r=600//radius(in m) of the track
9 v=180*10^3/(60*60)//speed(in m/s) of the car
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 //for vertical direction
14 //N*cosd(theta) = M*g.....(1)
15 //for horizontal direction
16 //N*sind(theta) = M*v*v/r.....(2)
17 //from above equations ,we get
18 theta=atand((v*v)/(r*g))
19
20 printf('the value of angle of banking is %3.2f

```

degree',theta)

Scilab code Exa 7.3 calculation of the magnitude of linear acceleration

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.3
5 //calculation of the magnitude of linear
   acceleration
6
7 //given data
8 r=10*10^-2//radius(in cm)
9 t=4//time(in s) taken
10
11 //calculation
12 d=2*%pi*r//distance covered
13 v=d/t//linear speed
14 a=(v*v)/r
15
16 printf('the linear acceleration is %3.2f m/s^2',a)
```

Scilab code Exa 7.4 calculation of the value of radial and tangential acceleration

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.4
5 //calculation of the value of radial and tangential
   acceleration
6
7 //given data
8 t=3//time(in s)
```

```

9 r=20*10^-2//radius(in cm) of the circle
10
11 function v1=f(t1)
12     v1=2*t1
13 endfunction
14
15 //calculation
16 v=f(t)
17 ar=(v*v)/r//radial acceleration
18 at=derivative(f,t)//tangential acceleration
19
20 printf('the value of radial acceleration is %d m/s^2
        ',ar)
21 printf('\nthe value of tangential acceleration is %d
        m/s^2',at)

```

Scilab code Exa 7.4w calculation of the value of elongation of the spring

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.4w
5 //calculation of the value of elongation of the
  spring
6
7 //given data
8 k=100//spring constant(N/m) of the given spring
9 l0=.5//natural length(in m) of the string
10 m=.5//mass(in kg) of the particle
11 w=2//angular velocity(in rad/s) of the mass
12
13 //calculation
14 //from the equation of horizontal force
15 //k*l = m*v*v/r = m*w*w*r = m*w*w*(l0+l)
    ..... (1)

```

```

16 //from above equation we get
17 l=(m*w*w*10)/(k-(m*w*w))
18
19 printf('the value of elongation of the spring is %3
    .2f m or %3.1f cm',l,l*100)

```

Scilab code Exa 7.5 calculation of the normal contact force by the side wall of the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.5
5 //calculation of the normal contact force by the
    side wall of the groove
6
7 //given data
8 r=25*10^-2//radius(in m) of the circlce
9 m=.1//mass(in kg) of the block
10 t=2//time(in s) taken by the block
11
12 //calculation
13 v=2*pi*r/t//speed of the block
14 a=(v*v)/r//acceleration of the block
15 N=m*a//newton's second law
16
17 printf('the normal contact force by the side wall of
    the groove is %3.2f N',N)

```

Scilab code Exa 7.6 calculation of the speed of vehicle on the turn

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 7.6
5 //calculation of the speed of vehicle on the turn
6
7 //given data
8 r=10//radius(in m) of the turn
9 theta=10//angle(in degree) of the bank
10 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 v=sqrt(r*g*tand(theta))//since tand(theta) = (v*v)/(
    r*g)
14
15 printf('for normal contact force providing the
    necessary centripetal force ,the speed of vehicle
    on the turn is %3.2f m/s',v)

```

Scilab code Exa 7.7 calculation of the weight of the body if spring balance is shifted

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.7
5 //calculation of the weight of the body if spring
    balance is shifted to the equator
6
7 //given data
8 W=98//weight(in N) of the body at north pole
9 R=6400*10^3//radius(in m) of the earth
10 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 m=W/g//formula of weight
14 w=(2*%pi)/(24*60*60)//angular speed of the earth

```

```

15 We=W-(m*w*w*R) // since We = W - (m*w*w*R)
16
17 printf('the weight of the body if spring balance is
    shifted to the equator is %3.2f N',We)

```

Scilab code Exa 7.7w calculation of the value of force exerted by the air on the p

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.7w
5 //calculation of the value of force exerted by the
    air on the plane
6
7 //given data
8 v=900*10^3/(60*60)//speed(in m/s) of the fighter
    plane
9 r=2000//radius(in m)of the vertical circle
10 M=16000//mass(in kg)
11 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 //from Newton's second law
15 //F-M*g = M*v*v/r
16 //from above equation ,we get
17 F=M*(g+(v*v/r))
18
19 printf(' the force exerted by the air ,on the plane
    in upward direction is %3.2e N',F)

```

Scilab code Exa 7.8w calculation of the angular speed of rotation

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.8w
5 //calculation of the angular speed of rotation
6
7 //given data
8 L=20*10^-2//length(in m) of the rod = length(in m)of
   the string
9 theta=30//angle(in degree) made by the string with
   the vertical
10 g=10//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation
13 //applying Newton's second law
14 //T*sind(theta) = m*w*w*L*(1+sind(theta))
   .....(1)
15 //applying Newton's first law in vertical direction
16 //T*cosd(theta) = m*g
   .....(2)
17 //from above equations,we get
18 //tand(theta)=((w*w*L*(1+sind(theta)))/g)
   .....(3)
19 w=sqrt((g*tand(theta))/(L*(1+sind(theta))))
20
21 printf('the angular speed of rotation is %3.1f rad/s
   ',w)

```

Scilab code Exa 7.10w calculation of the minimum speed at which floor may be removed

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 7.10w

```

```

5 //calculation of the minimum speed at which floor
   may be removed
6
7 //given data
8 r=2//radius(in m) of the rotor
9 mus=0.2//coefficient of static friction between the
   wall and the person
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 //by applying Newton's second law for horizontal
    direction
14 //fs = m*g.....(1)
15 //by limiting friction
16 //mus*N = m*g or mus*m*v*v/r = m*g.....(2)
17 //from above equations ,we get
18 v=sqrt(r*g/mus)
19
20 printf('the minimum speed at which floor may be
    removed is %3.1f m/s',v)

```

Chapter 8

Work and Energy

Scilab code Exa 8.1 calculation of the work done by the spring force

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.1
5 //calculation of the work done by the spring force
6
7 //given data
8 k=50//spring constant(in N/m) of the spring
9 x=1*10^-2//compression(in m) from natural position
10
11 //calculation
12 W=(k*x*x)/2//work done in compressing a spring
13
14 printf('the work done by the spring force is %3.1e J
         ',W)
```

Scilab code Exa 8.1w calculation of the work done by the porter on the suitcase

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.1w
5 //calculation of the work done by the porter on the
   suitcase
6
7 //given data
8 m=20//mass(in kg) of suitcase
9 h=2//height(in m) above the platform
10 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation
13 W=-m*g*h//work done by gravity
14 //the work done by the porter = negative of the work
   done by gravity
15
16 printf('the work done by the porter on the suitcase
   is %d J',-W)

```

Scilab code Exa 8.2 calculation of the work done by force of gravity

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.2
5 //calculation of the work done by force of gravity
6
7 //given data
8 m=20*10^-3//mass(in kg) of the particle
9 u=10//speed(in m/s) of the particle
10 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
11

```

```

12 //calculation
13 //from equation of motion.....(v*v)=(u*u)-(2*g*h)
    .....take v=0 we get
14 h=(u*u)/(2*g)
15 W=-m*g*h//law of conservation of energy
16
17 printf('the work done by force by gravity is %3.1f J
    ',W)

```

Scilab code Exa 8.2w calculation of the value of minimum horsepower of the motor

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.2w
5 //calculation of the value of minimum horsepower of
    the motor to be used
6
7 //given data
8 m=500//mass(in kg) of the elevator
9 v=.20//velocity(in m/s) of the elevator
10 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 P=m*g*v//power = force*velocity
14
15 printf('the value of minimum horsepower of the motor
    to be used is %3.2f hp',P/746)

```

Scilab code Exa 8.3w calculation of the power delivered by the pulling force and

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.3w
5 //calculation of the power delivered by the pulling
   force and average power
6
7 //given data
8 m=2//mass(in kg)
9 theta=30//angle(in degree)
10 a=1//acceleration(in m/s^2) of the block
11 t=4//time(in s)
12 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
13
14 //calculation
15 F=(m*g*sind(theta))+(m*a)//resolving the forces
   parallel to the incline
16 v=a*t
17 P=F*v//equation of power
18 d=a*t*t/2//from equation of motion
19 W=F*d
20 pavg=W/t//average power delivered
21
22 printf('the power delivered by the pulling force at
   t=4 s is %d W',P)
23 printf('\nthe average power delivered by the pulling
   force between t=0 s to t=4 s is %3.1f W',pavg)

```

Scilab code Exa 8.4w calculation of the work done by the given force

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.4w
5 //calculation of the work done by the given force

```

```

6
7 //given data
8 function F=f(x)
9     F=(10+(.50*x))
10 endfunction
11 x1=0//initial position(in m) of the particle
12 x2=2//final position(in m) of the particle
13
14 //calculation
15 W=integrate('f','x',x1,x2)//work done
16
17 printf('the work done by the given force for the
        given displacement is %d J',W)

```

Scilab code Exa 8.5 calculation of the speed of the pendulum of bob when it makes

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.5
5 //calculation of the speed of the pendulum of bob
  when it makes an angle of 60 degree with the
  vertical
6
7 //given data
8 v0=3//speed(in m/s)of the bob in its lowest position
9 theta=60//angle(in degree)made by the pendulum with
  vertical
10 l=.5//length(in m) of the pendulum
11 g=10//gravitational acceleration(in m/s^2) of the
  earth
12
13 //calculation
14 //from the law of conservation of energy
15 //(m*v0*v0/2) - (m*v1*v1/2) = m*g*l*(1-cosd(theta))

```

```

16 v1=sqrt((v0*v0)-(2*g*l*(1-cosd(theta))))
17
18 printf('the speed of the pendulum of bob when it
        makes an angle of 60 degree with the vertical is
        %d m/s ',v1)

```

Scilab code Exa 8.11w calculation of the speed of the particle at a given point

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.11w
5 //calculation of the speed of the particle at a
  given point
6
7 //given data
8 hA=1//height(in m) of point A
9 hB=.5//height(in m) of point B
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 //potential energies at point A and B are
14 //UA = M*g*hA
15 //UB = M*g*hB.....(1)
16 //principle of conservation of energy
17 //UA + KA = UB + KB.....(2)
18 vB=sqrt(2*g*(hA-hB))
19
20 printf('the speed of the particle at a B point is %3
        .2f m/s ',vB)

```

Scilab code Exa 8.12w calculation of the maximum compression of the spring

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 8.12w
5 //calculation of the maximum compression of the
   spring
6
7 //given data
8 k=400//spring constant(in N/m)
9 m=40*10-3//mass(in kg)
10 h=4.9//height(in m)
11 g=9.8//gravitational acceleration(in m/s2) of the
   earth
12
13 //calculation
14 //m*g*h = (k*x*x/2)
15 x=sqrt((2*m*g*h)/k)
16
17 printf('the maximum compression of the spring is %3
   .3f m or %3.1f cm',x,x*102)

```

Chapter 9

Centre of Mass Linear momentum Collision

Scilab code Exa 9.1w Locating the centre of maass of the system

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.1w
5 //Locating the centre of maass of the system
6
7 //given data
8 m1=.50//mass(in kg) at point1
9 m2=1//mass(in kg) at point2
10 m3=1.5//mass(in kg) at point3
11 x1=0//x coodinate (in cm) of point1
12 x2=4//x coodinate (in cm) of point2
13 x3=0//x coodinate (in cm) of point3
14 y1=0//y coodinate (in cm) of point1
15 y2=0//y coodinate (in cm) of point2
16 y3=3//y coodinate (in cm) of point3
17
18 //calculation
19 X=((m1*x1)+(m2*x2)+(m3*x3))/(m1+m2+m3)
```



```

20 Y=((m1*y1)+(m2*y2)+(m3*y3))/(m1+m2+m3)
21
22 printf('The centre of mass is %3.1f cm right and %3
        .1f cm left above the .5 kg paticle.',X,Y)

```

Scilab code Exa 9.4 calculation of the maximum compression of the string

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.4
5 //calculation of the maximum compression of the
  string
6
7 //given data
8 m=1//mass(in kg)
9 v=2//speed of the block(in m/s)
10 k=50//spring constant(in N/m)
11
12 //calculation
13 V=(m*v)/(m+m)//principle of conservation of linear
  momentum
14 ke1=(m*v*v/2)//initial kinetic energy
15 ke2=(m*V*V/2)+(m*V*V/2)//final kinetic energy
16 x=sqrt(2*(ke1-ke2)/k)//kinetic energy lost = elastic
  energy stored
17
18 printf('the maximum compression of the string is %3
        .1 f m',x)

```

Scilab code Exa 9.5 calculation of the speed of combined mass

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.5
5 //calculation of the speed of combined mass
6
7 //given data
8 ma=50//mass(in kg) of cart A
9 mb=20//mass(in kg) of cart B
10 va=20//velocity(in km/hr) of cart A
11 vb=10//velocity(in km/hr) of cart B
12
13 //calculation
14 V=((ma*va)-(mb*vb))/(ma+mb)//principle of
    conservation of linear momentum
15
16 printf('the speed of combined mass after collision
    is %3.2f km/hr ',V)

```

Scilab code Exa 9.6w calculation of the acceleration of the centre of mass

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.6w
5 //calculation of the acceleration of the centre of
    mass
6
7 //given data
8 M=2.5//mass(in kg) of the body
9 F1=6//force(in N) acting at point 1
10 F2=5//force(in N) acting at point 2
11 F3=6//force(in N) acting at point 3
12 F4=4//force(in N) acting at point 4
13 theta1=0//angle(in degree)
14 theta2=37//angle(in degree)

```

```

15 theta3=53//angle(in degree)
16 theta4=60//angle(in degree)
17
18 //calculation
19 Fx=(-F1*cosd(theta1))+(F2*cosd(theta2))+(F3*cosd(
    theta3))+(F4*cosd(theta4))//X component of
    resultant force
20 Fy=(F1*sind(theta1))+(F2*sind(theta2))+(-F3*sind(
    theta3))+(F4*sind(theta4))//Y component of
    resultant force
21 F=sqrt((Fx*Fx)+(Fy*Fy))
22 theta=atand(Fy/Fx)
23 acm=F/M//acceleration of centre of mass
24
25 printf('the acceleration of the centre of mass is %3
    .1f m/s^2 and is in the direction of the
    resultant force',acm)

```

Scilab code Exa 9.8w calculation of the distance from launching point

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.8w
5 //calculation of the distance from launching point
6
7 //given data
8 u=100//speed(in m/s) of the projectile
9 theta=37//angle(in degree) of the projectile above
    horizontal
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 xcm=(2*u*u*sind(theta)*cosd(theta))/g//range of

```

```

    original projectile
14 //also   xcm=((m1*x1)+(m2*x2))/(m1+m2)
15 //here m1=M/4   and   m2=3*M/4
16 x1=xcm/2//since small part falls from heighest point
    i.e half of range
17 x2=(4/3)*((xcm*((1/4)+(3/4)))-(x1/4))
18
19 printf('the distance of landing of heavier piece
    from launching point is %d m',x2)

```

Scilab code Exa 9.9w calculation of the distance moved by the bigger block

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.9w
5 //calculation of the distance moved by the bigger
    block
6
7 //given data
8 L=2.2//length(in m) of the base
9 n=10// mass of bigger block is 'n' number of times
    the mass of smaller block
10
11 //calculation
12 //centre of mass at rest initially will remain in
    horizontal position thus
13 //M*(L-X)=10*M*X
14 X=L/(n+1)
15
16 printf('distance moved by the bigger block at the
    instant the smaller block reaches the ground is
    %3.1f m',X)

```

Scilab code Exa 9.10w calculation of the average force exerted by the hero on the

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.10w
5 //calculation of the average force exerted by the
   hero on the machine gun
6
7 //given data
8  $m=50 \times 10^{-3}$  //mass(in kg) of the bullet
9  $v=1 \times 10^3$  //velocity(in m/s) of the bullet
10  $n=20$  //number of bullets fired
11  $t=4$  //time(in s) required in firing the bullets
12
13 //calculation
14  $me=m*v$  //momentumof each bullet
15  $f=me*n/t$  //force=rate of change of momentum
16
17 printf('the average force exerted by the hero on the
   machine gun is %d N',f)
```

Scilab code Exa 9.11w calculation of the fractional change in kinetic energy

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.11w
5 //calculation of the fractional change in kinetic
   energy
6
7 //given data
```

```

8 vb=20//speed(in m/s) of the block
9 v1=30//velocity(in m/s) of one of the part
10
11 //calculation
12 M=1//taking mass M=1 kg for solving the equation
13 v=(1/M)*((M*vb*2)-(M*v1))//principle of conservation
    of linear momentum
14 deltake=(M*v1*v1/(2*2))+(M*v*v/(2*2))-(M*vb*vb/2)//
    change in the kinetic energy
15 fdeltake=deltake/(M*vb*vb/2)//fractional change in
    the kinetic energy
16
17 printf('the fractional change in the kinetic energy
    is %3.2f',fdeltake)

```

Scilab code Exa 9.13w calculation of the final velocity of the shuttle

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.13w
5 //calculation of the final velocity of the shuttle
6
7 //given data
8 v1=4000//speed(in km/hr) of shuttle with respect to
    the earth
9 v2=100//speed(in km/hr) of the module with respect
    to the shuttle
10
11 //calculation
12 M=1//taking mass M=1 kg for solving the equation
13 vdash=v1-v2//speed of module with respect to the
    earth
14 V=(1/5)*((1*v1*6)-(vdash*1))//principle of
    conservation of linear momentum

```

```
15
16 printf('the final velocity of the shuttle is %d km/h
    ',V)
```

Scilab code Exa 9.14w calculation of the velocity with which the board recoils

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.14w
5 //calculation of the velocity with which the board
    recoils
6
7 //given data
8 m1=25//mass(in kg) of the boy
9 m2=10//mass(in kg) of the board
10 v1=5//velocity(in m/s) of the boy
11
12 //calculation
13 v=(m1*v1)/m2//principle of conservation of linear
    momentum
14 vsep=v1+v//velocity of separation
15
16 printf('the velocity with which the board recoils is
    %3.1f m/s',v)
17 printf('\\nthe velocity of separation of the boy and
    the board is %3.1f m/s',vsep)
```

Scilab code Exa 9.17w calculation of the speed of the bullet

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
```

```

4 //example 9.17w
5 //calculation of the speed of the bullet
6
7 //given data
8 mb=50*10^-3//mass(in kg) of the bullet
9 mp=450*10^-3//mass(in kg) of the bob
10 h=1.8//height(in m) attained by the bob
11 g=10//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 //using principle of conservation of linear momentum
    and equation of motion (v*v) = (u*u) + (2*a*x)
15 v=((mb+mp)*(sqrt(h*2*g)))/mb
16
17 printf('the speed of the bullet is %d m/s',v)

```

Scilab code Exa 9.22w calculation of the loss of kinetic energy due to the collision

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 9.22w
5 //calculation of the loss of kinetic energy due to
    the collision
6
7 //given data
8 m=1.2//mass(in kg) of the block1
9 v=20*10^-2//velocity(in m/s) of the approach
10 e=3/5//value of coefficient of restitution
11 vdash=e*v//velocity (in m/s) of the separation
12
13 //calculation
14 //by principle of conservation of linear momentum
    ....v1 + v2 = v m/s.....(1)

```



```

15 //as the coefficient of restitution is
    3/5.....v2 - v1 = vdash m/s.....(2)
16 //from equation (1),we get.....v2=v-v1
17 //substituting v2 in equation (2),we get
18 v1=(v-vdash)/2
19 v2=v-v1//from equation (1)
20 lke=(m/2)*((v*v)-(v1*v1)-(v2*v2))
21
22 printf('the loss of kinetic energy during the
    collision is %3.1e J',lke)

```

Chapter 10

Rotational Mechanics

Scilab code Exa 10.1 calculation of the number of revolutions made

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.1
5 //calculation of the number of revolutions made
6
7 //given data
8 wzero=100*2*%pi/60//initial angular velocity(in rad/
   s) of the motor
9 w=0//final angular velocity(in rad/s) of the motor
10 t=15//time interval(in s)
11
12 //calculation
13 alpha=(w-wzero)/t//equation of angular motion
14 theta=(wzero*t)+(alpha*t*t/2)//equation of angular
   motion
15
16 printf('the number of revolutions the motor makes
   before coming to rest is %3.1f',theta/(2*%pi))
```

Scilab code Exa 10.1w calculation of the number of revolutions made by the wheel

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.1w
5 //calculation of the number of revolutions made by
   the wheel
6
7 //given data
8 wzero=0//initial angular velocity(in rad/s) of the
   wheel
9 alpha=2//angular acceleration(in rad/s^2)
10 t=10//time(in s) interval
11
12 //calculation
13 theta=(wzero*t)+(alpha*t*t/2)//equation of angular
   motion
14 n=round(theta/(2*%pi))//number of revolutions
15
16 printf('the number of revolutions made by the wheel
   is %d',n)
```

Scilab code Exa 10.2 calculation of the time taken by the fan to attain half of th

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.2
5 //calculation of the time taken by the fan to attain
   half of the maximum speed
6
```

```

7 //given data
8 wzero=0//initial angular velocity(in rad/s) of the
   fan
9 w=400*(2*pi/60)//final angular velocity(in rad/s)
   of the fan
10 t=5//tiem(in s) taken
11
12 //calculation
13 alpha=(w-wzero)/t//equation of angular motion
14 wdash=w/2//half of maximum speed
15 t1=(wdash-wzero)/alpha//equation of angular motion
16
17 printf('the time taken by the fan to attain half of
   the maximum speed is %3.1f s',t1)

```

Scilab code Exa 10.2w calculation of the angle rotated during the next second

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.2w
5 //calculation of the angle rotated during the next
   second
6
7 //given data
8 theta=2.5//angular displacement(in rad) of the wheel
9 t=1//time(in s) required
10
11 //calculation
12 alpha=(theta*2)/(t*t)//equation of angular motion
13 theta1=(alpha*(t+1)*(t+1)/2)//angle rotated during
   first two seconds
14 thetar=theta1-theta//angle rotated during next
   second
15

```

```
16 printf('the angle rotated during the next second is
    %3.1f rad ',thetar)
```

Scilab code Exa 10.3 calculation of the angular velocity and angular acceleration

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.3
5 //calculation of the angular velocity and angular
    acceleration of the pulley
6
7 //given data
8 v=20//linear speed(in cm/s) of the bucket
9 r=10//radius(in cm) of the pulley
10 a=4*10^2//linear acceleration(in cm/s^2) of the
    pulley
11
12 //calculation
13 w=v/r//formula of angular velocity
14 alpha=a/r//formula of angular acceleration
15
16 printf('the angular velocity of the pulley is %d rad
    /s and angular acceleration of the pulley is %d
    rad/s^2 ',w,alpha)
```

Scilab code Exa 10.3w calculation of the torque required to stop the wheel in one

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.3w
```

```

5 //calculation of the torque required to stop the
   wheel in one minute
6
7 //given data
8 wzero=50*(2*%pi/60)//initial angular velocity(in rad
   /s) of the wheel
9 w=0//final angular velocity(in rad/s) of the wheel
10 t=60//time(in s) taken to stop the wheel
11 I=2//moment of inertia(in kg-m^2) of the wheel
12
13 //calculation
14 alpha=(w-wzero)/t//equation of angular motion
15 tau=I*abs(alpha)//torque
16
17 printf('the torque required to stop the wheel in one
   minute is %3.2f N-m',tau)

```

Scilab code Exa 10.4w calculation of the angular velocity of the wheel

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.4w
5 //calculation of the angular velocity of the wheel
6
7 //given data
8 F=20//force(in N) of pull applied
9 I=.2//moment of inertia(in kg-m^2)
10 r=20*10^-2//radius(in m) of the wheel
11 t=5//time(in s) interval
12 wzero=0//initial angular velocity(in rad/s) of the
   wheel
13
14 //calculation
15 tau=F*r//torque applied to the wheel

```

```

16 alpha=tau/I//angular acceleration
17 w=wzero+(alpha*t)//equation of angular motion
18
19 printf('the angular velocity of the wheel after 5 s
        is %d rad/s',w)

```

Scilab code Exa 10.5 calculation of the moment of inertia of the wheel

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.5
5 //calculation of the moment of inertia of the wheel
6
7 //given data
8 r=10*10^-2//radius(in m) of the wheel
9 F=5//force(in N) of pulling
10 aplha=2//angular acceleration(in rad/s^2) of the
    wheel
11
12 //calculation
13 tau=F*r//net torque
14 I=tau/aplha//moment of inertia
15
16 printf('the moment of inertia of the wheel is %3.2f
        kg-m^2',I)

```

Scilab code Exa 10.7w calculation of the position of second kid on a balanced see

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.7w

```

```

5 //calculation of the position of second kid on a
   balanced seesaw
6
7 //given data
8 ma=10//mass(in kg) of kid A
9 mb=15//mass(in kg) of kid B
10 l=5//length(in m) of the seesaw
11 la=(l/2)//distance of A kid from fulcrum as he is
   sitting at an end
12
13 //calculation
14 //taking torque about fulcrum.....(mb*g*x) = (
   ma*g*)
15 x=(ma*la)/mb
16
17 printf('the second kid should sit at a distance of
   %3.1f m from the centre ',x)

```

Scilab code Exa 10.8w calculation of the normal force and the frictional force th

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.8w
5 //calculation of the normal force and the frictional
   force that the floor exerts on the ladder
6
7 //given data
8 m=10//mass(in kg) of the ladder
9 theta=53//angle(in degree) made by the ladder
   against the vertical wall
10 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation

```



```

13 //taking horizontal and vertical components
14 //N1 = f.....(1)
15 //N2 = W.....(2)
16 //taking torque about B
17 W=m*g
18 N2=W//from equation (2)
19 f=(W*sind(theta)/2)/(cosd(theta))//from equation (1)
20
21 printf('the normal force that the floor exerts on
        the ladder is %d N',N2)
22 printf('\nthe frictional force that the floor exerts
        on the ladder is %d N',f)

```

Scilab code Exa 10.9w calculation of the contact force exerted by the floor on ea

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.9w
5 //calculation of the contact force exerted by the
  floor on each leg of ladder
6
7 //given data
8 theta=60//angle(in degree) between the two legs
9 m=80//mass(in kg) of the person
10 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 N=m*g/2
14 T=(N*2*tand(90-theta))/1
15
16 printf('the contact force exerted by the floor on
        each leg of ladder %d N',N)
17 printf('\nthe tension in the crossbar is %d N',T)

```

Scilab code Exa 10.12 calculation of the kinetic energy of the sphere

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.12
5 //calculation of the kinetic energy of the sphere
6
7 //given data
8 M=200*10^-3//mass(in kg) of the sphere
9 vcm=2*10^-2//speed(in m/s) of the sphere
10
11 //calculation
12 //kinetic energy is  $K = (I_{cm} * w * w / 2) + (M * v_{cm} * v_{cm} / 2)$ 
13 //taking  $I_{cm} = (2 * M * r * r * w * w / 5)$  and  $w = v_{cm} / r$ 
14  $K = (M * v_{cm} * v_{cm} / 5) + (M * v_{cm} * v_{cm} / 2)$  //kinetic energy
15
16 printf('the kinetic energy of the sphere is %3.1e J'
        ,K)
```

Scilab code Exa 10.13w calculation of the kinetic energy and angular momentum of

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.13w
5 //calculation of the kinetic energy and angular
   momentum of the disc
6
7 //given data
8 M=200*10^-3//mass(in kg) of the disc
```

```

9 r=4*10^-2//radius(in m) of the disc
10 w=10//angular velocity(in rad/s)
11
12 //calculation
13 I=(M*r*r)/4//moment of inertia
14 K=(I*w*w/2)//kinetic energy
15 L=I*w//angular momentum
16
17 printf('the kinetic energy of the disc is %3.1e J',K
)
18 printf('\nthe angular momentum of the disc is %3.1e
J-s ',L)

```

Scilab code Exa 10.14w calculation of the work done by the torque in first two seconds

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.14w
5 //calculation of the work done by the torque in
first two seconds
6
7 //given data
8 wzero=20//initial angular velocity(in rad/s) of the
motor
9 w=0//final angular velocity(in rad/s) of the motor
10 t=4//time(in s) taken to attain rest position
11 I=.20//moment of inertia(in kg-m^2) of the disc
about axis of rotation
12 t1=2//time(in s)
13
14 //calculation
15 alpha=(wzero-w)/t//equation of angular motion in
case of deceleration
16 tau=I*alpha//torque

```

```

17 theta=(wzero*t1)-(alpha*t1*t1/2)//equation of
    angular motion
18 W=tau*theta//work done by the torque
19
20 printf('the work done by the torque in first two
    seconds is %d J',W)

```

Scilab code Exa 10.19w calculation of the moment of inertia of the system about t

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.19w
5 //calculation of the moment of inertia of the system
    about the axis perpendicular to the rod passing
    through its middle point
6
7 //given data
8 m=1.2//mass(in kg) of the sphere
9 R=10*10^-2//radius(in cm) of the sphere
10 sep=50*10^-2//separation(in m) between the two
    spheres
11
12 //calculation
13 d=sep/2//distance of each sphere from centre
14 Icm=(2*m*R*R)/5//moment of inertia about diameter
15 I=Icm+(m*d*d)//by parallel axis theorem,moment of
    inertia about given axis
16 //since second sphere has same moment of inertia
17 Isys=2*I//moment of inertia of the system
18
19 printf('the moment of inertia of the system about
    the axis perpendicular to the rod passing through
    its middle point is %3.3f kg-m^2',Isys)

```

Scilab code Exa 10.22w calculation of the number of revolutions made by the wheel

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 10.22w
5 //calculation of the number of revolutions made by
   the wheel per second
6
7 //given data
8 p=220*10^-2//perimeter(in cm) of the wheel
9 v=9*10^3/(60*60)//linear speed(in m/s) of wheel on
   the road
10
11 //calculation
12 r=p/(2*%pi)//radius of the wheel
13 w=v/r//angular speed
14 n=w/(2*%pi)//number of revolutions
15
16 printf('the number of revolutions made by the wheel
   per second is %3.2f rev/s',n)
```

Chapter 11

Gravitation

Scilab code Exa 11.1 calculation of the initial acceleration of the particles

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.1
5 //calculation of the initial acceleration of the
   particles
6
7 //given data
8 m1=1//masss(in kg) of particle1
9 m2=2//masss(in kg) of particle2
10 r=50*10-2//separation(in m) between the two
   particles
11 G=6.67*10-11//universal constant of gravitation(in
   N-m2/kg2)
12
13 //calculation
14 F=G*m1*m2/(r*r)//force of gravitation
15 a1=F/m1//initial acceleration of the particle1
16 a2=F/m2//initial acceleration of the particle2
17
18 printf('the initial acceleration of the particle1
```

```

    towards particle2 is %3.1e m/s^2',a1)
19 printf('\nthe initial acceleration of the particle2
    towards particle1 is %3.1e m/s^2',a2)

```

Scilab code Exa 11.2 calculation of the work done in bringing three particles together

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.2
5 //calculation of the work done in bringing three
  particles together
6
7 //given data
8 m1=100*10^-3//mass(in kg) of particle1
9 r=20*10^-2//separation(in m) between the two
  particles
10 G=6.67*10^-11//universal constant of gravitation(in
  N-m^2/kg^2)
11
12 //calculation
13 //since the work done by the gravitational force is
  equal to change in the potential energy
14 U=3*(-G*m1*m1/r)
15
16 printf('the work done in bringing three particles is
  %3.1e J',U)

```

Scilab code Exa 11.2w calculation of the distance from the earth surface where re

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 11.2w
5 //calculation of the distance from the earth's
   surface where resultant gravitational field due
   to the earth and the moon is zero
6
7 //given data
8 Me=6*10^24//mass(in kg) of the earth
9 Mm=7.4*10^22//mass(in kg) of the moon
10 d=4*10^5*10^3//distance(in m) between the earth and
   the moon
11
12 //calculation
13 //gravitational field due to the earth at that point
14 //E1 = G*Me/x^2.....(1)
15 //gravitational field due to the moon at that point
16 //E2 = G*Mm/(d-x)^2.....(2)
17 //E1 = E2..... given
18 x=(d*sqrt(Me/Mm))/(1+sqrt(Me/Mm))
19
20 printf('the distance from the earth surface where
   resultant gravitational field due to the earth
   and the moon is zero is %3.1e km',x*10^-3)

```

Scilab code Exa 11.4 calculation of the gravitational field

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.4
5 //calculation of the gravitational field
6
7 //given data
8 F=2//gravitational force(in N)
9 m=50*10^-3//mass(in kg) of the particle
10

```



```

11 //calculation
12 E=F/m//gravitational field
13
14 printf('the gravitational field along the direction
    of force is %d N/kg',E)

```

Scilab code Exa 11.4w calculation of the separation between the particles under mutual attraction

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.4w
5 //calculation of the separation between the
    particles under mutual attraction
6
7 //given data
8 mA=1//mass(in kg) of particle A
9 mB=2//mass(in kg) of particle B
10 R=1//initial distance(in m) between the two
    particles
11 vB=3.6*10^-2/(60*60)//speed(in m/s) of the particle
    B
12 G=6.67*10^-11//universal constant of gravitation(in
    N-m^2/kg^2)
13
14 //calculation
15 v=(mB*vB)/mA//principle of conservation of linear
    momentum
16 U1=-G*mA*mB/R//initial potential energy of the pair
17 d=U1/(U1-(mB*vB*vB/2)-(mA*v*v/2))//principle of
    conservation of energy
18
19 printf('the speed of particle A is %3.1e m/s',v)
20 printf('\nthe separation between the particles under
    mutual attraction is %3.2f m',d)

```

Scilab code Exa 11.5w calculation of the work done by an external agent

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.5w
5 //calculation of the work done by an external agent
6
7 //given data
8 //E = (10 N/kg)(i + j).....given gravitational field
9 Ex=10//value of X-component of gravitational field(
   in N/kg)
10 Ey=10//value of Y-component of gravitational field(
   in N/kg)
11 m=2//mass(in kg) of the gravitational field
12 x0=0//value of X component of initial location(in m)
13 x1=5//value of X component of final location(in m)
14 y0=0//value of Y component of initial location(in m)
15 y1=4//value of Y component of final location(in m)
16
17 //calculation
18 function Fx=fx(x)
19     Fx=m*Ex//value of X component of force
20 endfunction
21
22 function Fy=fy(x)
23     Fy=m*Ey//value of Y component of force
24 endfunction
25
26 //calculation
27 W1=integrate('fx','x',x0,x1)//work done by X
   component of external force
28 W2=integrate('fy','x',y0,y1)//work done by Y
   component of external force
```

```

29 W=W1+W2
30
31 printf('the work done by the external agent is %d J'
        ,-W)

```

Scilab code Exa 11.7 calculation of the gravitational field due to the moon at its

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.7
5 //calculation of the gravitational field due to the
   moon at its surface
6
7 //given data
8 M=7.36*10^22//mass(in kg) of the moon
9 G=6.67*10^-11//universal constant of gravitation(in
   N-m^2/kg^2)
10 a=1.74*10^6//radius(in m) of the moon
11
12 //calculation
13 E=G*M/(a*a)//formula of gravitational field
14
15 printf('the gravitational field due to the moon at
   its surface is %3.2f N/kg',E)

```

Scilab code Exa 11.8 calculation of the value of acceleration due to gavity

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.8

```

```

5 //calculation of the value of acceleration due to
   gavity
6
7 //given data
8 h=5*10^3//height(in m) above the earth's surface
9 R=6400*10^3//radius(in m) of the earth
10 g0=9.8//gravitational acceleration(in m/s^2) of the
   earth
11 d=5*10^3//depth(in m) below the earth's surface
12
13 //calculation
14 gh=g0*(1-(2*h/R))//formula of gravitational
   acceleration at height h above the earth's
   surface
15 gd=g0*(1-(d/R))//formula of gravitational
   acceleration at depth d below the earth's surface
16
17 printf('the value of gravitational acceleration at
   height 5 km above the earth surface is %3.2f m/s
   ^2',gh)
18 printf('\nthe value of gravitational acceleration at
   depth 5 km below the earth surface is %3.2f m/s
   ^2',gd)

```

Scilab code Exa 11.9 calculation of the speed and time period of the satellite

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.9
5 //calculation of the speed and time period of the
   satellite
6
7 //given data
8 h=600*10^3//height(in m) of the satellite

```

```

 9 M=6*10^24//mass(in kg) of the earth
10 R=6400*10^3//radius(in m) of the earth
11 G=6.67*10^-11//universal constant of gravitation(in
    N-m^2/kg^2)
12
13 //calculation
14 a=h+R//distance of satellite from centre of the
    earth
15 v=sqrt(G*M/a)//speed of satellite
16 T=(2*pi*a)/v//time period of satellite
17
18 printf('the speed of the satellite is %3.1e m/s or
    %3.1f km/s ',v,v*10^-3)
19 printf('\nthe time period of the satellite is %3.1e
    s ',T)

```

Scilab code Exa 11.9w calculation of the maximum height attained by the particle

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.9w
5 //calculation of the maximum height attained by the
    particle
6
7 //given data
8 v0=9.8*10^3//speed(in m/s) the particle is fired
9 R=6400*10^3//radius(in m) of the earth
10 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 //by the principle of conservation of energy
14 //(-G*M*m/R) + (m*v0*v0/2) = -(G*M*m/(R+H))
15 H=(R*R/(R-(v0*v0/(2*g))))-R

```

```
16
17 printf('the maximum height attained by the particle
    is %d km',H*10^-3)
```

Scilab code Exa 11.10 calculation of the escape velocity from the moon

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.10
5 //calculation of the escape velocity from the moon
6
7 //given data
8 M=7.4*10^22//mass(in kg) of the moon
9 R=1740*10^3//radius(in m) of the moon
10 G=6.67*10^-11//universal constant of gravitation(in
    N-m^2/kg^2)
11
12 //calculation
13 v=sqrt(2*G*M/R)//formula of the escape velocity
14
15 printf('the escape velocity from the moon is %3.1f
    km/s',v*10^-3)
```

Scilab code Exa 11.10w calculation of the stretch produced in the spring

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.10w
5 //calculation of the stretch produced in the spring
6
7 //given data
```

```

8 d=1*10^-2//stretch(in m) of the spring
9 R=6400*10^3//radius(in m) of the earth
10 h=800*10^3//height(in m) above the earth's surface
11
12 //calculation
13 //The extension in the spring on the surface is
14 //1*10^-2 = (G*M*m)/(k*R^2) .....(1)
15 //The extension in the spring at height h above the
    surface
16 //x = (G*M*m)/(k*(R+h)^2) .....(2)
17 //from above equations ,we get
18 x=d*((R^2)/(R+h)^2)
19
20 printf('the stretch produced in the spring is %3.2f
    cm',x*10^2)

```

Scilab code Exa 11.11w calculation of time period of the pendulum if used at the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.11w
5 //calculation of time period of the pendulum if used
    at the equator
6
7 //given data
8 t=2//time period (in s) of the pendulum at North
    pole
9 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
10 G=6.67*10^-11//universal constant of gravitation(in
    N-m^2/kg^2)
11 w=(2*%pi)/(24*60*60)//angular velocity(in rad/s) of
    the earth
12 R=6400*10^3//radius(in m) of the earth

```

```

13
14 //calculation
15 //By equilibrium conditions ,we get
16 //t = 2*%pi*sqrt(1/g)
17 //tdash = 2*%pi*sqrt(1/(g-(w*w*R)))
18 //from equations (1) and (2),we get
19 tdash=t*(1+(w*w*R/(2*g)))
20
21 printf('the value of time period of the pendulum if
        used at the equator is %3.4f s',tdash)

```

Scilab code Exa 11.12w calculation of the speed of projection of the satellite in

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.12w
5 //calculation of the speed of projection of the
6 //satellite into an orbit
7
8 //given data
9 r=8000*10^3//radius(in m) of the orbit of the
10 //satellite
11 R=6400*10^3//radius(in m) of the earth
12 g=9.8//gravitational acceleration(in m/s^2) of the
13 //earth
14
15 //calculation
16 //using Newton's second law
17 //((G*M*m)/(r*r)) = m*v*v/r
18 v=sqrt(g*R*R/r)
19 t=(2*%pi*r/v)//time period of the satellite
20

```



```

18 printf('the speed of projection of the satellite
    into the orbit is %3.2f km/s',v*10^-3)
19 printf('\nthe time period of the satellite in the
    orbit is %d minutes',t*(1/(60)))

```

Scilab code Exa 11.13w calculation of the speed and the angular speed of the satellite

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 11.13w
5 //calculation of the speed and the angular speed of
    the satellite S2 relative to the satellite S1
6
7 //given data
8 T1=1//period of revolution(in h) of satellite S1
9 T2=8//period of revolution(in h) of satellite S2
10 R1=10^4//radius(in km) of the orbit of satellite S1
11
12 //calculation
13 //by Kepler's third law
14 //(R2/R1)^3 = (T2/T1)^2
15 R2=R1*(((T2/T1)^2)^(1/3))
16 v1=(2*%pi*R1/T1)//speed(in km/h) of satellite S1
17 v2=(2*%pi*R2/T2)//speed(in km/h) of satellite S2
18 v=abs(v2-v1)//speed of satellite S2 with respect to
    satellite S1
19 w=v/(R2-R1)//angular speed of satellite S2 as
    observed by an astronaut in satellite S1
20
21 printf('the speed of the satellite S2 with respect
    to the satellite S1 is %3.1e km/h',v)
22 printf('\nthe angular speed of the satellite S2 as
    observed by an astronaut in the satellite S1 is
    %3.2f rad/h',w)

```


Chapter 12

Simple Harmonic Motion

Scilab code Exa 12.1 calculation of the spring constant

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.1
5 //calculation of the spring constant
6
7 //given data
8 F=4//force(in N) acting
9 x=5*10-2//distance(in m) from the centre
10
11 //calculation
12 k=F/x//value of spring constant
13
14 printf('the value of spring constant is %d N/m',k)
```

Scilab code Exa 12.1w calculation of the amplitude time period maximum speed and v

```
1 //developed in windows XP operating system 32bit
```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.1w
5 //calculation of the amplitude,time period ,maximum
   speed and velocity at time t
6
7 //given data
8 //x = (5 m)*sind((%pi s^-1)t + (180/3)) .....
   equation of simple harmonic motion
9
10 //calculation
11 A=5//amplitude(in m)
12 w=%pi
13 T=(2*%pi)/w//time period(in s)
14 vmax=A*w//maximum speed
15 v=A*w*cosd(180+(180/3))
16
17 printf('the amplitude is %d m',A)
18 printf('\\nthe time period is %d s',T)
19 printf('\\nthe maximum speed is %3.2f m/s',vmax)
20 printf('\\nthe velocity at time t=1 s is %3.2f m/s',v
   )

```

Scilab code Exa 12.2 calculation of the amplitude of the motion

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.2
5 //calculation of the amplitude of the motion
6
7 //given data
8 m=0.5//mass(in kg) of the particle
9 //F = -50*x ..... force(in N/m)
10 v=10//speed(in m/s) of the oscillation

```

```

11
12 //calculation
13 E=(m*v*v/2)//kinetic energy of the particle at
    centre of oscillation
14 //from principle of conservation of energy .....E =
    (k*A*A/2)
15 A=sqrt(E*2/50)
16
17 printf('the amplitude of the motion is %d m',A)

```

Scilab code Exa 12.2w calculation of the maximum force exerted by the spring on the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.2w
5 //calculation of the maximum force exerted by the
    spring on the block
6
7 //given data
8 m=5//mass(in kg) of the block
9 A=0.1//amplitude(in m) of the motion
10 T=3.14//time period(in s) of the motion
11
12 //calculation
13 w=2*pi/T//angular frequency
14 k=m*w*w//spring constant
15 F=k*A//maximum force
16
17 printf('the maximum force exerted by the spring on
    the block is %d N',F)

```

Scilab code Exa 12.3 calculation of the time period of oscillation of the particle

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.3
5 //calculation of the time period of oscillation of
   the particle
6
7 //given data
8 m=200*10-3//mass(in kg) of the particle
9 k=80//spring constant(in N/m)
10
11 //calculation
12 T=2*%pi*sqrt(m/k)//formula of time period
13
14 printf('the time period of oscillation of the
   particle is %3.2f s',T)

```

Scilab code Exa 12.3w calculation of the maximum time period maximum speed maximum

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.3w
5 //calculation of the maximum time period ,maximum
   speed ,maximum acceleration ,speed for a given
   displacement ,speed at a given time
6
7 //given data
8 w=6.28//angular frequency(in s-1) of simple
   harmonic motion
9 A=10*10-2//amplitude(in m) of simple harmonic
   motion
10 x=6*10-2//displacement(in m) from the mean position
11 t=1/6//time(in s)
12

```

```

13 //calculation
14 T=2*%pi/w//time period
15 vmax=A*w//maximum speed
16 amax=A*w^2//maximum acceleration
17 vx=w*sqrt(A^2-x^2)//speed for displacement x from
    mean position
18 vt=-A*w*sind((w*t)*(180/%pi))//speed at time t
19
20 printf('the time period is %d s',T)
21 printf('\nthe maximum speed is %3.3f m/s',vmax)
22 printf('\nthe maximum acceleration is %d m/s^2',
    round(amax))
23 printf('\nthe speed for displacement x=6 cm from
    mean position is %3.1f cm/s',vx*10^2)
24 printf('\nthe speed at time t= 1/6 s is %3.1f cm/s',
    vt*10^2)

```

Scilab code Exa 12.4 calculation of the value of phase constant

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.4
5 //calculation of the value of phase constant
6
7 //given data
8 //x = A/2
9 //x = A *sind((w*t) + delta)..... equation
10
11 //calculation
12 //at t=0 delta=asind((A/2)/A)
13 delta=asind(1/2)
14 delta1=180-delta//another value of delta
15 //v = dx/dt = A*w*cosd((w*t) + delta)
16 //at t=0 , v = A*w*cosd(delta)

```

```

17 m1=cosd(delta)
18 m2=cosd(delta1)
19 if(m1>0)
20     deltaf=delta//value of v positive at t=0
21 else
22     deltaf=delta1
23 end
24
25 printf('the value of phase constant is %d degree',
        deltaf)

```

Scilab code Exa 12.5 calculation of the total mechanical energy of the system

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.5
5 //calculation of the total mechanical energy of the
  system
6
7 //given data
8 m=40*10^-3//mass(in kg) of the particle
9 A=2*10^-2//amplitude(in cm) of motion
10 T=0.2//time period(in s) of oscillation
11
12 //calculation
13 E=(2*pi*pi*m*A*A)/(T*T)//total mechanical energy
  of the system
14
15 printf('the total mechanical energy of the system is
        %3.1e J',E)

```

Scilab code Exa 12.6 writing the equation giving angular displacement as a function


```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.6
5 //writing the equation giving angular displacement
   as a function of time
6
7 //given data
8 theta0=%pi/10//amplitude(in rad) of motion
9 theta=%pi/10//displacement(in rad) at t=0 s
10 T=.05//time period(in s)
11
12 //calculation
13 //required equation is .....theta = theta0*sind((w*
   t) + delta)
14 w=(2*%pi)/T//value of w in above equation
15 delta=asind(theta/theta0)//value of delta in above
   equation...i.e at t=0
16
17 printf('equation giving angular displacement as a
   function of time is \n theta = (%3.2f rad)*sin[(
   %3.2f s^-1)t + %d] ',theta0,w,delta)

```

Scilab code Exa 12.6w calculation of the maximum speed of the block and the speed

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.6w
5 //calculation of the maximum speed of the block and
   the speed when the spring is stretched
6
7 //given data
8 nu=10//frequency(in s^-1) of oscillation
9 l=.20*10^-2//stretch(in m) of the spring

```

```

10 g=%pi^2//gravitational acceleration(in m/s^2) of the
    earth
11
12 //calculation
13 //Amplitude.....A = m*g/k
    .....(1)
14 //angular frequency.....w=sqrt(k/m)
    .....(2)
15 //from above equations,we get
16 w=2*pi*nu//angular frequency
17 A=((1/w)^2)*g
18 vmax=A*w//maximum speed
19 x=A-1//displacement(in m) from mean position
20 v=w*(sqrt(A^2-x^2))
21
22 printf('the maximum speed of the block is %3.2f cm/s
    ',vmax*10^2)
23 printf('\nthe speed when the spring is stretched by
    0.20 cm is %3.1f cm/s',v*10^2)

```

Scilab code Exa 12.7 calculation of the time period of a pendulum

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.7
5 //calculation of the time period of a pendulum
6
7 //given data
8 g=%pi^2//gravitational acceleration(in m/s^2) of the
    earth
9 l=1//length(in m) of the pendulum
10
11 //calculation
12 T=2*pi*sqrt(l*g^-1)//formula of time period

```

```

13
14 printf('the time period of the pendulum is %3.1f s',
    T)

```

Scilab code Exa 12.8 calculation of the value of the acceleration due to gravity

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.8
5 //calculation of the value of the acceleration due
  to gravity
6
7 //given data
8 t=36//time(in s) taken
9 n=20//number of oscillations
10 l=80*10^-2//effective length(in m)
11
12 //calculation
13 T=t/n//time period
14 g=(4*%pi^2*l)/(T^2)//formula of time period
    .....T=2*%pi*sqrt(l*g^-1)
15
16 printf('the value of the acceleration due to gravity
    is %3.2f m/s^2',g)

```

Scilab code Exa 12.9 calculation of the time period of oscillation

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.9
5 //calculation of the time period of oscillation

```

```

6
7 //given data
8 L=1//length(in m) of the rod
9 g=9.8//gravitational acceleration(in m/s^2) of the
  earth
10
11 //calculation
12 //from formula of time period.....T = 2*%pi*sqrt(I
  /(m*g*l))
13 //for uniform rod ....I = (m*L*L*L/3) and l=L/2
14 T=2*%pi*sqrt((2*L)/(3*g))
15
16 printf('the time period of oscillation is %3.2f s',T
  )

```

Scilab code Exa 12.10 calculation of the value of torsional constant of the wire

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.10
5 //calculation of the value of torsional constant of
  the wire
6
7 //given data
8 m=200*10^-3//mass(in kg) of the disc
9 r=5*10^-2//radius(in m) of the disc
10 T=0.2//time period(in s) of oscillation
11
12 //calculation
13 I=m*r*r/2//moment of inertia of the disc about the
  wire
14 k=4*%pi^2*I/T^2//from formula of time period.....T
  = 2*%pi*sqrt(I/k)
15

```

```
16 printf('the value of torsional constant of the wire
    is %3.2f kg-m^2/s^2',k)
```

Scilab code Exa 12.11 calculation of the amplitude of the simple harmonic motion

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.11
5 //calculation of the amplitude of the simple
    harmonic motion
6
7 //given data
8 //x1 = (2.0 cm)*sind(w*t)
9 //x2 = (2.0 cm)*sind((w*t) + (180/3))
10 A1=2//amplitude(in cm) of the wave 1
11 A2=2//amplitude(in cm) of the wave 2
12 delta=180/3//phase difference(in degree) between the
    two waves
13
14 //calculation
15 A=sqrt(A1^2+A2^2+(2*A1*A2*cosd(delta)))//amplitude
    of the resultant wave
16
17 printf('the amplitude of the simple harmonic motion
    is %3.1f cm',A )
```

Scilab code Exa 12.14w calculation of the time period linear amplitude speed and

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.14w
```

```

5 //calculation of the time period ,linear amplitudde ,
   speed and angular acceleration
6
7 //given data
8 l=40*10^-2//length(in m) of the pendulum
9 theta=.04//angular amplitude(in radian)
10 theta1=.02//angle(in radian) with the vertical
11 g=10//gravitational acceleration(in m/s^2) of the
   earth
12 t=5//time(in s) taken
13
14 //calculation
15 w=sqrt(g/l)//angular frequency
16 T=2*pi/w//time period
17 A=l*theta//linear amplitude
18 ohm=t*sqrt(theta^2-theta1^2)//angular speed at disp[
   lacement theta1
19 v=l*ohm//linear speed
20 alpha=theta*w^2//angular acceleration
21
22 printf('the time period of the pendululum is %3.2f s
   ',T)
23 printf('\nthe linear amplitude of the pendulum is %3
   .1f cm',A*10^2)
24 printf('\nthe linear speed of the pendulum at
   displacement of 0.02 rad is %3.1f cm/s',v*10^2)
25 printf('\nthe angular acceleration of the pendulum
   is %d rad s^-2',alpha)

```

Scilab code Exa 12.16w calculation of the time period of small oscillations

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.16w

```

```

5 //calculation of the time period of small
   oscillations
6
7 //given data
8 //h=R..... height equal to radius of the circle
9 g=%pi^2//gravitational acceleration(in m/s^2) of the
   earth
10 l=1//length(in m) of the string
11
12 //calculation
13 //at height R
14 //gdash = G*M/(R+R)^2 = g/4
15 gdash=g/4
16 T=2*%pi*sqrt(1/gdash)//time period
17
18 printf('The time period of small oscillations is %d
   s ',T)

```

Scilab code Exa 12.18w calculation of the time period of small oscillation about

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.18w
5 //calculation of the time period of small
   oscillation about the point of suspension
6
7 //given data
8 l=1//length(in m) of the stick
9 d=40*10^-2//distance(in m) of the centre from point
   of suspension
10 g=10//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation

```

```

13 //moment of inertia.....I = (m*l*l/12) + (m*d*d)
14 //time period.....T=2*%pi*sqrt(I/m*g*d)
15 //solving the above equations ,we get
16 T=2*%pi*sqrt(((l*l/12)+(d*d))/(g*d))
17
18 printf('the time period of small oscillation about
the point of suspension is %3.2f s',T)

```

Scilab code Exa 12.19w calculation of the moment of inertia of the second disc ab

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.19w
5 //calculation of the moment of inertia of the second
disc about the wire
6
7 //given data
8 I=0.2//moment of inertia(in kg-m^2) of the original
disc
9 T=2//time period(in s) of the oscillation of the
original disc
10 T1=2.5//time period(in s) of the oscillation of the
system of two discs
11
12 //calculation
13 //from equation of time period.....T = 2*%pi*sqrt(I
/K)
14 I1=((T1^2/T^2)*(I))-I//moment of inertia of the
second disc
15
16 printf('the moment of inertia of the second disc
about the wire is %3.2f kg-m^2',I1)

```

Scilab code Exa 12.22w calculation of the phase difference between the individual

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 12.22w
5 //calculation of the phase difference between the
   individual motions
6
7 //given data
8 //amplitudes of both the waves are same
9 //resultant amplitude is equal to individual
   amplitudes
10
11 //calculation
12 //the resultant amplitude is ..... $A = \sqrt{A^2 + A^2 + 2*A*A*\cos(\delta)}$ 
13 //on further solving ..... $A = 2*A*\cos(\delta/2)$ 
14 delta=2*(acosd(1/2))
15
16 printf('the phase difference between the individual
   motions is %d degree',delta)
```

Chapter 13

Fluid Mechanics

Scilab code Exa 13.1 calculation of the force exerted by the water on the bottom

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.1
5 //calculation of the force exerted by the water on
   the bottom
6
7 //given data
8 h=20*10^-2//height(in m) of the flask
9 r=10*10^-2//radius(in m) of the bottom of the flask
10 P0=1.01*10^5//atmospheric pressure(in Pa)
11 rho=1000//density of water(in kg/m^3)
12 g=10//gravitational acceleration(in m/s^2) of the
   earth
13
14 //calculation
15 P=P0+(h*rho*g)//pressure at the bottom
16 A=%pi*r^2//area of the bottom
17 F=P*A//force on the bottom
18
19 printf('the force exerted by the water on the bottom
```

is %d N',F)

Scilab code Exa 13.1w calculation of the force exerted by the mercury on the bottom

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.1w
5 //calculation of the force exerted by the mercury on
   the bottom of the beaker
6
7 //given data
8 h=10*10^-2//height(in m) of the mercury
9 r=4*10^-2//radius(in m) of the beaker
10 g=10//gravitational acceleration(in m/s^2) of the
   earth
11 P0=1*10^5//atmospheric pressure(in Pa)
12 rho=13600//density of mercury(in kg/m^3)
13
14 //calculation
15 P=P0+(h*rho*g)//pressure at the bottom
16 A=%pi*r^2//area of the bottom
17 F=P*A//force on the bottom
18
19 printf('the force exerted by the mercury on the
   bottom of the beaker is %d N',F)
```

Scilab code Exa 13.2 calculation of the volume of the cube outside the water

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.2
```

```

5 //calculation of the volume of the cube outside the
   water
6
7 //given data
8 m=700*10^-3//mass(in kg) of the cube
9 l=10*10^-2//length(in m) of the cube
10 rho=1000//density of water(in kg/m^3)
11
12 //calculation
13 V=m/rho//weight of displaced water = V*rho*g
14 Vtotal=l^3//total volume of the cube
15 Vout=Vtotal-V//volume of the cube outside the water
16
17 printf('the volume of the cube outside the water is
   %d cm^3 ',Vout*10^6)

```

Scilab code Exa 13.2w calculation of the height of the atmosphere to exert the same

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.2w
5 //calculation of the height of the atmosphere to
   exert the same pressure as at the surface of the
   earth
6
7 //given data
8 P0=1*10^5//atmospheric pressure(in Pa)
9 rho=1.3//density of air(in kg/m^3)
10 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
11
12 //calculation
13 h=P0/(g*rho)
14

```

```
15 printf('the height of the atmosphere to exert the
    same pressure as at the surface of the earth is
    %d m',round(h))
```

Scilab code Exa 13.3 calculation of the speed of the outgoing liquid

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.3
5 //calculation of the speed of the outgoing liquid
6
7 //given data
8 A1=1*10^-4//area(in m^2) of the inlet of the tube
9 A2=20*10^-6//area(in m^2) of the outlet of the tube
10 v1=2//speed(in cm/s) of the ingoing liquid
11
12 //calculation
13 v2=A1*v1/A2//equation of continuity
14
15 printf('the speed of the outgoing liquid is %d cm/s'
    ,v2)
```

Scilab code Exa 13.3w calculation of the height of the water coloumn

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.3w
5 //calculation of the height of the water coloumn
6
7 //given data
8 h1=2*10^-2//difference in the height(in m)
```

```

9 s=13.6//specific gravity of mercury
10
11 //calculation
12 //P = P0 + (h*rho*g).....using this equation
13 h=h1*s//height of the water coloumn
14
15 printf('the height of the water coloumn is %d cm',h
        *10^2)

```

Scilab code Exa 13.4 calculation of the difference in the pressures at A and B point

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.4
5 //calculation of the difference in the pressures at
   A and B point
6
7 //given data
8 A1=1*10^-4//area(in m^2) at point A of the tube
9 A2=20*10^-6//area(in m^2) at point B of the tube
10 v1=10*10^-2//speed(in m/s) of the ingoing liquid
11 rho=1200//density of the liquid(in kg/m^3)
12
13 //calculation
14 v2=A1*v1/A2//equation of continuity
15 //by Bernoulli equtation..... $P_1 + (\rho * g * h_1) + (\rho * v_1^2 / 2) = P_2 + (\rho * g * h_2) + (\rho * v_2^2 / 2)$ 
16 deltaP=(1/2)*rho*(v2^2-v1^2)
17
18 printf('the difference in the pressures at A and B
        point is %d Pa',deltaP)

```

Scilab code Exa 13.5 calculation of the speed of the water coming out of the tap

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.5
5 //calculation of the speed of the water coming out
   of the tap
6
7 //given data
8 h=6//depth(in m) of the tap
9 g=9.8//gravitational acceleration(in m/s^2) of the
   earth
10
11 //calculation
12 v=sqrt(2*g*h)//torricelli 's theorem
13
14 printf('the speed of the water coming out of the tap
   is %d m/s ',round(v))
```

Scilab code Exa 13.5w calculation of the force applied on the water in the thicker

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.5w
5 //calculation of the force applied on the water in
   the thicker arm
6
7 //given data
8 A1=1*10^-4//area(in m^2) of arm 1
9 A2=10*10^-4//area(in m^2) of arm 2
10 f=5//force(in N) applied on the water in the thinner
   arm
11
```

```

12 //calculation
13 //P = P0 + (h*rho*g).....using this equation
14 F=f*A2/A1//force applied on the water in the thicker
    arm
15
16 printf('the force applied on the water in the
    thicker arm is %d N',F)

```

Scilab code Exa 13.6w calculation of the elongation of the spring

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.6w
5 //calculation of the elongation of the spring
6
7 //given data
8 m=10*10^-3//mass(in kg) of the copper piece
9 l=1*10^-2//elongation(in m) in the spring
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11 rho=9000//density of copper(in kg/m^3)
12 rho0=1000//density of water(in kg/m^3)
13
14 //calculation
15 k=m*g/l//spring constant
16 V=m/rho//volume of copper
17 Fb=V*rho0*g//force of buoyancy
18 x=((k*l)-Fb)/k//elongation of the spring
19
20 printf('the elongation of the spring is %3.2f cm',x
    *10^2)

```

Scilab code Exa 13.7w calculation of the maximum weight that can be put on the block

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.7w
5 //calculation of the maximum weight that can be put
   on the block without wetting it
6
7 //given data
8 l=3*10^-2//length(in m) of the edge of the cubical
   block
9 rho=800//density of wood(in kg/m^3)
10 k=50//spring constant(in N/m)
11 g=10//gravitational acceleration(in m/s^2) of the
   earth
12 rho0=1000//density of water(in kg/m^3)
13
14 //calculation
15 s=rho/rho0//specific gravity
16 hin=l*s//height inside water
17 hout=l-hin//height outside water
18 V=l^3//volume of the block
19 Fb=V*rho0*g//force of buoyancy
20 Fs=k*hout//force exerted by the spring
21 Wdash=V*rho*g//weight of the block
22 W=Fb+Fs-Wdash//maximum weight
23
24 printf('the maximum weight that can be put on the
   block without wetting it is %3.2f N',W)
```

Scilab code Exa 13.8w calculation of the angle that the plank makes with the vertical

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
```

```

3  clc;clear;
4  //example 13.8w
5  //calculation of the angle that the plank makes with
    the vertical in equilibrium
6
7  //given data
8  l=1//length(in m) of the plank
9  h=0.5//height(in m) of the water level in the tank
10 s=0.5//specific gravity of the plank
11
12 //calculation
13 //A = OC/2 = l/(2*cosd(theta))
14 // mg = 2*l*rho*g
15 //buoyant force Fb=(2*l*rho*g)/cosd(theta)
16 //m*g*(OB)*sind(theta) = F(OA)*sind(theta)
17 theta=acosd(sqrt(1/2))
18
19 printf('the angle that the plank makes with the
    vertical in equilibrium is %d degree',theta)

```

Scilab code Exa 13.10w calculation of the rate of water flow through the tube

```

1  //developed in windows XP operating system 32bit
2  //platform Scilab 5.4.1
3  clc;clear;
4  //example 13.10w
5  //calculation of the rate of water flow through the
    tube
6
7  //given data
8  A1=30//area(in cm^2) of the tube at point A
9  A2=15//area(in cm^2) of the tube at point B
10 deltaP=600//change in pressure(in N/m^2)
11 rho0=1000//density of the water(in kg/m^3)
12

```

```

13 //calculation
14 r=A1/A2//ratio of area
15 //from equation of continuity vB/vA = A1/A2 = r = 2
16 //by Bernoulli equation.....P1 + (rho*g*h1) + (rho*
    v1^2/2) = P2 + (rho*g*h2) + (rho*v2^2/2)
17 //take vB = vA*2
18 vA=sqrt(deltaP*(r/(r+1))*(1/rho0))
19 Rflow=vA*A1//rate of water flow
20
21 printf('the rate of water flow through the tube is
    %d cm^3/s',Rflow*10^2)

```

Scilab code Exa 13.11w calculation of the velocity of the water coming out of the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 13.11w
5 //calculation of the velocity of the water coming
    out of the opening
6
7 //given data
8 AA=.5//area(in m^2) of the tank
9 AB=1*10^-4//area(in m^2) of the cross section at the
    bottom
10 m=20//mass(in kg) of the load
11 h=50*10^-2//height(in m)of the water level
12 g=10//gravitational acceleration(in m/s^2) of the
    earth
13 rho=1000//density of the water(in kg/m^3)
14
15 //calculation
16 //from the equation.....P = P0 + (h*rho*g)//
    pressure at the bottom
17 r=m*g/AA//in above equation it is the value of (h*

```

```

    rho*g)
18 //on solving ,we get .....PA = P0 + (400 N/m^2)
19 //from Bernoulli equtation .....P1 + (rho*g*h1) + (
    rho*v1^2/2) = P2 + (rho*g*h2) + (rho*v2^2/2)
20 //we get
21 vB=sqrt((2*(r+(rho*g*h)))/rho)
22
23 printf('the velocity of the water coming out of the
    opening is %3.1f m/s',vB)

```

Chapter 14

Some Mechanical Properties of Matter

Scilab code Exa 14.1 calculation of the tensile stress developed in the wire

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.1
5 //calculation of the tensile stress developed in the
   wire
6
7 //given data
8 m=4//mass(in kg) of the load
9 r=2*10^-3//radius(in m) of the wire
10 g=3.1*%pi//gravitational acceleration(in m/s^2) of
   the earth
11
12 //calculation
13 F=m*g//gravitational force
14 A=%pi*r^2//area
15 St=F/A//tensile stress
16
17 printf('the tensile stress developed in the wire is
```

```
%3.1 e N/m2 ',St)
```

Scilab code Exa 14.1w calculation of the extension of the wire

```
1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 14.1w
5 //calculation of the extension of the wire
6
7 //given data
8 L=2//lengh(in m)of the wire
9 A=.2*10-4//area(in m2)
10 m=4.8//mass(in kg)
11 Y=2*1011//Young modulus of steel
12 g=10//gravitational acceleration(in m/s2) of the
    earth
13
14 //calculation
15 T=m*g//weight
16 l=(T*L)/(A*Y)//exension
17
18 printf('the extension of the wire is %3.1e m',l)
```

Scilab code Exa 14.2 calculation of the value of Young modulus

```
1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 14.2
5 //calculation of the value of Young modulus
6
7 //given data
```

```

8 m=4//mass(in kg) of the load
9 l=20//length(in m) of the steel wire
10 r=2*10^-3//radius(in m) of the steel wire
11 dl=.031*10^-3//increase in the length(in m)
12 g=3.1*%pi//gravitational acceleration(in m/s^2) of
    the earth
13
14 //calculation
15 Ssl=(m*g)/(%pi*r^2)//longitudinal stress
16 Stl=dl/l//longitudinal strain
17 Y=Ssl/Stl//Young modulus
18
19 printf('the value of Young modulus is %3.1e N/m^2',Y
    )

```

Scilab code Exa 14.2w calculation of the elongation of the rope and corresponding

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.2w
5 //calculation of the elongation of the rope and
    corresponding change in the diameter
6
7 //given data
8 L=4.5//length(in m) of the nylon rope
9 d=6*10^-3//diameter(in m) of the nylon rope
10 T=100//weight(in N) of the monkey
11 Y=4.8*10^11//Young modulus(in N/m^2) of the rope
12 Pr=.2//Poisson ratio of nylon
13
14 //calculation
15 A=%pi*(d/2)^2//area of cross section
16 l=(T*L)/(A*Y)//elongation
17 deltd=(Pr*l*d)/(L)//change in diameter

```

```

18
19 printf('the elongation of the rope is %3.2e m',l)
20 printf('\nthe corresponding change in the diameter
    is %3.1e m',deltad)

```

Scilab code Exa 14.3 calculation of the elastic potential energy stored in the str

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.3
5 //calculation of the elastic potential energy stored
    in the stretched steel wire
6
7 //given data
8 l=2//length(in m) of the steel wire
9 A=4*10^-6//cross sectional area(in m^2) of the steel
    wire
10 dl=2*10^-3//increase in the length(in m)
11 Y=2*10^11//Young modulus(in N/m^2)
12
13 //calculation
14 St=dl/l//strain in the wire
15 Ss=Y*St//stress in the wire
16 V=A*l//volume of the steel wire
17 U=Ss*St*V/2
18
19 printf('the elastic potential energy stored in the
    stretched steel wire is %3.1f J',U)

```

Scilab code Exa 14.3w calculation of the minimum radius of the wire used if it is

```

1 //developed in windows XP operating system 32bit

```



```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.3w
5 //calculation of the minimum radius of the wire used
   if it is not to break
6
7 //given data
8 m1=1//mass(in kg) of block1
9 m2=2//mass(in kg) of block2
10 Ss=2*10^9//breaking stress(in N/m^2) of the metal
11 g=10//gravitational acceleration(in m/s^2) of the
   earth
12
13 //calculation
14 //using equation ....stress = tension / Area of
   cross section
15 //T - (m1*g) = m1 * a.....(1)
16 //(m2*g) - T = m2*a.....(2)
17 //Adding equation (1) and equation (2),we get
18 a=((m2*g)-(m1*g))/(m1+m2)
19 T=(m1*g)+(m1*a)//tension in the string from equation
   (1)
20 r=sqrt(T/(Ss*%pi))//radius
21
22 printf('the minimum radius of the wire used if it is
   not to break is %3.1e m',r)

```

Scilab code Exa 14.4 calculation of the force by which the surface on one side of

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.4
5 //calculation of the force by which the surface on
   one side of the diameter pulls the surface on the

```

```

        other side
6
7 //given data
8 r=5*10^-2//radius(in m) of the beaker
9 S=.075//surface tension(in N/m) of the water
10
11 //calculation
12 l=2*r//length of diameter of the surface
13 F=S*l//force
14
15 printf('the force by which the surface on one side
        of the diameter pulls the surface on the other
        side is %3.1e N',F)

```

Scilab code Exa 14.4w calculation of the ratio of the lengths of the two wire

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.4w
5 //calculation of the ratio of the lengths of the two
  wire
6
7 //given data
8 Ys=2*10^11//Young modulus(in N/m^2) of the steel
  wire
9 Yc=1.1*10^11//Young modulus(in N/m^2) of the copper
  wire
10
11 //calculation
12 //r = Ls/Lc.....required ratio
13 r=Ys/Yc//required ratio
14
15 printf('the ratio of the lengths of the two wire(Ls/
  Lc) is %f:1',r)

```

Scilab code Exa 14.5 calculation of the gain in the surface energy

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.5
5 //calculation of the gain in the surface energy
6
7 //given data
8 R=10^-2//radius(in m) of the drop
9 n=1000//number of droplets formed
10 S=.075//surface tension(in N/m) of the water
11
12 //calculation
13 //volume of original drop = total volume of all
    droplets formed
14 r=R/n^(1/3)//radius of each droplet
15 A1=4*pi*R^2//surface area of drop
16 A2=n*(4*pi*r^2)//surface area of each droplet
17 deltaA=A2-A1//change in surface area
18 deltaU=deltaA*S//change in surface energy
19
20 printf('the gain in the surface energy is %3.1e J',
    deltaU)
```

Scilab code Exa 14.5w calculation of the decrease in the volume of the sample of w

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.5w
```

```

5 //calculation of the decrease in the volume of the
   sample of water
6
7 //given data
8 V1=1000*10^-6//initial volume(in m^3)
9 P1=10^5//initial pressure(in N/m^2)
10 P2=10^6//final pressure(in N/m^2)
11 C=50*10^-11//compressibility(in m^2/N)of the water
12
13 //calculation
14 deltap=P2-P1//change in pressure
15 //compressibility = 1/Bulk modulus = -(deltaV/V)/
   deltaP
16 deltaV=-(C*deltap*V1)
17
18 printf('the decrease in the volume of the sample of
   water is %3.2f cm^3',-deltaV*10^6)

```

Scilab code Exa 14.6 calculation of the excess pressure inside a mercury drop

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.6
5 //calculation of the excess pressure inside a
   mercury drop
6
7 //given data
8 R=2*10^-3//radius(in m) of the drop
9 S=.464//surface tension(in N/m) of the drop
10
11 //calculation
12 deltaP=2*S/R//excess pressure
13
14 printf('the excess pressure inside a mercury drop is

```

```
%d N/m2',deltaP)
```

Scilab code Exa 14.6w calculation of the longitudinal strain in two wires

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 14.6w
5 //calculation of the longitudinal strain in two
   wires
6
7 //given data
8 m1=1//mass(in kg) of load 1
9 m2=2//mass(in kg) of load 2
10 A=.005*10-4//area(in m2) of the cross section
11 Y=2*1011///Young modulus(in N/m2) of the wire
12 g=10//gravitational acceleration(in m/s2) of the
   earth
13
14 //calculation
15 T1=m1*g//tension in wire 1
16 Ss1=T1/A//longitudinal stress
17 St1=Ss1/Y//longitudinal strain
18 T2=(m2*g)+T1//tension in wire 2
19 Ss2=T2/A//longitudinal stress
20 St2=Ss2/Y//longitudinal strain
21
22 printf('the longitudinal strain in wire 1 is %3.1e',
   St1)
23 printf('\\nthe longitudinal strain in wire 2 is %3.1e
   ',St2)
```

Scilab code Exa 14.7 calculation of the density of the liquid

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.7
5 //calculation of the density of the liquid
6
7 //given data
8 h=.02*10^-2//height(in m) of the column of liquid
9 R=7.5*10^-3//radius(in m) of the soap bubble
10 S=.03//surface tension(in N/m) of the soap solution
11 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 deltaP=4*S/R//excess pressure inside the soap bubble
15 rho=deltaP/(h*g)//densiy
16
17 printf('the density of the liquid is %3.1e kg/m^3',
    rho)

```

Scilab code Exa 14.7w calculation of the longitudinal strain developed in each wire

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.7w
5 //calculation of the longitudinal strain developed
    in each wire
6
7 //given data
8 m=3//mass(in kg) of each block
9 A=.005*10^-4//area(in m^2) of the cross section
10 Y=2*10^11//Young modulus(in N/m^2) of the wire
11 g=10//gravitational acceleration(in m/s^2) of the
    earth

```

```

12
13 //calculation
14 //using equation of motion ,
15 //TA = m*a.....(1)
16 //TB - TA = m*a.....(2)
17 //m*g - TB = m*a.....(3)
18 //adding equation (2) and equation (3) and
    substituting TA from equation (1),we get
19 a=(m*g)/(3*m)//acceleration
20 TA=m*a//Tension(in N) in wire A
21 TB=(m*a)+TA//Tension(in N) in wire B..from equation
    (2)
22 StA=(TA)/(A*Y)//longitudinal strain in wire A
23 StB=(TB)/(A*Y)//longitudinal strain in wire B
24
25 printf('the longitudinal strain developed in wire A
    is %3.1e',StA)
26 printf('\nthe longitudinal strain developed in wire
    B is %3.1e',StB)

```

Scilab code Exa 14.8 calculation of the height of the water in the column

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.8
5 //calculation of the height of the water in the
    column
6
7 //given data
8 r=.2*10^-3//radius(in m) of the tube
9 S=.075//surface tension(in N/m) of the water
10 g=10//gravitational acceleration(in m/s^2) of the
    earth
11 rho=1000//density of the water(in kg/m^3)

```

```

12 theta=0//tube dipped vertically
13
14 //calculation
15 h=(2*S*cosd(theta))/(r*rho*g)//height in column
16
17 printf('the height of the water in the column is %3
    .1 f cm',h*10^2)

```

Scilab code Exa 14.8w calculation of the elastic potential energy stored in the w

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.8w
5 //calculation of the elastic potential energy stored
    in the wire
6
7 //given data
8 A=3*10^-6//area(in m^2) of the cross section
9 l=50*10^-2//natural length(in m)
10 m=2.1//mass(in kg) hanged
11 Y=1.9*10^11///Young modulus(in N/m^2) of the wire
12 g=10//gravitational acceleration(in m/s^2) of the
    earth
13
14 //calculation
15 V=A*l//volume of the wire
16 T=m*g//tension in the wire
17 Ss=T/A//stress
18 St=Ss/Y//strain
19 U=(Ss*St*V/2)//elastic potential energy
20
21 printf('the elastic potential energy stored in the
    wire is %3.1e J',U)

```

Scilab code Exa 14.9 calculation of the value of the coefficient of viscosity of t

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.9
5 //calculation of the value of the coefficient of
  viscosity of the solution
6
7 //given data
8 d=2*10^-3//diameter(in m) of the air bubble
9 sigma=1750//density(in kg/m^3) of the solution
10 v=.35*10^-2//rate of flow(in m/s)
11 g=9.8//gravitational acceleration(in m/s^2) of the
  earth
12
13 //calculation
14 r=d/2//radius of the air bubble
15 //force of buoyancy is ..... $B = (4/3)*\pi*r^3*$ 
  sigma*g
16 //viscous force is ..... $F = 6*\pi*\eta*r*v$ 
17 //above two forces are equal,thus we get
18 eta=(2*r^2*sigma*g)/(9*v)//coefficient of viscosity
19
20 printf('the value of the coefficient of viscosity of
  the solution is %d poise',round(eta*10))//0 1
  poise = .1 N-s/m^2
```

Scilab code Exa 14.9w calculation of the elongation of the wire

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
```

```

3  clc;clear;
4  //example 14.9w
5  //calculation of the elongation of the wire
6
7  //given data
8  W=10//weight(in N) of the block
9  A=3*10-6//area(in m2) of the cross section
10 r=20*10-2//radius(in m) of the circle of rotation
11 v=2//speed(in m/s) of the block
12 Y=2*1011//Young modulus(in N/m2) of the wire
13 g=10//gravitational acceleration(in m/s2) of the
    earth
14
15 //calculation
16 m=W/g//mass of the block
17 T=W+(m*v*v/r)//tension
18 L=r
19 l=(T*L)/(A*Y)//elongation
20
21 printf('the elongation of the wire is %3.1e cm',l
    *102)

```

Scilab code Exa 14.11w calculation of the amount by which the pressure inside the

```

1  //developed in windows XP operating system 32bit
2  //platform Scilab 5.4.1
3  clc;clear;
4  //example 14.11w
5  //calculation of the amount by which the pressure
    inside the bubble is greater than the atmospheric
    pressure
6
7  //given data
8  r=1*10-3//radius(in m) of the air bubble
9  S=.075//surface tension(in N/m)

```

```

10 rho=1000//density(in kg/m^3) of the liquid
11 h=10*10^-2//depth(in m) of the bubble
12 g=9.8//gravitational acceleration(in m/s^2) of the
    earth
13
14 //calculation
15 //P = P0 +(h*rho*g).....(1)
16 //Pdash = P + (2*S/r).....(2)
17 //deltaP = Pdash - P0
18 deltaP=(h*rho*g)+(2*S/r)//difference in the pressure
19
20 printf('the pressure inside the bubble is greater
    than the atmospheric pressure by %d Pa',deltaP)

```

Scilab code Exa 14.12w calculation of the load W suspended from wire to keep it in equilibrium

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.12w
5 //calculation of the load W suspended from wire to
    keep it in equilibrium
6
7 //given data
8 l=10*10^-2//length(in m) of the wire
9 //1 dyne = 10^-5 N
10 S=25*10^-5*10^2//surface tension(in N/m) of the soap
    solution
11 g=10//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 F=2*l*S//force exerted by the film on the wire
15 m=F/g//mass of the load
16

```

```

17 printf('the load W suspended from wire to keep it in
    equilibrium should be %3.1e N',F)
18 printf('\nthe mass of the load suspended from wire
    to keep it in equilibrium should be %3.1e kg or
    %3.1f g',m,m*10^3)

```

Scilab code Exa 14.13w calculation of the radius of the capillary tube

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.13w
5 //calculation of the radius of the capillary tube
6
7 //given data
8 h=7.5*10^-2//height(in m) by which the capillary
    rises
9 S=7.5*10^-2//surface tension(in N/m) of water
10 theta=0//contact angle(in degree) between water and
    glass
11 g=10//gravitational acceleration(in m/s^2) of the
    earth
12 rho=1000//density(in kg/m^3) of water
13
14 //calculation
15 r=(2*S*cosd(theta))/(h*rho*g)//from formula of
    height in capillary tube
16
17 printf('the radius of the capillary tube is %3.1f mm
    ',r*10^3)

```

Scilab code Exa 14.15w calculation of the tangential force needed to keep the pla

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.15w
5 //calculation of the tangential force needed to keep
   the plate moving
6
7 //given data
8 A=10//area(in m^2) of the plate
9 v=2//speed(in m/s) of the plate
10 d=1//depth(in m) of the river
11 // 1 poise = .1 N-s/m^2...unit of viscosity
12 eta=10^-2*10^-1//coefficient of viscosity(in N-s/m
   ^2)
13
14 //calculation
15 dvbydx=v/d//velocity gradient
16 F=eta*dvbydx*A//force exerted
17
18 printf('the tangential force needed to keep the
   plate moving is %3.2f N',F)

```

Scilab code Exa 14.16w calculation of the shearing stress between the horizontal

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.16w
5 //calculation of the shearing stress between the
   horizontal layers of water
6
7 //given data
8 v=18*10^3/(60*60)//velocity(in m/s) of the water in
   river
9 d=5//depth(in m) of the river

```

```

10 // 1 poise = 0.1 N-s/m^2
11 eta=10^-2*10^-1//coefficient of viscosity(in N-s/m
    ^2) of the water
12
13 //calculation
14 dvbydx=v/d//velocity gradient
15 //force of viscosity .....F=eta*A*(dvbydx)
16 //shearing stress ..... Ss=F/A
17 Ss=eta*(dvbydx)
18
19 printf('the shearing stress between the horizontal
    layers of water is %3.1e N/m^2',Ss)

```

Scilab code Exa 14.17w calculation of the terminal velocity of the rain drop

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 14.17w
5 //calculation of the terminal velocity of the rain
    drop
6
7 //given data
8 r=.01*10^-3//radius(in m) of the drop
9 eta=1.8*10^-5//coefficient of viscosity(in N-s/m^2)
    of the air
10 rho=1.2//density(in kg/m^3) of the air
11 rho0=1000//density(in kg/m^3) of the water
12 g=10//gravitational acceleration(in m/s^2) of the
    earth
13
14 //calculation
15 //at terminal velocity ..... $6 * \pi * \eta * r * v = (4/3) * \pi * r^3 * \rho * g$ 
16 v=(2*r^2*rho0*g)/(9*eta)//terminal velocity

```

```
17
18 printf('the terminal velocity of the rain drop is %3
    .1f cm/s^2',v*10^2)
```

Chapter 15

Wave Motion and Waves on a String

Scilab code Exa 15.1 calculation of the velocity function $f(t)$ giving displacement f

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.1
5 //calculation of the velocity ,function  $f(t)$  giving
   displacement ,function  $g(x)$  giving shape
6
7 //given data
8 // $y = y_0 \exp -(((t/T) - (x/\lambda))^2)$ 
9  $y_0=4*10^{-3}$  //value of  $y_0$ (in m)
10  $T=1$  //value of  $T$ (in s)
11  $\lambda=4*10^{-2}$  //value of  $\lambda$ (in m)
12
13 //calculation
14  $v=\lambda/T$  //velocity of the wave
15 //by putting  $x=0$  in equation (1) .....  $f(t) = y_0 \exp$ 
    $-((t/T)^2)$ 
16 //by putting  $t=0$  in equation (1) .....  $g(x) = y_0 \exp$ 
    $-((x/\lambda)^2)$ 
```



```

17
18 printf('the velocity of the wave is %d cm/s',v*10^2)
19 printf('\nthe function f(t) giving displacement is
    ..... f(t) = y0*exp-((t/T)^2)')
20 printf('\nthe function g(x) giving shape of the
    string at t=0 is ..... g(x) = y0*exp-((x/lambda)^2)
    ')

```

Scilab code Exa 15.1w calculation of the amplitude wavelength frequency speed of

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.1w
5 //calculation of the amplitude ,wavelength ,frequency ,
    speed of the wave
6
7 //given data
8 //given wave equation is .....y = (3.0cm)*sin
    (6.28(.50*x - 50*t))
9
10 //calculation
11 //comparing with standard equation of wave....y = A*
    sin*2*%pi*((x/lambda) - (t/T)),we get
12 A=3//amplitude(in cm)
13 lambda=(1/0.50)//wavelength(in cm)
14 T=1/50//time period(in s)
15 nu=1/T//frequency(in Hz)
16 v=nu*lambda//wave velocity(in cm s^-1)
17
18 printf('the amplitude is %d cm',A)
19 printf('\nthe wavelength is %d cm',lambda)
20 printf('\nthe frequency is %d Hz',nu)
21 printf('\nthe wave velocity is %d cm/s',v)

```

Scilab code Exa 15.2 calculation of the amplitude wave number wavelength frequency

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.2
5 //calculation of the amplitude ,wave number ,
   wavelength ,frequency ,time period ,wave velocity
6
7 //given data
8 //given equation .....y = (5mm)*sin((1cm-1)*x - (60
   s-1)*t)
9 w=60//angular frequency
10
11 //calculation
12 A=5//amplitude(in cm)
13 k=1//wave number(in cm-1)
14 lambda=(2*%pi)/k//wavelength(in cm)
15 nu=w/(2*%pi)//frequency(in Hz)
16 T=1/nu//Time period(in s)
17 v=nu*lambda//wave velocity(in cm/s)
18
19 printf('the amplitude is %d mm',A)
20 printf('\\nthe wave number is %d cm-1',k)
21 printf('\\nthe wavelength is %3.2 f cm',lambda)
22 printf('\\nthe frequency is %3.2 f Hz',nu)
23 printf('\\nthe time period is %3.2 f s',T)
24 printf('\\nthe wave velocity is %d cm/s',v)
```

Scilab code Exa 15.2w calculation of the maximum velocity and acceleraion of the p

```
1 //developed in windows XP operating system 32bit
```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.2w
5 //calculation of the maximum velocity and
   acceleration of the particle
6
7 //given data
8 //given wave equation is .....y = (3.0cm)*sind((3.14
   cm^-1)x - (3.14 s^-1)*t))
9 t=0//time taken(in s)
10 t1=.11//time(in s) for acceleration
11 function yv=f(t)
12     yv = (3.0)*sin(-(3.14)*t)//take x=0 (after
   derivative )..for maximum velocity
13 endfunction
14
15 //calculation
16 //V = dy/dt
17 vmax=derivative(f,t)
18 //vn=(-9.4)*(314)*(sin((3.14*x)+(314*t))) ..... take
   x=6(after derivative)...for acceleration at x=6
   cm
19 a=- (2952)*sin(6*%pi-11*%pi)
20
21 printf('the maximum velocity is %3.2f m/s',vmax)
22 printf('\\nthe acceleration at t=0.11 s and x= 6 cm
   is %d cm^2/s',a)

```

Scilab code Exa 15.3 calculation of the time taken by the pulse in travelling thro

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.3
5 //calculation of the time taken by the pulse in

```

```

        travelling through a distance
6
7 //given data
8 m=1//mass(in kg) of the block
9 mu=1*10^-3*10^2//mass density(in kg/m)
10 l=50*10^-2//disatnce(in m) travelled
11 g=10//gravitational acceleration(in m/s^2) of the
    earth
12
13 //calculation
14 F=m*g//tension in the string
15 v=sqrt(F/mu)//wave velocity
16 T=1/v//time taken
17
18 printf('the time taken by the pulse in travelling
    through a distance of 50 cm is %3.2f s',T)

```

Scilab code Exa 15.3w calculation of the speed and displacement of the particle

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.3w
5 //calculation of the speed and displacement of the
    particle
6
7 //given data
8 A=.80*10^-6//area(im m^2) of the string
9 rho=12.5*10^-3*10^6//density(in kg/m^3)
10 nu=20//transverse frequency(in Hz)
11 F=64//tension(in N)
12
13 //calculation
14 mu=A*1*rho//mass of 1 m of the string = linear mass
    density

```

```

15 v=sqrt(F/mu)//wave speed
16 w=2*pi*nu//angular velocity
17 //substituting above values equation becomes.....y =
    (1.0cm)*cos(125*(t-(x/v)))
18
19 function y=f(t,x)
20     y=1*cos(2*pi*nu*(t-(x/v)))
21 endfunction
22 t=0.05//time taken(in s)
23 x=50*10^-2 //displacement(in m)
24 yn=f(t,x)
25
26 function yfv=ffv(t)
27     yfv=1*cos(2*pi*nu*(t-((50*10^-2)/v)))//putting
        value of x ..to be substituted after
        derivation
28 endfunction
29 vn=derivative(ffv,t)
30
31 printf('the wave speed is %d m/s',v)
32 printf('\nthe wave equation is .....y = (1.0cm)*
    cos(%d*(t-(x/%d)))',w,v)
33 printf('\nthe displacement of the particle at x=50
    cm at time t=0.05 s is %3.2f cm',yn)
34 printf('\nthe velocity of the particle at that
    position is %d cm/s',round(vn))

```

Scilab code Exa 15.4 calculation of the power transmitted through a given point

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.4
5 //calculation of the power transmitted through a
    given point

```

```

6
7 //given data
8 P1=.20//average power(in W)
9 A1=2//amplitude(in mm) at this point
10 A2=3//amplitude(in mm)
11
12 //calculation
13 //transmitted power is proportional to the square of
    the amplitude
14 P2=P1*(A2/A1)^2
15
16 printf('the power transmitted through the given
    point is %3.2f W',P2)

```

Scilab code Exa 15.4w calculation of the extension of the wire over its natural length

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.4w
5 //calculation of the extension of the wire over its
    natural length
6
7 //given data
8 m=5*10^-3//mass(in kg) of the wire
9 L=50*10^-2//length(in cm) of the wire
10 v=80//speed(in m/s) of the wave
11 Y=16*10^11//Young modulus(in N/m^2)
12 A=1*10^-6//area(in m^2) of cross section of the wire
13
14 //calculation
15 mu=m/L//linear mass density
16 F=mu*v^2//tension in the wire
17 deltaL=(F*L)/(A*Y)//extension in the length of wire
18

```

```

19 printf('the extension of the wire over its natural
    length is %3.2f mm',deltaL*10^3)

```

Scilab code Exa 15.5 calculation of the phase difference between the waves and amp

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.5
5 //calculation of the phase difference between the
    waves and amplitude of the resultant wave
6
7 //given data
8 //equations of the wave are
9 //y1 = A1*sin(k(x-v*t)) .....(1)
10 //y2 = A2*sin(k(x-v*t+x0)) .....(2)
11 k=6.28*10^2//wave number(in m^-1)
12 x0=1.50*10^-2//value of x0(in m)
13 A1=5*10^-3//amplitude(in m) of wave 1
14 A2=4*10^-3//amplitude(in m) of wave 2
15
16 //calculation
17 deltaP=k*x0//phase difference
18 deltaA=abs(A1-A2)//amplitude of the wave
19
20 printf('the phase difference between the waves is %3
    .2f rad',deltaP)
21 printf('\nthe amplitude of the resultant wave is %3
    .1f mm',deltaA*10^3)

```

Scilab code Exa 15.5w calculation of the wavelength of the pulse when it reaches

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.5w
5 //calculation of the wavelength of the pulse when it
   reaches the top of the rope
6
7 //given data
8 lr=12//length(in m) of the rope
9 mr=6//mass(in kg) of the rope
10 mb=2//mass(in kg) of the block
11 lambda=.06//wavelength(in m) of the wave produced at
   the lower end
12
13 //calculation
14 //from equation ..... $v = nu*\lambda$ 
15 //putting  $v = \sqrt{F/\lambda}$ ....we get
16 //sqrt(F/lambda) = nu*sqrt(mu).... using this
   equation ,we get
17 lambda1=lambda*sqrt((mr+mb)/mb)
18
19 printf('the wavelength of the pulse when it reaches
   the top of the rope is %3.2f m',lambda1)

```

Scilab code Exa 15.6 calculation of the velocity node closest to origin antinode c

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.6
5 //calculation of the velocity ,node closest to origin
   ,antinode closest to origin ,amplitude at x
6
7 //given data
8 //equation of the wave is ..... $y = A*\cosd(k*x)*sind(
   w*t)$ 

```



```

 9 A=1//amplitude(in mm)
10 k=1.57//value of k(in cm-1)
11 w=78.5//angular velocity(in s-1)
12 x=2.33//value of x(in cm)
13
14 //calculation
15 v=w/k//wave velocity
16 xn=%pi/(2*k)//for a node ... cosd(kx) = 0
17 xa=%pi/k//for a antinode ... |cosd(kx)| = 1
18 Ar=A*abs(cos(k*x))
19
20 printf('the velocity of the wave is %d cm/s',v)
21 printf('\nthe node closest to the origin is located
    at x=%d cm',xn)
22 printf('\nthe antinode closest to the origin is
    located at x=%d cm',xa)
23 printf('\nthe amplitude at x=2.33 is %3.2f mm',Ar)

```

Scilab code Exa 15.6w calculation of the displacement of the particle

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.6w
5 //calculation of the displacement of the particle
6
7 //given data
8 //given equations are
9 //y1 = (1.0 cm)*sin((3.14 cm-1)*x - (157 s-1)*t)
    .....(1)
10 //y2 = (1.5 cm)*sin((1.57 cm-1)*x - (314 s-1)*t)
    .....(2)
11
12 //calculation
13 function y1=f1(t,x)

```

```

14     y1=1*sin((3.14*x)-(157*t))
15 endfunction
16
17 function y2=f2(t,x)
18     y2=1.5*sin((1.57*x)-(314*t))
19 endfunction
20
21 x=4.5//given value of x(in cm)
22 t=5*10^-3//given value of t(in s)
23 //y = y1 + y2.....net displacement
24 y=f1(t,x)+f2(t,x)
25
26 printf('the displacement of the particle at x=4.5 cm
        and t=5.0 ms is %3.2f cm',y)

```

Scilab code Exa 15.7 calculation of the fundamental frequency of the portion of the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.7
5 //calculation of the fundamental frequency of the
  portion of the string between the wall and the
  pulley
6
7 //given data
8 m=1.6//mass(in kg) of the load
9 mw=20*10^-3//mass(in kg) of the wire
10 l=50*10^-2//length(in kg/m) of wire
11 g=10//gravitational acceleration(in m/s^2) of the
  earth
12 L=40*10^-2//length(in m) of the string between the
  wall and the pulley
13
14 //calculation

```

```

15 F=m*g//tension in the string
16 mu=mw/l//linear mass density
17 nu0=(1/(2*L))*sqrt(F/mu)//fundamental frequency
18
19 printf('the fundamental frequency of the portion of
    the string between the wall and the pulley is %d
    Hz',nu0)

```

Scilab code Exa 15.7w calculation of the maximum displacement wavelengths and wave

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.7w
5 //calculation of the maximum displacement,
    wavelengths and wave speed, velocity, nodes and
    antinodes, number of loops
6
7 //given data
8 //given equation is ....y = (5.00 mm)*sin(1.57 cm
    ^-1)*sin((314 s^-1)*t)
9
10 //calculation
11 //at x=5.66 cm
12 A=(5*10^-3)*sin(1.57*5.66)//amplitude
13 k=1.57//value of k(in cm^-1)
14 w=314//angular frequency(in s^-1)
15 lambda=(2*pi)/k//wavelength
16 nu=(w)/(2*pi)//frequency
17 //v = dy/dt = (157 cm/s)*sin(1.57 cm^-1*x)*cos((314
    s^-1)*t)
18 function v=f(t,x)
19     v=157*sin(1.57*x)*cos((314)*t)
20 endfunction
21 x=5.66//value of x (in cm)

```

```

22 t=2//value of t (in s)
23 vn=f(t,x)//velocity of the particle
24
25 //for nodes.....sin(1.57 cm-1)*x = 0.....
    gives x=2*n
26 //since l=10 cm..nodes occur at 0 cm,2 cm,4 cm,6 cm
    ,8 cm,10 cm
27 //antinodes occur in between at 1 cm,3 cm,5 cm,7 cm
    ,9 cm
28 nloops=10*(1/2)
29
30 printf('the amplitude is %3.2f mm',103*A)
31 printf('\nthe wavelength is %3.2f cm',lambda)
32 printf('\nthe velocity is %3.2f cm/s',vn)// Textbook
    Correction : correct answer is 76.48 cm/s
33 printf('\nnodes occur at 0 cm,2 cm,4 cm,6 cm,8 cm,10
    cm')
34 printf('\nantinodes occur in between at 1 cm,3 cm,5
    cm,7 cm,9 cm')
35 printf('\nthe number of loops is %d',nloops)

```

Scilab code Exa 15.8 calculation of the length of the experimental wire to get the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.8
5 //calculation of the length of the experimental wire
    to get the resonance
6
7 //given data
8 nu1=256//frequency(in Hz) of the tuning fork 1
9 nu2=384//frequency(in Hz) of the tuning fork 2
10 l1=21//length(in cm) of the wire for tuning fork 1
11

```

```

12 //calculation
13 l2=(nu1/nu2)*l1//law of length
14
15 printf('the length of the experimental wire to get
    the resonance is %d cm',l2)

```

Scilab code Exa 15.8w calculation of the pressing in the guitar to produce required

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.8w
5 //calculation of the pressing in the guitar to
    produce required fundamental frequency
6
7 //given data
8 L1=90//length(in cm) of the guitar string
9 nu1=124//fundamental frequency(in Hz) for L1
10 nu2=186//required fundamental frequency(in Hz)
11
12 //calculation
13 //from equation of fundamental frequency....nu =
    (1/(2*L))*sqrt(F/mu)
14 L2=L1*(nu1/nu2)
15
16 printf('the pressing in the guitar to produce the
    fundamental frequency of 186 Hz is %d cm',L2)

```

Scilab code Exa 15.9w calculation of the position of bridges in sonometer wire

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 15.9w
5 //calculation of the position of bridges in
   sonometer wire
6
7 //given data
8 //nu1 : nu2 : nu3 = 1 : 2 : 3
9 L=1//length(in m) of the sonometer wire
10 m1=1//taking value from ratio
11 m2=2//taking value from ratio
12 m3=3//taking value from ratio
13
14 //calculation
15 //from formula of fundamental frequency.....nu =
   (1/(2*L))*sqrt(F/mu)
16 L1=L/((1/m1)+(1/m2)+(1/m3))//position of bridge 1
   from one end
17 L2=L1/2
18 L3=L1/3//position of bridge 2 from the other end
19
20 printf('the position of bridge 1 from one end is %3
   .2f m',L1)
21 printf('\nthe position of bridge 2 from the other
   end is %3.2f m',L3)

```

Scilab code Exa 15.10w calculation of the length of the wire

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 15.10w
5 //calculation of the length of the wire
6
7 //given data
8 mu=5*10^-3//mass density(in kg/m) of the wire
9 F=450//tension(in N) produced in the wire

```

```

10 nu1=420//frequency(in Hz) of nth harmonic
11 nu2=490//frequency(in Hz) of (n+1)th harmonic
12
13 //calculation
14 //from formula of fundamental frequency .....nu =
    (1/(2*L))*sqrt(F/mu) .....(1)
15 n=nu1/(nu2-nu1)//value of n
16 L=(n/(2*nu1))*sqrt(F/mu)//erom equation (1)
17
18 printf('the length of the wire is %3.1f m',L)

```

Chapter 16

Sound Waves

Scilab code Exa 16.1 calculation of the audibility of a wave

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.1
5 //calculation of the audibility of a wave
6
7 //given data
8 v=300//velocity(in m/s) of the wave
9 lambda=.60*10^-2//wavelength(in m) of the wave
10
11 //calculation
12 nu=v/lambda//frequency of the wave
13 if(nu<20)
14     printf('the wave is not audible')
15 elseif(nu>20000)
16     printf('the wave is not audible')
17 else
18     printf('the wave is audible')
19 end
```

Scilab code Exa 16.1w calculation of the depth of the sea and wavelength of the s

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.1w
5 //calculation of the depth of the sea and wavelength
   of the signal in the water
6
7 //given data
8 nu=50*103//frequency(in Hz) of the given signal
9 t=0.8//time(in s)requires for reflected wave to
   return
10 v=1500//speed(in m/s) of the sound in water
11
12 //calculation
13 d=v*t/2//depth of the sea
14 lambda=v/nu//wavelength in water
15
16 printf('the depth of the sea is %d m',d)
17 printf('\\nthe wavelength of the signal in the water
   is %3.1f cm',lambda*102)
```

Scilab code Exa 16.2 calculation of the amplitude of vibration of the particles of

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.2
5 //calculation of the amplitude of vibration of the
   particles of the medium
6
```

```

7 //given data
8 lambda=40*10^-2//wavelength(in m) of the wave
9 deltap=1*10^-3//difference between the minimum and
  the maximum pressure(in N/m^2)
10 B=1.4*10^5//Bulk modulus(in N/m^2)
11
12 //calculation
13 p0=deltap/2//pressure amplitude
14 s0=(p0*lambda)/(2*pi*B)//from equation of Bulk
  modulus
15
16 printf('the amplitude of vibration of the particles
  of the medium is %3.2e m',s0)

```

Scilab code Exa 16.2w calculation of the location of the plane

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.2w
5 //calculation of the location of the plane
6
7 //given data
8 v=510*10^3/(60*60)//speed(in m/s) of the plane
9 h=2000//height(in m) of the plane
10 vs=340//speed(in m.s) of the sound in air
11
12 //calculation
13 t=h/vs//time taken by the sound to reach the
  observer
14 d=v*t//location of the plane
15
16 printf('the plane will be %d m ahead of the observer
  on its line of motion',d)

```

Scilab code Exa 16.3 calculation of the intensity of the sound wave

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.3
5 //calculation of the intensity of the sound wave
6
7 //given data
8 p0=2*10^-2//pressure amplitude(in N/m^2)
9 p0dash=2.5*10^-2//new pressure amplitude(in N/m^2)
10 I=5.0*10^-7//intensity(in W/m^2) of the wave
11
12 //calculation
13 //intensity of the wave is proportional to square of
    the pressure amplitude
14 Idash=I*((p0dash/p0)^2)
15
16 printf('the intensity of the sound wave is %3.1e W/m
    ^2',Idash)
```

Scilab code Exa 16.3w calculation of the frequency wavelength speed maximum and m

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.3w
5 //calculation of the frequency ,wavelength ,speed ,
    maximum and minimum pressures of the sound wave
6
7 //given data
8 //equation of the sound wave is
```

```

9 //p = (0.01 N/m^2)*sin((1000 s^-1)*t - (3.0 m^-1)*x)
  ..... (1)
10 peq=1.0*10^5//equilibrium pressure(in N/m^2) of the
    air
11
12 //calculation
13 //comparing equation (1) with standard equation p =
    p0*sin(w*(t-(x/v)))...we get
14 w=1000//value of w(in s^-1)
15 nu=w/(2*%pi)//frequency
16 v=w/3//velocity
17 lambda=v/nu//wavelength
18 p0=0.01//pressure amplitude(in N/m^2)
19
20 printf('the frequency is %d Hz',nu)
21 printf('\nthe wavelength is %3.1f m',lambda)
22 printf('\nthe speed of the sound wave is %d m/s',v)
23 printf('\nthe maximum pressure amplitude is (%3.2e +
    %3.2f) N/m^2',peq,p0)
24 printf('\nthe minimum pressure amplitude is (%3.2e -
    %3.2f) N/m^2',peq,p0)

```

Scilab code Exa 16.4 calculation of the increase in the sound level in decibels

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.4
5 //calculation of the increase in the sound level in
    decibels
6
7 //given data
8 r=20//intensity is increase by r factor
9
10 //calculation

```

```

11 //using the equation.....beta = 10*log(I/I0)...we
    get
12 deltabeta=10*log10(r)//increase in sound level
13
14 printf('the increase in the sound level in decibels
    is %d dB',deltabeta)

```

Scilab code Exa 16.4w calculation of the minimum separation between the two points

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.4w
5 //calculation of the minimum separation between the
    two points for a given phase difference
6
7 //given data
8 nu=10*10^3//frequency(in Hz) of the sound wave
9 v=340//speed(in m/s) of the wave
10 delta=60//phase difference(in degree)
11
12 //calculation
13 lambda=v/nu//wavelength
14 k=2*pi/lambda//wave number
15 d=(delta*pi/180)/k
16
17 printf('the minimum separation between the two
    points for phase difference of 60 degree is %3.2f
    cm',d*10^2)

```

Scilab code Exa 16.5 calculation of the nature of interference

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.5
5 //calculation of the nature of interference
6
7 //given data
8 nu=1*10^3//frequency(in Hz) of the source
9 deltax=83*10^-2//difference in the length(in m) of
  paths
10 v=332//speed(in m/s) of the sound in air
11
12 //calculation
13 lambda=v/nu//wavelength
14 delta=(2*%pi/lambda)*deltax
15 n=delta/%pi//phase difference is 'n' multiple of pi
16 if(modulo(n,2)==0)
17     printf('the waves will interfere constructively.
  ')//for even values of 'n'
18 else
19     printf('the waves will interfere destructively.'
  )//for odd values of 'n'
20 end

```

Scilab code Exa 16.5w calculation of the atmospheric temperature

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.5w
5 //calculation of the atmospheric temperature
6
7 //given data
8 v1=336//speed(in m/s) travelled by the sound
9 v0=332//speed(in m/s) of the sound at 0
  degreecelsius

```

```

10 T0=0+273//temperature(in kelvin)
11
12 //calculation
13 T=((v1/v0)^2)*T0//temperature (in kelvin)
14 t=T-273//temperature(in degreecelsius)
15
16 printf('the atmospheric temperature is %d
    degreecelsius ',round(t))

```

Scilab code Exa 16.6 calculation of the distance of the piston from the open end f

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.6
5 //calculation of the distance of the piston from the
    open end,for tube to vibrate in its first
    overtone
6
7 //given data
8 nu=416//frequency(in Hz) of the tuning fork
9 v=333//speed(in m/s) of the sound in air
10
11 //calculation
12 lambda=v/nu//wavelength
13 L=3*lambda/4//length of the tube
14
15 printf('the distance of the piston from the open end
    ,for tube to vibrate in its first overtone is %3
    .1f cm',L*10^2)

```

Scilab code Exa 16.6w calculation of the speed of sound wave in hydrogen

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.6w
5 //calculation of the speed of sound wave in hydrogen
6
7 //given data
8 gama=1.4//value of constant gama for hydrogen
9 voxygen=470//speed(in m/s) of the sound wave in
   oxygen
10
11 //calculation
12 //speed of sound wave in a gas is ..... $v = \sqrt{\text{gama} \cdot P / \rho}$ 
13 //at STP ,density of oxygen is 16 times density of
   hydrogen
14 vhydrogen=voxygen*sqrt(16)//speed of sound in
   hydrogen
15
16 printf('the speed of sound wave in hydrogen is %d m/
   s ',vhydrogen)

```

Scilab code Exa 16.7 calculation of the tuning frequency of fork B

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.7
5 //calculation of the tuning frequency of fork B
6
7 //given data
8 nu1=384//tunning frequency(in Hz) of fork A
9 n=6//number of beats
10 t=2//time(in s) taken by the beats
11

```



```

12 //calculation
13 deltanu=n/t//frequency of beats
14 nu2=nu1+deltanu//frequency of fork B
15 nu2dash=nu1-deltanu//another frequency of fork B
16
17 printf('the tuning frequency of fork B is %d Hz or
        %d Hz',nu2dash,nu2)

```

Scilab code Exa 16.7w calculation of the energy delivered to the microphone

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.7w
5 //calculation of the energy delivered to the
  microphone
6
7 //given data
8 A=.80*10^-4//area(in m^2) of the cross section
9 U=3//power(in W) output of the speaker
10 d=2//distance(in m) between the microphone and the
  speaker
11 t=5//time(in s) taken
12
13 //calculation
14 U0=A*U/(4*pi*d^2)//energy falling on the microphone
  in 1 s
15 Udash=U0*t//energy falling on the microphone in t s
16
17 printf('the energy delivered to the microphone in t
        =5 s is %d microJ',round(Udash*10^6))

```

Scilab code Exa 16.8 calculation of the most dominant frequency

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.8
5 //calculation of the most dominant frequency
6
7 //given data
8 us=36*103/(60*60) //speed(in m/s) of the train
9 nudash=12*103//frequency(in Hz) detected by the
   detector
10 v=340 //velocity(in m/s) of the sound in air
11
12 //calculation
13 //frequency detected is ..... nudash = (v*nu0)/(v-us
   )
14 nu0=(1-(us/v))*nudash//required frequency
15
16 printf('the most dominant frequency is %3.1f kHz',
   nu0*10-3)

```

Scilab code Exa 16.8w calculation of the amplitude of vibration of the particles of the air

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.8w
5 //calculation of the amplitude of vibration of the
   particles of the air
6
7 //given data
8 I=2*10-6//intensity(in W/m2) of the sound wave
9 nu=1*103//frequency(in Hz) of the sound wave
10 rho0=1.2//density(in kg/m3) of the air
11 v=330 //speed(in m/s) of the sound in the air
12

```

```

13 //calculation
14 s0=sqrt(I/(2*%pi^2*nu^2*rho0*v))//equation of
    displacement amplitide
15
16 printf('the amplitude of vibration of the particles
    of the air is %3.1e m',s0)

```

Scilab code Exa 16.9w calculation of the factor by which the pressure amplitide :

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.9w
5 //calculation of the factor by which the pressure
    amplituide increases
6
7 //given data
8 n=30//increase(in dB) of the sound level
9
10 //calculation
11 //m = I2/I1 = intensity ratio
12 m=10^(n/10)
13 //since p2/p1 = sqrt(I2/I1)
14 f=sqrt(m)//require factor
15
16 printf('the factor by which the pressure amplituide
    increases is %d',round(f))

```

Scilab code Exa 16.10w calculation of the frequency at which the maxima of intens

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 16.10w
5 //calculation of the frequency at which the maxima
  of intensity are detected
6
7 //given data
8 r=20*10^-2//radius(in m) of the semicircular part
9 v=340//speed(in m/s) of the sound in air
10
11 //calculation
12 l1=2*r//straight distance
13 l2=%pi*r//curve distance
14 deltal=l2-l1
15 nu=v/deltal
16
17 printf('the frequency at which the maxima of
  intensity are detected are %d Hz and %d Hz',nu,2*
  nu)

```

Scilab code Exa 16.11w calculation of the minimum distance between the source and

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.11w
5 //calculation of the minimum distance between the
  source and the detector for maximum sound
  detection
6
7 //given data
8 nu=180//frequency(in Hz)
9 d=2//distance(in m)
10 v=360//speed(in m/s) of the sound wave in air
11
12 //calculation
13 //path difference ..... delta = (2*((2^2) + (x^2/4))

```

```

    ^ (1/2)) - (x)
14 lambda=v/nu//wavelength
15 delta=lambda
16 //solving the above equation ,we get
17 x=4-1
18
19 printf('the minimum distance between the source and
    the detector for maximum sound detection is %d m'
    ,x)

```

Scilab code Exa 16.12w calculation of the length of the shortest closed organ pipe

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.12w
5 //calculation of the length of the shortest closed
    organ pipe that will resonate with the tuning
    fork
6
7 //given data
8 nu=264//frequency(in Hz)of the tuning fork
9 v=350//speed(in m/s) of the sound in air
10
11 //calculation
12 //from the equation of the resonate frequency of the
    closed organ pipe....l = (n*v)/(4*nu)
13 n=1//for l to be minimum
14 lmin=(v)/(4*nu)//equation of the resonate frequency
    of the closed organ pipe
15
16 printf('the length of the shortest closed organ pipe
    that will resonate with the tuning fork is %d
    cm',lmin*10^2)

```

Scilab code Exa 16.13w calculation of the length of the closed pipe

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.13w
5 //calculation of the length of the closed pipe
6
7 //given data
8 l0=60*10^-2//length(in m) of the open pipe
9
10 //calculation
11 //from the equation of the resonate frequency of the
    closed organ pipe.... $l=(n*v)/(4*nu)$ 
12 l1=l0/4
13
14 printf('the length of the closed pipe is %d cm',l1
    *10^2)
```

Scilab code Exa 16.14w calculation of the speed of the sound in air

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.14w
5 //calculation of the speed of the sound in air
6
7 //given data
8 nu=800//frequency(in Hz) of the tuning fork
9 l1=9.75*10^-2//distance(in m) where resonance is
    observed
```

```

10 l2=31.25*10^-2//distance(in m) where resonance is
    observed
11 l3=52.75*10^-2//distance(in m) where resonance is
    observed
12
13 //calculation
14 //from the equation of the resonate frequency ....l
    = (n*v)/(4*nu)
15 //(n*v)/(4*l1) = nu.....(1)
16 //((n+2)*v)/(4*l2) = nu.....(2)
17 //((n+4)*v)/(4*l3) = nu.....(3)
18 //form above equations ,we get
19 v=2*nu*(l2-l1)
20
21 printf('the speed of the sound in air is %d m/s',v)

```

Scilab code Exa 16.15w calculation of the fundamental frequency if the air is rep

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.15w
5 //calculation of the fundamental frequency if the
    air is replaced by hydrogen
6
7 //given data
8 nu0=500//fundamental frequency(in Hz)
9 rhoa=1.20//density(in kg/m^3) of air
10 rhoh=0.089//density(in kg/m^3) of hydrogen
11
12 //calculation
13 //fundamental frequency of an organ pipe is
    proportional to the speed of the sound
14 nu=nu0*sqrt(rhoa/rhoh)
15

```

```
16 printf('the fundamental frequency if the air is
    replaced by hydrogen is %d Hz',nu)
```

Scilab code Exa 16.16w calculation of the speed wavelength in the rod frequency w

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.16w
5 //calculation of the speed ,wavelength in the rod ,
    frequency ,wavelength in the air
6
7 //given data
8 l=90*10^-2//length(in m) of the rod
9 rho=2600//density(in kg/m^3) of the aluminium
10 Y=7.80*10^10//Young modulus(in N/m^2)
11 vai=340//speed(in m/s) of the sound in the air
12
13 //calculation
14 v=sqrt(Y/rho)//speed of the sound in aluminium
15 lambda=2*l//wavelength....since rod vibrates with
    fundamental frequency
16 nu=v/lambda//frequency
17 lambdaai=vai/nu//wavelength in the air
18
19 printf('the speed of the sound in aluminium is %d m/
    s',v)//Textbook Correction : correct answer is
    5477 m/s
20 printf('\nthe wavelength of the sound in aluminium
    rod is %d cm',lambda*10^2)
21 printf('\nthe frequency of the sound produced is %d
    Hz',nu)//Textbook Correction : correct answer is
    3042 Hz
22 printf('\nthe wavelength of the sound in air is %3.1
    f cm',lambdaai*10^2)
```

Scilab code Exa 16.17w calculation of the frequency of the note emitted by the ta

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.17w
5 //calculation of the frequency of the note emitted
   by the taut string
6
7 //given data
8 nu1=440//frequency(in Hz) of the string
9 n=4//number of beats per second
10 nuf=440//tunning frequency(in Hz) of the fork
11
12 //calculation
13 fre=nuf+n//required frequency
14
15 printf('the frequency of the note emitted by the
   taut string is %d Hz',fre)
```

Scilab code Exa 16.18w calculation of the apparent frequency

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.18w
5 //calculation of the apparent frequency
6
7 //given data
8 us=36*10^3/(60*60)//speed(in m/s)of the car
9 v=340//speed(in m/s) of the sound in the air
```

```

10 nu=500//frequency(in Hz)
11
12 //calculation
13 nudash=(v/(v+us))*nu//apparent frequency heard by
    the observer
14 nudashdash=(v/(v-us))*nu//frequency received by the
    wall
15
16 printf('the apparent frequency heard by the ground
    observer is %d Hz',round(nudash))
17 printf('\nthe frequency of the reflected wave as
    heard by the ground observer is %d Hz',nudashdash
    )

```

Scilab code Exa 16.19w calculation of the frequency of the whistle of the train

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.19w
5 //calculation of the frequency of the whistle of the
    train
6
7 //given data
8 us=72*10^3/(60*60)//speed(in m/s) of the train 1
9 u0=54*10^3/(60*60)//speed(in m/s) of the train 2
10 nu=600//frequency(in Hz) of the whistle
11 v=340//speed(in m/s)of sound in the air
12
13 //calculation
14 nudash=((v+u0)/(v-us))*nu//frequency heard by the
    observer before the meeting of the trains
15 nudashdash=((v-u0)/(v+us))*nu//frequency heard by
    the observer after the crossing of the trains
16

```

```

17 printf('the frequency heard by the observer before
    the meeting of the trains is %d Hz',round(nudash)
    )
18 printf('\nthe frequency heard by the observer after
    the crossing of the trains is %d Hz',round(
    nudashdash))

```

Scilab code Exa 16.20w calculation of the main frequency heard by the person

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.20w
5 //calculation of the main frequency heard by the
    person
6
7 //given data
8 us=36*10^3/(60*60)//speed(in m/s) of the person on
    the scooter
9 v=340//speed(in m/s) of sound in the air
10 nu=600//frequency(in Hz) of the siren
11
12 //calculation
13 nudash=(v/(v+us))*nu//main frequency
14
15 printf('the main frequency heard by the person is %d
    Hz',round(nudash))

```

Scilab code Exa 16.21w calculation of the original frequency of the source

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 16.21w
5 //calculation of the original frequency of the
   source
6
7 //given data
8 u0=10//speed(in m/s) of the observer going away from
   the source
9 us=10//speed(in m/s) of the source going away from
   observer
10 nudash=1950//frequency(in Hz) of the sound detected
   by the detector
11 v=340//speed(in m/s) of the sound in the air
12
13 //calculation
14 nu=((v+us)/(v-u0))*nudash//original frequency
15
16 printf('the original frequency of the source is %d
   Hz',round(nu))

```

Scilab code Exa 16.22w calculation of the speed of the car

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.22w
5 //calculation of the speed of the car
6
7 //given data
8 nudash=440//frequency(in Hz) emitted by the wall
9 nudashdash=480//frequency(in Hz) heard by the car
   driver
10 v=330//speed(in m/s) of the sound in the air
11
12 //calculation
13 //frequency received by the wall..... nudash

```

```

    = (v/(v-u))*nu.....(1)
14 //frequency(in Hz) heard by the car driver....
    nudashdash = ((v+u)/v)*nudash....(2)
15 //from above two equations,we get
16 u=((nudashdash-nudash)/(nudashdash+nudash))*v//speed
    of the car
17
18 printf('the speed of the car is %3.1f m/s or %d km/h
    ',u,round(u*10^-3*60*60))

```

Scilab code Exa 16.23w calculation of the frequency of train whistle heard by the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 16.23w
5 //calculation of the frequency of train whistle
    heard by the person standing on the road
    perpendicular to the track
6
7 //given data
8 v=340//speed(in m/s) of the sound in the air
9 d1=300//distance(in m) of the train from the
    crossing
10 u=120*10^3/(60*60)//speed(in m/s) of the train
11 nu=640//frequency(in Hz) of the whistle
12 d2=400//distance(in m) of the person from the
    crossing ,perpendicular to the track
13
14 //calculation
15 theta=acosd(d1/sqrt(d1^2+d2^2))//pythagoras theorem
16 nudash=(v/(v-(u*cosd(theta))))*nu//frequency of the
    whistle heard
17
18 printf('the frequency of train whistle heard by the

```

person standing on the road perpendicular to the
track is %d Hz',nudash)

Chapter 17

Light Waves

Scilab code Exa 17.1 calculation of the speed of light in glass

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.1
5 //calculation of the speed of light in glass
6
7 //given data
8 mu=1.5//refractive index of glass
9 v0=3*10^8//speed(in m/s) of light in vacuum
10
11 //calculation
12 v=v0/mu//definition of the refractive index
13
14 printf('the speed of light in glass is %3.1e m/s',v)
```

Scilab code Exa 17.1w calculation of the limits of wavelengths in the water

```
1 //developed in windows XP operating system 32bit
```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.1w
5 //calculation of the limits of wavelengths in the
   water
6
7 //given data
8 lambda01=400//mimimum wavelength(in nm) of the light
   used
9 lambda02=700//maximum wavelength(in nm) of the light
   used
10 mu=1.33//refractive index of water
11
12 //calculation
13 lambda1=lambda01/mu//definition of the refractive
   index
14 lambda2=lambda02/mu//definition of the refractive
   index
15
16 printf('the limits of wavelengths in the water are
   %d nm and %d nm',lambda1,lambda2)

```

Scilab code Exa 17.2 calculation of the separation between successive bright fringes

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.2
5 //calculation of the separation between successive
   bright fringes
6
7 //given data
8 d=0.10*10-3//separation(in m) between the slits
9 lambda=600*10-9//wavelength(in m) of the light used
10 D=1//separation(in m) between the slits and the

```



```

        screen
11
12 //calculation
13 w=D*lambda/d//separation between successive bright
    fringes
14
15 printf('the separation between successive bright
    fringes is %3.1e m or %3.1f mm',w,w*10^3)

```

Scilab code Exa 17.2w calculation of the refractive index of the glass

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.2w
5 //calculation of the refractive index of the glass
6
7 //given data
8 x1=2//distance(in cm)travelled through the glass
9 x2=2.25//distance(in cm)travelled through the water
10 muw=1.33//refractive index of water
11
12 //calculation
13 //for 'x' distance travelled through a medium of
    refractive index 'mu',the optical path is 'mu*x'
14 mug=muw*x2/x1//refractive index of glass
15
16 printf('the refractive index of the glass is %3.2f',
    mug)

```

Scilab code Exa 17.3 calculation of the wavelength of light in the water

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.3
5 //calculation of the wavelength of light in the
   water
6
7 //given data
8 lambdan=589//wavelength(in nm) of light in vacuum
9 mu=1.33//refractive index of water
10
11 //calculation
12 lambda=lambdan/mu//definition of the refractive
   index
13
14 printf('the wavelength of light in the water is %d
   nm',round(lambda))

```

Scilab code Exa 17.3w calculation of the wavelengths of the violet and the red light

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.3w
5 //calculation of the wavelengths of the violet and
   the red light
6
7 //given data
8 D=2.5//separation(in m) between the slit and the
   screen
9 d=0.5*10^-3//separation(in m) between the slits
10 yv=2*10^-3//distance(in m) between the central white
   fringe and the first violet fringe
11 yr=3.5*10^-3//distance(in m) between the central
   white fringe and the first red fringe
12

```

```

13 //calculation
14 lambdav=yv*d/D//wavelength of the violet light
15 lambdar=yr*d/D//wavelength of the red light
16
17 printf('the wavelength of the violet light is %d nm',
        ,lambdav*10^9)
18 printf('\nthe wavelength of the red light is %d nm',
        lambdar*10^9)

```

Scilab code Exa 17.4 calculation of the minimum thickness of the film

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.4
5 //calculation of the minimum thickness of the film
6
7 //given data
8 lambda=589//wavelength(in nm) of the light used
9 mu=1.25//refractive index of the material
10
11 //calculation
12 //for strong reflection .....2*mu*d = lambda/2
13 d=lambda/(4*mu)//minimum thickness
14
15 printf('the minimum thickness of the film is %d nm',
        round(d))

```

Scilab code Exa 17.4w calculation of the separation between the slits

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 17.4w
5 //calculation of the separation between the slits
6
7 //given data
8 lambda=589.3*10^-9//wavelength(in m) of the sodium
  light
9 D=100*10^-2//separation(in m) between the slit and
  the screen
10 n=10//number of the bright fringe
11 x=12*10^-3//distance(in m) between the central
  maximum and the tenth bright fringe
12
13 //calculation
14 d=n*lambda*D/x//separation between the slits
15
16 printf('the separation between the slits is %3.1e m
  or %3.2 f mm',d,d*10^3)

```

Scilab code Exa 17.5 calculation of the angular divergence for most of the light g

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.5
5 //calculation of the angular divergence for most of
  the light getting diffracted
6
7 //given data
8 lambda=450*10^-9//wavelength(in m) of the light used
9 b=0.2*10^-3//width(in m) of the slit
10
11 //calculation
12 //for theta tends to zero.....sin(theta) = theta
13 theta1=lambda/b//angle of minima
14 theta2=-lambda/b//angle of minima

```

```

15 theta=theta1-theta2//angular divergence
16
17 printf('the angular divergence for most of the light
    getting diffracted is %3.1e radian',theta)

```

Scilab code Exa 17.5w calculation of the ratio of maximum intensity to the minimum

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.5w
5 //calculation of the ratio of maximum intensity to
    the minimum intensity in the interference fringe
    pattern
6
7 //given data
8 //intensity of the light coming from one slit in
    Young's double slit experiment is double the
    intensity of the light coming from the other slit
9 n=2
10
11 //calculation
12 r=((sqrt(n)+1)^2)/((sqrt(n)-1)^2)//required ratio
13
14 printf('the ratio of maximum intensity to the
    minimum intensity in the interference fringe
    pattern is %d',round(r))

```

Scilab code Exa 17.6 calculation of the diameter of the disc image

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 17.6
5 //calculation of the diameter of the disc image
6
7 //given data
8 lambda=590*10^-9//wavelength(in m) of the light used
9 b=10*10^-2//diameter(in m) of the converging lens
   used
10 d=20//distance(in m) between the lens and the point
   of focus
11
12 //calculation
13 sintheta=1.22*lambda/b//angular radius
14 r=d*sintheta//radius of the disc image
15 d=2*r//diameter of the disc image
16
17 printf('the diameter of the disc image is %3.1e cm',
   d)

```

Scilab code Exa 17.6w calculation of the ratio of maximum intensity to the minimum

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.6w
5 //calculation of the ratio of maximum intensity to
   the minimum intensity in the interference pattern
6
7 //given data
8 //width of one slit in Young's double slit
   experiment is double that of the other
9 n=2
10
11 //calculation
12 r=((n+1)^2)/((n-1)^2)//required ratio
13

```

```
14 printf('the ratio of maximum intensity to the
    minimum intensity in the interference pattern is
    %d',r)
```

Scilab code Exa 17.7w calculation of the maximum and the minimum path difference

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.7w
5 //calculation of the maximum and the minimum path
  difference at the detector
6
7 //given data
8 lambda=600*10^-9//wavelength(in m) of the light
9 d=1*10^-2*10^-2//distance(in m) between the sources
10
11 //calculation
12 pdmax=d//path difference maximum
13 pdmin=0//path difference minimum
14 //farthest minima occurs for path difference lambda
  /2
15 //sqrt(D^2 + d^2) - D = lambda/2
16 D=(d^2/lambda)-(lambda/4)//distance of the farthest
  minima
17
18 printf('the maximum path difference on moving the
  detector along S1P line is %3.1e cm',pdmax*10^2)
19 printf('\nthe minimum path difference on moving the
  detector along S1P line is %3.1f cm',pdmin*10^2)
20 printf('\nthe farthest minimum is located at a
  distance of %3.1f cm from the point S1',D*10^2)
```

Scilab code Exa 17.8w calculation of the distance of bright fringe from the centre

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 17.8w
5 //calculation of the distance of bright fringe from
   the central maximum
6
7 //given data
8 lambda1=6500*10^-10//wavelength(in m) of the light
   beam1
9 lambda2=5200*10^-10//wavelength(in m) of the light
   beam2
10 d=2.0*10^-3//separation(in m) between the slits
11 D=120*10^-2//separation(in m) between the slits and
   the screen
12 n=3//number of the bright fringe
13
14 //calculation
15 y=n*lambda1*D/d//the distance of bright fringe from
   the central maximum
16 //from the equation of the distance of bright fringe
   from the central maximum.....y=n*lambda*D/d
17 //let m th bright fringe of beam 1 coincides with n
   th bright fringe of beam 2
18 //ym = yn
19 //m : n = 4 : 5.....is their minimum integral ratio
20 m=4
21 ym=m*lambda1*D/d//least distance from the central
   maximum where both wavelengths coincides
22 printf('the distance of the third bright fringe from
   the central maximum is %3.2f cm',y*10^2)
23 printf('\\nthe least distance from the central
   maximum where both the wavelengths coincides is
   %3.2f cm',ym*10^2)
```

Scilab code Exa 17.9w calculation of the number of fringes that will shift due to

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.9w
5 //calculation of the number of fringes that will
   shift due to introduction of the sheet
6
7 //given data
8 lambda=600*10-9//wavelength(in m) of the light used
9 t=1.8*10-5//thickness(in m) of the transparent
   sheet
10 mu=1.6//refractive index of the material
11
12 //calculation
13 n=((mu-1)*t)/lambda//number of fringes shifted
14
15 printf('the number of fringes that will shift due to
   introduction of the sheet is %d',n)
```

Scilab code Exa 17.10w calculation of the wavelengths in the visible region that

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.10w
5 //calculation of the wavelengths in the visible
   region that are strongly reflected
6
7 //given data
8 d=.5*10-6//thickness(in m) of the glass plate
```

```

9 mu=1.5//refractive index of the medium
10 lambda1=400*10^-9//minimum wavelength(in m) of the
    visible region
11 lambda2=700*10^-9//maximum wavelength(in m) of the
    visible region
12
13 //calculation
14 //condition for strong reflection of light of
    wavelength lambda is
15 //2*mu*d = (n + (1/2))*lambda.....(1)
16 n1=round((2*mu*d/lambda1)-(1/2))//integral value of
    n for lambda1
17 n2=round((2*mu*d/lambda2)-(1/2))//integral value of
    n for lambda2
18 lambda1n=(2*mu*d)/(n1+(1/2))//from equation (1)
19 lambda2n=(2*mu*d)/(n2+(1/2))//from equation (1)
20
21 printf('the wavelengths in the visible region that
    are strongly reflected are %d nm and %d nm',round
    (lambda1n*10^9),round(lambda2n*10^9))

```

Scilab code Exa 17.11w calculation of the distance between the two first order mi

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 17.11w
5 //calculation of the distance between the two first
    order minima
6
7 //given data
8 b=.40*10^-3//width(in m) of the slit
9 D=40*10^-2//separation(in m) between the slit and
    the screen
10 lambda=546*10^-9//wavelength(in m) of the light used

```

```
11
12 //calculation
13 //linear distances from the central maxima are given
    by ..  $x = D \tan(\theta) = D \sin(\theta) = \pm \lambda D$ 
    /b
14 sep=2*lambda*D/b//separation between the minima
15
16 printf('the distance between the two first order
    minima is %3.1f mm',sep*10^3)
```

Chapter 18

Geometrical Optics

Scilab code Exa 18.1 calculation of position of the image of an object placed at a

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.1
5 //calculation of position of the image of an object
   placed at a distance from the mirror.
6
7 //given data
8 u=-12; //object distance(in cm)
9 R=20; //radius of curvature of the mirror (in cm)
10
11 //calculation
12 v=1/((2/R)-(1/u)); //mirror formula
13
14 if(v>0)
15     disp(v, 'virtual image is formed on right side of
       mirror at a distance(in cm)');
16 else
17     disp(v, 'real image is formed on left side of
       mirror at a distance(in cm)');
18 end
```

Scilab code Exa 18.1w calculation of position and nature of the image of an object

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.1w
5 //calculation of position and nature of the image of
   an object placed at a distance from a concave
   mirror
6
7 //given data
8 u=-8; //object distance(in cm)
9 f=-10; //focal length of the concave mirror(in cm)
10
11 //calculation
12 v=1/((1/f)-(1/u)); //mirror formula
13
14 if(v>0)
15     disp(v, 'virtual image is formed on right side of
   mirror at a distance(in cm)');
16 else
17     disp(v, 'real image is formed on left side of
   mirror at a distance(in cm)');
18 end
```

Scilab code Exa 18.2 calculation of length of the image of an object placed at a d

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.2
```

```

5 //calculation of length of the image of an object
   placed at a distance from a concave mirror.
6
7 //given data
8 //F=-f    focal length(in cm)
9 //u=-1.5f  object distance(in cm)
10 h1=2.5; //object height(in cm)
11
12 //calculation
13 //v=1/((1/F)-(1/u))    mirror formula
14 //v=-3f
15 //also m=-v/u    lateral magnification formula for
   mirror
16 //m=-2    lateral magnification ratio
17
18 m=-2; //lateral magnification ratio
19 h2=m*h1; //lateral magnification formula
20
21 if(h2>0)
22     disp(h2,'image is erect and is of length(in cm)')
   );
23 else
24     disp(h2,'image is inverted and is of length(in
   cm)');
25 end

```

Scilab code Exa 18.2w calculation of length of the image of an object placed hori

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.2w
5 //calculation of length of the image of an object
   placed horizontal at a distance from the mirror.
6

```

```

7 //given data
8 u=-30; //object distance of point A(in cm)
9 f=-10; //focal length of the mirror(in cm)
10 //r=2f=20 cm
11 //image of B is formed at centre of curvature since
    it is located at the centre of curvature.
12
13 //calculation
14 v=1/((1/f)-(1/u)); //mirror formula
15
16 disp(v+(2*-f), 'length(in cm) of the image is');

```

Scilab code Exa 18.3 calculation of shift in the position of printed letters by a

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.3
5 //calculation of shift in the position of printed
    letters by a glass cube
6
7 //given data
8 t=6; //thickness of the cube(in cm)
9 mu=1.5; //refractive index of glass cube
10
11 //calculation
12 deltat=(1-1/mu)*t; //vertical shift formula derived
    from snell's law
13
14 disp(deltat, 'shift(in cm) in the position of printed
    letters is');

```

Scilab code Exa 18.3w calculation of object distance for half image height as comp

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.3w
5 //calculation of object distance for half image
   height as compared to original height in case of
   reflection by convex mirror
6
7 //given data
8 m=.5; //magnification ratio
9 f=2.5; //focal length of the convex mirror(in m)
10
11 //calculation
12 //(1/u)+(1/v)=(1/f); //mirror formula
13 //now m=-v/u=0.5
14 u=-f; //from formula taking  $v=-u/2$  mirror formula
   gives this relation
15
16 disp(abs(u), 'the boy should stand at a distance(in m
   ) from the convex mirror');
```

Scilab code Exa 18.4 calculation of refractive index of material from known critical

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.4
5 //calculation of refractive index of material from
   known critical angle
6
7 //given data
8 thetac=48.2; //critical angle for water(in degree)
9
10 //calculation
11 //snell's law with respect to total internal
```



```

    reflection
12 mu=1/sind(thetac); //sind represents that the
    argument is in degree
13
14 disp(mu,'refractive index of material is ');

```

Scilab code Exa 18.4w calculation of image distance and focal length of concave m

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.4w
5 //calculation of image distance and focal length of
    concave mirror
6
7 //given data
8 h1=2; //height of object(in cm)
9 h2=-5; //height of image(in cm)
10 u=-12; //object distance in cm
11
12 //calculation
13 v=-(h2/h1)*u //image distance(in cm) using formula
    of lateral magnification
14
15 if(v<0)
16     disp(-v,'image is formed on same side of object
        at a distance(in cm)');
17 else
18     disp(v,'image is formed on opposite side of
        mirror at a distance(in cm)');
19 end
20
21 f=(u*v)/(u+v); //mirror formula
22
23 disp(abs(f),'focal length(in cm) of the given

```

```
concave mirror is ');
```

Scilab code Exa 18.5 calculation of refractive index of material from known value

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 18.5
5 //calculation of refractive index of material from
   known value of angle of minimum deviation by
   prism
6
7 //given data
8 deltam=37; // angle of minimum deviation by prism of
   the material(in degree)
9 A=53; //angle of prism(in degree)
10
11 //calculation
12 mu=sind((A+deltam)/2)/sind(A/2); //relation between
   refractive index and angle of minimum deviation
   by prism
13
14 disp(mu, 'refractive index of material of the prism
   is ');
```

Scilab code Exa 18.5w calculation of maximum angle of reflection for a surface

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 18.5w
5 //calculation of maximum angle of reflection for a
   surface
```

```

6
7 //given data
8 mu=1.25; //refractive index of medium
9
10 //calculation
11 thetadashdash=asind(1/mu); //critical angle for
    total internal reflection(in degree)
12 thetadash=90-thetadashdash;
13 theta=asind(mu*sind(thetadash)); //snell's law
    sin(theta)/sin(thetadash)=mu
14
15 disp(theta,'maximum value of theta(in degree) for
    total internal reflection at vertical surface');

```

Scilab code Exa 18.6 calculation of position of the image of an object placed at a

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.6
5 //calculation of position of the image of an object
    placed at a distance from spherical convex
    surface
6
7 //given data
8 u=-15; //object distance(in cm)
9 R=30; //radius of curvature of the spherical convex
    surface(in cm)
10 mu1=1; //refractive index of the medium in which
    object is kept
11 mu2=1.5; //refractive index of the medium of
    spherical convex surface
12
13 //calculation
14 v=mu2/((mu2-mu1)/R+(mu1/u)); //formula for

```

```

    refraction at spherical surface
15
16 if(v>0)
17     disp(v,'real image is formed on right side of
        spherical surface at a distance(in cm)');
18 else
19     disp(v,'virtual image is formed on left side of
        spherical surface at a distance(in cm)');
20 end

```

Scilab code Exa 18.6aw calculation of minimum refractive index for parallel emergence

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.6aw
5 //calculation of minimum refractive index for
    parallel emergence for given condition in right
    prism
6
7 //given data
8 thetac=45; //critical angle(in degree) for given
    conditions
9
10 //calculation
11 mu=1/(sind(thetac)); //anell's law
12
13 disp(mu, 'for total internal reflection refractive
    index of material of given right prism should be
    greater than or equal to');

```

Scilab code Exa 18.6bw verification of total internal reflection for given condit

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.6bw
5 //verification of total internal reflection for
   given conditions of right prism
6
7 //given data
8 mu=5/3; //refractive index of the material of the
   right prism
9
10 //calculation
11 thetac=asind(1/mu) //snell 's law
12
13 if(thetac<60)
14     disp('total internal reflection does not take
           place for given conditions of right prism');
15 else
16     disp('total internal reflection do take place
           for given conditions of right prism');
17 end

```

Scilab code Exa 18.7 calculation of the size of the image of an object placed at a

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.7
5 //calculation of the size of the image of an object
   placed at a distance from the spherical concave
   surface
6
7 //given data
8 u=-40; //object distance(in cm)
9 R=-20; //radius of curvature of the spherical

```

```

    concave surface(in cm)
10 mu1=1; //refractive index of the medium in which
    object is kept
11 mu2=1.33; //refractive index of the medium of
    spherical concave surface
12 h1=1; //size of the object(in cm)
13
14 //calculation
15 v=mu2/((mu2-mu1)/R+(mu1/u)); //formula for
    refraction at spherical surface
16 h2=(mu1*v*h1)/(mu2*u); //formula for lateral
    magnification
17
18 if(h2>0)
19     disp(h2,'image is erect and is of size(in cm)');
20 else
21     disp(h2,'image is inverted and is of size(in cm)
        ');
22 end

```

Scilab code Exa 18.8 calculation of focal length of a biconvex lens from known val

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.8
5 //calculation of focal length of a biconvex lens
    from known value of radii of curvature of
    refracting surfaces
6
7 //given data
8 R1=20; //radius of curvature(in cm) of first surface
    of biconvex lens
9 R2=-20; //radius of curvature(in cm) of second
    surface of biconvex lens

```

```

10 mu=1.5; //refractive index of the material of lens
11
12 //calculation
13 f=1/((mu-1)*(1/R1-1/R2)); //lens maker's formula
14
15 disp(f,'focal length(in cm) of the given biconvex
    lens is ');

```

Scilab code Exa 18.9 calculation of size of the image of an object placed at a distance

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.9
5 //calculation of size of the image of an object
    placed at a distance from a convex lens
6
7 //given data
8 f=12 //focal length(in cm)
9 u=-8 //object distance(in cm)
10 h1=2; //object height(in cm)
11
12 //calculation
13 v=1/((1/f)+(1/u)); //lens formula
14 m=v/u; //lateral magnification formula for lens
15 h2=m*h1; //lateral magnification formula for lens
16
17 if(h2>0)
18     disp(h2,'image is erect and is of length(in cm)')
19     );
19 else
20     disp(h2,'image is inverted and is of length(in
        cm) ');
21 end

```

Scilab code Exa 18.11w locating image of a dust particle on the surface of water :

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc; clear;
4 //example 18.11w
5 //locating image of a dust particle on the surface
   of water filled in a concave mirror as observed
   from top
6
7 //given data
8 R=-40; //radius of curvature(in cm) of the concave
   mirror
9 u=-5; //object distance(in cm) from the concave
   mirror
10 mu=1.33; //refractive index of water
11
12 //calculation
13 v=1/((2/R)-(1/u))//mirror formula
14
15 if(v>0)
16     disp(v,'virtual image is formed due to
   reflection through concave mirror below
   surface of mirror at a depth(in cm) of ');
17 else
18     disp(v,'real image is formed due to reflection
   through concave mirror above surface of
   mirror at a height(in cm) of ');
19 end
20
21 total_distance=v+(-u); //water is filled upto
   height equal to object distance of dust particle
   P
22 vfinal=total_distance*(1-1/mu); //snell's law
```


23

```
24 disp(total_distance-vfinal,'final image is formed  
    below water surface at a distance(in cm)');
```

Scilab code Exa 18.12w calculation of position of final image formed due to a sys

```
1 //developed in windows XP operating system 32bit  
2 //platform Scilab 5.4.1  
3 clc;clear;  
4 //example 18.12w  
5 //calculation of position of final image formed due  
    to a system of glass slab and a concave mirror  
6  
7 //given data  
8 u=-21; //object distance(in cm) from concave mirror  
9 R=20; //radius of curvature(in cm) of the concave  
    mirror  
10 mu=1.5; //refractive index of the glass '  
11 t=3; //thickness of glass slab(in cm)  
12  
13 //calculation  
14 tshift=t*(1-1/mu); //snell's law  
15 img_pos=-u-tshift; //image position with respect to  
    glass slab,i.e object distance(in cm) of concave  
    mirror  
16  
17 if(img_pos==R)  
18     disp('here img_pos is same as radius of  
        curvature of concave mirror and thus final  
        image is formed at P itself');
```

Scilab code Exa 18.13w calculation angle of minimum deviation for equilateral pri

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.13w
5 //calculation angle of minimum deviation for
   equilateral prism of silicate flint glass from
   known vlue of wavelength
6
7 //given data
8 A=60; //angle of prism(in degree)
9 mu1=1.66; //refractive index of silicate flint glass
   for 400nm wavelength
10 mu2=1.61; //refractive index of silicate flint glass
   for 700nm wavelength
11
12 //calculation
13 //mu=sind((A+deltam)/2)/sind(A/2) relation
   between refractive index and angle of minimum
   deviation by prism
14 deltam1=2*((asind(mu1*sind(A/2)))-30);
15 deltam2=2*((asind(mu2*sind(A/2)))-30);
16
17 disp(deltam1,'minimum angle of deviation(in degree)
   for 400nm wavelength in equilateral prism of
   silicate flint glass is ');
18 disp(deltam2,'minimum angle of deviation(in degree)
   for 700nm wavelength in equilateral prism of
   silicate flint glass is ');

```

Scilab code Exa 18.14w calculation of angle of rotation of the mirror in given se

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.14w

```

```

5 //calculation of angle of rotation of the mirror in
   given setup
6
7 //given data
8 mu=1.5; //refractive index of convex lens
9 A=4; //angle of prism (in degree)
10
11 //calculation
12 delta=(mu-1)*A
13
14 disp(delta,'the mirror should be rotated by angle(in
   degree) of');

```

Scilab code Exa 18.15w calculation of location of the image of an object placed a

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.15w
5 //calculation of location of the image of an object
   placed at a distance from the spherical convex
   surface
6
7 //given data
8 u=-25; //object distance(in cm)
9 R=20; //radius of curvature of the spherical convex
   surface(in cm)
10 mu1=1; //refractive index of the medium in which
   object is kept
11 mu2=1.5; //refractive index of the medium of
   spherical convex surface
12
13 //calculation
14 v=mu2/((mu2-mu1)/R+(mu1/u)) //formula for refraction
   at spherical surface

```

```

15
16 if(v>0)
17     disp(v,'image is formed on the right of the
           separating surface at a distance(in cm) of');
18 else
19     disp(-v,'image is formed on the left of the
           separating surface at a distance(in cm) of');
20 end

```

Scilab code Exa 18.16w calculation of height of the image of an object placed along

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.16w
5 //calculation of height of the image of an object
   placed along axis ,at a distance from a
   horizontal cylindrical glass rod
6
7 //given data
8 u=-20; //object distance (in cm)
9 R=5; //radius of curvature of the spherical convex
   surface (in cm)
10 mu1=1; //refractive index of the medium in which
   object is kept
11 mu2=1.5; //refractive index of the medium of
   spherical concave surface
12 h1=.5; //height of the object in mm
13
14 //calculation
15 v=mu2/((mu2-mu1)/R+(mu1/u)) //formula for refraction
   at spherical surface
16 m=(mu1*v)/(mu2*u); //lateral magnification ratio
17
18 if(v>0)

```

```

19     disp(v, 'image is formed inside the rod at a
        distance(in cm) of');
20 if(m==-1)
21     disp('the image will be of same height as the
        object and is inverted');
22 if(m==1)
23     disp('the image will be of same height as the
        object and is erect');
24 end

```

Scilab code Exa 18.17w calculation of apparent depth of a air bubble inside a gla

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.17w
5 //calculation of apparent depth of a air bubble
   inside a glass sphere
6
7 //given data
8 u=-4; //object distance (in cm)
9 R=-10; //radius of curvature of the spherical glass
   sphere(in cm)
10 mu1=1.5; //refractive index of the glass sphere
11 mu2=1; //refractive index of air bubble
12
13 //calculation
14 v=mu2/((mu2-mu1)/R+(mu1/u)) //formula for refraction
   at spherical surface
15
16 if(v<0)
17     disp(-v, 'below the surface the bubble will
        appear at a distance(in cm) of');
18 else
19     disp(v, 'above the surface the bubble will

```

```
        appear at a distance(in cm) of');
20 end
```

Scilab code Exa 18.18w calculation of position of image due to refraction at the

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.18w
5 //calculation of position of image due to refraction
   at the first surface and position of final image
6
7 //given data1
8 //u=infinite    object distance (in cm)
9 R=2; //radius of curvature of the spherical convex
   surface (in mm)
10 mu1=1.33; //refractive index of the medium from
   which light beam is entering
11 mu2=1; //refractive index of the medium of spherical
   air bubble
12 nR=-2; //radius of curvature of the spherical convex
   surface (in mm)
13 nmu1=1; //refractive index of the medium in which
   previous image is formed
14 nmu2=1.33; //refractive index of the medium from
   which light beam is entering
15
16 //calculation
17 v=R/(mu2-mu1) //formula for refraction at spherical
   surface for object at infinite distance
18 nu=-(-v+- (2*nR))
19
20 if(v<0)
21     disp(-v,'virtual image is formed on the same
   side of water at a distance(in mm) of');
```

```

22  else
23      disp(v,'real image is formed on the other side
           of water at a distance(in mm) of');
24  end
25
26  nv=nmu2/((nmu2-nmu1)/nR+(nmu1/nu)) //formula for
           refraction at spherical surface
27
28  if(nv<0)
29      disp(-nv,'final image is formed on the same side
           of air at a distance(in mm) of');
30  else
31      disp(nv,'final image is formed on the other
           side of air at a distance(in mm) of');
32  end
33
34  disp(-v+R,'from the centre first image is formed on
           the side from which incident rays are coming at a
           distance(in mm) of ');
35  disp(-nv+nR,'from the centre second image is formed
           on the side from which incident rays are coming
           at a distance(in mm) of ');

```

Scilab code Exa 18.19w calculation of focal length of thin lens

```

1  //developed in windows XP operating system 32bit
2  //platform Scilab 5.4.1
3  clc;clear;
4  //example 18.19w
5  //calculation of focal length of thin lens
6
7  //given data
8  R1=10; //radius of curvature(in cm) of first surface
           of given lens
9  R2=20; //radius of curvature(in cm) of second

```

```

        surface of given lens
10 mu=1.5; //refractive index of the material of lens
11
12 //calculation
13 f=1/((mu-1)*(1/R1-1/R2)); //lens maker's formula
14
15 disp(f,'focal length(in cm) of the given lens is');

```

Scilab code Exa 18.20w calculation of position of diverging mirror to obtain real

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.20w
5 //calculation of position of diverging mirror to
   obtain real image at the source itself for given
   system
6
7 //given data
8 u=-15; //object distance(in cm)
9 f=10; //focal length(in cm) of converging lens
10 fm=12; //focal length(in cm) of convex mirror
11
12 //calculation
13 v=1/((1/f)+(1/u)); //lens formula
14 LI1=2*abs(u);
15 MI1=2*abs(fm);
16 LM=LI1-MI1;
17
18 disp(LM,'on the right of the lens mirror should be
   placed at a distance(in cm) of');

```

Scilab code Exa 18.21w calculation of separation between mirror and the lens for


```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.21w
5 //calculation of separation between mirror and the
   lens for parallel emergence of the final beam
6
7 //given data
8 u=-12; //object distance(in cm)
9 f=15; //focal length(in cm) of the converging lens
10
11 //calculation
12 v=1/((1/f)+(1/u)); //lens formula
13
14 if(v<0)
15     disp(-v, 'image due to lens is formed on the left
   side of the lens at a distance(in cm) of');
16 else
17     disp(v, 'image due to lens is formed on the right
   side of the lens at a distance(in cm) of');
18 end
19
20 I1L=2*abs(v);
21 LI2=abs(f);
22 I1I2=I1L+LI2;
23
24 //let distance of mirror from I2 is x
25 //I1I2=75 cm
26 //u=- (75+x) cm
27 //v=-x cm
28 //f_mirror=-20 cm
29 //(1/v)+(1/u)=(1/f); mirror formula
30 //substituting u,v,f we get equation  $x^2+35X$ 
   -1500=0
31
32 a=1; // for above equation coefficient of  $x^2$ 
33 b=35; // for above equation coefficient of  $x^1$ 
34 c=-1500; // for above equation coefficient of  $x^0$  or

```

```

        the constant
35
36 x1=(-b+sqrt((b*b)-(4*a*c)))/(2*a); //first solution
37 x2=(-b-sqrt((b*b)-(4*a*c)))/(2*a); //second solution
38 //considering only the positive value of the
    solution ,as negative value has no physical
    meaning
39 if(x1>0)
40     disp(f+x1,'the separation(in cm) between the
        lens and the mirror is ');
41 if(x2>0)
42     disp(f+x2,'the separation(in cm) between the
        lens and the mirror is ');
43 end

```

Scilab code Exa 18.22w calculation of object distance from the lens with one side

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.22w
5 //calculation of object distance from the lens with
    one side silvered
6
7 //given data
8 v=-25; //image distance (in cm)
9 R=25; //radius of curvature of the spherical convex
    surface (in cm)
10 mu1=1; //refractive index of the medium in which
    object is kept
11 mu2=1.5; //refractive index of the medium of lens
12
13 //calculation
14 u=mu1/((mu2/v)-((mu2-mu1)/R)); //formula for
    refraction at spherical surface

```

```

15
16 disp(abs(u), 'object should be placed at a distance(
    in cm) of');

```

Scilab code Exa 18.23w calculation of location of image of an object placed in fr

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.23w
5 //calculation of location of image of an object
  placed in front of a concavo-convex lens made of
  glass
6
7 //given data
8 R1=20; //radius of curvature(in cm) of first surface
  of concavo-convex lens
9 R2=60; //radius of curvature(in cm) of second
  surface of concavo-convex lens
10 mu=1.5; //refractive index of the material of lens
11 u=-80; //object distance(in cm)
12 C1C2=160; //coaxial distance(in cm) between both the
  lenses
13
14 //calculation
15 f=1/((mu-1)*(1/R1-1/R2)); //lens maker's formula
16
17 disp(f, 'focal length(in cm) of the given concavo-
  convex lens is ');
18
19 v=1/((1/u)+(1/f)); //lens formula
20
21 if(v>0)
22     disp(v, 'first image is formed on right side of
  first lens at a distance(in cm) of');

```

```

23 else
24     disp(-v,'first image is formed on left side of
           first lens at a distance(in cm) of');
25 end
26
27 ff=f; //focal length(in cm) of the second lens same
        as first lens
28 uf=v-C1C2 //object distance(in cm) for second lens
           since image by first lens acts as object of the
           second lens
29 vf=1/((1/uf)+(1/ff)); //lens formula
30
31 if(vf>0)
32     disp(vf,'final image is formed on right side of
           second lens at a distance(in cm) of');
33 else
34     disp(-vf,'final image is formed on left side of
           second lens at a distance(in cm) of');
35 end

```

Scilab code Exa 18.24w calculation of new focal length of a lens on immersing in

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.24w
5 //calculation of new focal length of a lens on
  immersing in water
6
7 //given data
8 f=12; //focal length(in cm) of the lens
9 mu1=1; //refractive index of air
10 mu2=1.5; //refractive index of glass
11 mu3=1.33; //refractive index of water
12 //let  $(1/R1)-(1/R2)=a$  variable

```

```

13
14 //calculation
15 a=1/((mu2/mu1-1)*(f)) //refractive mediums in
    cascading
16 f_new=1/((mu2/mu3-1)*a) //refractive mediums in
    cascading
17
18 disp(f_new, 'new focal length(in cm) of a lens on
    immersing it in water is ');

```

Scilab code Exa 18.25w calculation of location of final image for an object on the

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.25w
5 //calculation of location of final image for an
    object on the axis of a cylindrical tube
    containing water closed by an equiconvex lens
6
7 //given data
8 u=-21; //object distance(in cm)
9 f=10; //focal length(in cm) of the lens
10 mu1=1; //refractive index of air
11 mu2=1.5; //refractive index of lens
12 mu3=1.33; //refractive index of water
13 //v1 image due to refraction at the first surface
14
15 //calculation
16 //from formula of refraction at the spherical
    surface
17 //  $(\frac{\mu_2}{v_1}) - (1/u) = (\mu_2 - \mu_1) / R$  (1)
18 //  $(\frac{\mu_3}{v}) - (\frac{\mu_2}{v_1}) = (\mu_3 - \mu_2) / -R$  (2)
19 //adding (1) and (2)
20 //  $(1/v) = (1/(2*R)) - (1/28)$  (3)

```

```

21 //f=1/((mu2-1)*(1/R+1/R)) refractive surfaces in
    cascade
22
23 R=2*f*(mu2-1) //refractive surfaces in cascade
24 v=1/((1/(2*R))-(1/28)) //from equation (3)
25
26 disp(v,'the image is formed inside the cylindrical
    tube at distance(in cm) of');

```

Scilab code Exa 18.26w calculation of new position of the slide of projector if t

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.26w
5 //calculation of new position of the slide of
    projector if the position of the screen is
    changed
6
7 //given data
8 v=10; //image distance(in m)
9 m=500; //lateral magnification ratio
10 d=2; //distance(in m) the screen is moved
11
12 //calculation
13 u=-v/m; //lateral magnification formula
14 f=1/((1/v)-(1/u)) //lens formula
15 vdash=v-d //effect of moving screen d m closer
16 udash=1/((1/vdash)-(1/f)) //lens formula
17
18 if(udash<0)
19     disp(-udash,'away from the lens,the slide should
        be moved by a distance(in m) of');
20 else
21     disp(udash,'towards the lens,the slide should be

```

```

        moved by a distance(in m) of ');
22 end

```

Scilab code Exa 18.27w calculation of position of the object to get a focused image

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.27w
5 //calculation of position of the object to get a
   focused image
6
7 //given data
8 v=10; //image distance(in cm) from the convex lens
9 u=-10; //object distance(in cm) from the convex lens
10 mu=1.5; //refractive index of glass
11 t=1.5; //thickness(in cm) of the glass plate
   inserted
12
13 //calculation
14 f=1/((1/v)-(1/u)) //lens formula
15 I1I=t*(1-1/mu) //shift in position(in cm) of image
   due to glass plate
16 v_new=v-I1I //lens forms image at this distance(in
   cm) from itself
17 u_new=1/((1/v_new)-(1/f)) //lens formula
18
19 disp(abs(u_new),'from the lens ,the object should be
   placed at a distance(in cm) of ');

```

Scilab code Exa 18.28aw finding the image of a distant object formed by combinati

```

1 //developed in windows XP operating system 32bit

```

```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.28aw
5 //finding the image of a distant object formed by
   combination of two convex lens by using thin lens
   formula
6
7 //given data
8 f=20; //focal length(in cm) of the given convex lens
9 d=60; //coaxial separation(in cm) between the two
   convex lenses
10 u=-(d-f); //object distance(in cm) for the second
   lens since first image is formed at focus of
   first lens
11
12 //calculation
13 v=1/((1/u)+(1/f)); //lens formula
14
15 disp(v,'final image is formed on the right of the
   second lens at a distance(in cm) of');

```

Scilab code Exa 18.28bw finding the image of a distant object formed by combinati

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 18.28bw
5 //finding the image of a distant object formed by
   combination of two convex lens by using
   equivalent lens method
6
7 //given data
8 f1=20; //focal length(in cm) of the first convex
   lens
9 f2=20; //focal length(in cm) of the first convex

```



```

    lens
10  d=60; //coaxial separation(in cm) between the two
    convex lenses
11
12  //calculation
13  F=1/((1/f1)+(1/f2)-(d/(f1*f2))); //equivalent focal
    length formula for equivalent lens method
14  D=d*F/f1; //distance(in cm) from the second lens at
    which equivalent lens is to be placed
15  //image of distant object is formed at focus of
    equivalent lens
16
17  disp(abs(D)-abs(F), 'on right side of the second lens
    ,the final image is formed at a distance(in cm)
    of ');

```

Chapter 19

Optical Instruments

Scilab code Exa 19.1 determining which boy appears taller

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.1
5 //determining which boy appears taller
6
7 //given data
8 d1=4//distance(in m) of boy1 from the eye
9 d2=5//distance(in m) of boy2 from the eye
10 h1=52//height(in inch) of boy1
11 h2=55//height(in inch) of boy2
12
13 //calculation
14 alpha1=h1/d1//angle subtended by the first boy on
   the eye
15 alpha2=h2/d2//angle subtended by the second boy on
   the eye
16 if(alpha1>alpha2)
17     printf('the first boy will look taller to the
   eye ')
18 elseif(alpha1<alpha2)
```

```

19     printf('the second boy will look taller to the
           eye')
20 else
21     printf('Both boys will appear same in height to
           the eye')
22 end

```

Scilab code Exa 19.1w calculation of the angular magnification

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.1w
5 //calculation of the angular magnification
6
7 //given data
8 f=12*10^-2//focal length(in m) of the simple
   microscope
9 D=25*10^-2//distance(in m) at which the image is
   formed away from the eye
10
11 //calculation
12 m=1+(D/f)//angular magnification
13
14 printf('the angular magnification is %3.2f',m)

```

Scilab code Exa 19.2 calculation of the angular magnification and the length of th

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.2

```

```

5 //calculation of the angular magnification and the
   length of the microscope tube
6
7 //given data
8 fo=1*10^-2//focal length(in m) of the objective lens
9 fe=2.5*10^-2//focal length(in m) of the eyepiece
10 u=-1.2*10^-2//object distance(in m)
11 D=25*10^-2//least distance(in m) for the clear
   vision
12
13 //calculation
14 v=1/((1/fo)+(1/u))//distance where the first image
   is formed ....by the lens formula
15 m=(v*D)/(u*fe)//angular magnification
16 L=v+fe//length of the tube
17
18 printf('the angular magnification is %d',round(m))
19 printf('\nthe length of the microscope tube is %3.1f
   cm',L*10^2)

```

Scilab code Exa 19.2w calculation of the object distance to obtain maximum angular magnification

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.2w
5 //calculation of the object distance to obtain
   maximum angular magnification for a normal eye
6
7 //given data
8 D=10//power(in D) of the lens
9 v=-25*10^-2//image distance(in m) i.e at the near
   point
10
11 //calculation

```

```

12 f=1/D//focal length
13 u=1/((1/v)-(1/f))//lens formula
14
15 printf('the object distance to obtain maximum
    angular magnification for a normal eye is %3.1f
    cm',u*10^2)

```

Scilab code Exa 19.3 calculation of the power of lens for the spectacles

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.3
5 //calculation of the power of lens for the
    spectacles
6
7 //given data
8 d=1.5//distance(in m) upto which the man can clearly
    see objects
9
10 //calculation
11 f=-d//focal length of the lens
12 P=1/f//definition of power of the lens
13
14 printf('the power of lens for the spectacles is %3.2
    f D',P)

```

Scilab code Exa 19.3w calculation of the position of the image linear magnification

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.3w

```

```

5 //calculation of the position of the image ,linear
   magnification and the angular magnification
6
7 //given data
8 u=-3.6*10^-2//object distance(in m)
9 f=4*10^-2//focal length(in m)
10 D=25*10^-2//least distance for clear vision
11
12 //calculation
13 v=1/((1/f)+(1/u))//lens formula
14 m=v/u//linear magnification
15 alpha=D/abs(u)//angular magnification
16
17 printf('the image distance is %d cm',v*10^2)
18 printf('\nthe linear magnification is %d',m)
19 printf('\nthe angular magnification is %3.1f',round(
   alpha))

```

Scilab code Exa 19.4w calculation of the object distance and the angular magnification

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.4w
5 //calculation of the object distance and the angular
   magnification
6
7 //given data
8 fo=1*10^-2//focal length(in m) of the objective lens
9 fe=5*10^-2//focal length(in m) of the eyepiece
10 d=12.2*10^-2//separation(in m) between the objective
   lens and the eyepiece
11 D=25*10^-2//least distance(in m) for the clear visio
12
13 //calculation

```

```

14 ve=-D//image distance for the eyepiece
15 ue=1/((1/ve)-(1/fe))//object distance for eyepiece
    ....by the lens formula
16 vo=d-abs(ue)//image distance for objective lens
17 uo=1/((1/vo)-(1/fo))//object distance for objective
    lens....by the lens formula
18 m=(vo/uo)*(1+(D/fe))//angular magnification
19
20 printf('the object should be placed at a distance of
    %3.1f cm from the objective lens to focus it
    properly ',abs(uo*10^2))
21 printf('\nthe angular magnification is %d',m)

```

Scilab code Exa 19.5w calculation of the object distance and the angular magnification

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.5w
5 //calculation of the object distance and the angular
    magnification for the least strain in the eyes
6
7 //given data
8 fo=.5*10^-2//focal length(in m) of the objective
    lens
9 fe=5*10^-2//focal length(in m) of the eyepiece
10 d=7*10^-2//separation(in m) between the objective
    lens and the eyepiece
11 D=25*10^-2//least distance(in m) for the clear
    vision
12
13 //calculation
14 v=d-fe//distance at which the first image should be
    formed
15 u=1/((1/v)-(1/fo))//lens formula for the objective

```

```

    lens
16 m=(v*D)/(u*fe)//angular magnification
17
18 printf('the object distance for the least strain in
    the eyes is %3.1f cm',abs(u*10^2))
19 printf('\nthe angular magnification for the least
    strain in the eyes is %d',m)

```

Scilab code Exa 19.6w calculation of the length of the tube and the angular magni

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.6w
5 //calculation of the length of the tube and the
    angular magnification produced by the telescope
6
7 //given data
8 fo=200*10^-2//focal length(in m) of the objective
    lens
9 fe=4*10^-2//focal length(in m) of the eyepiece
10 u=10*10^3//object distance(in m)
11
12 //calculation
13 L=fo+fe//length of the tube
14 m=-fo/fe//angular magnification
15
16 printf('the length of the tube is %d cm',L*10^2)
17 printf('\ngthe angular magnification is %d',m)

```

Scilab code Exa 19.7w calculation of the tube length magnifying power and angular

```

1 //developed in windows XP operating system 32bit

```



```

2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.7w
5 //calculation of the tube length ,magnifying power
   and angular magnification
6
7 //given data
8 fo=50*10^-2//focal length(in m) of the objective
   lens
9 fe=-5*10^-2//focal length(in m) of the eyepiece
10 u=-2//object distance(in m)
11
12 //calculation
13 L=fo-abs(fe)//length of the tube
14 m=-fo/fe//magnifying power
15 v=1/((1/fo)+(1/u))//by lens formula for the
   objective lens
16 Ldash=v-abs(fe)//tube length
17 mdash=v/abs(fe)//angular magnification
18
19 printf('the tube length for large distance viewing
   is %d cm',L*10^2)
20 printf('\\nthe magnifying power for the large
   distance viewing is %d',m)
21 printf('\\nthe tube length for viewing object at 2 m
   is %3.2f cm',Ldash*10^2)
22 printf('\\nthe angular magnification for viewing
   object at 2 m is %3.2f',mdash)

```

Scilab code Exa 19.8w calculation of the angular magnification due to the converg

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.8w

```

```

5 //calculation of the angular magnification due to
   the converging lens
6
7 //given data
8 f=50*10^-2//focal length(in m) of the converging
   lens
9 d=25*10^-2//distance(in m) from where the image can
   be seen by unaided eye
10
11 //calculation
12 //linear size = f*alpha
13 //angle formed ..... abs(beta) = f*abs(alpha)/d
14 m=-f/d//angular magnification ...m = -abs(beta)/abs(
   alpha)
15
16 printf('the angular magnification due to the
   converging lens is %d',m)

```

Scilab code Exa 19.9w calculation of the power of lens and maximum distance that

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.9w
5 //calculation of the power of lens and maximum
   distance that can be seen clearly
6
7 //given data
8 u=-25*10^-2//object distance(in m)
9 v=-40*10^-2//image distance(in m)....i.e equal to
   near point distance
10 vdash=-250*10^-2//maximum distance(in m) that an
   unaided eye can see....i.e equal to far point
   distance
11

```

```

12 //calculation
13 f=1/((1/v)-(1/u))//focal length ....by using the
    lens formula
14 P=1/f//power of the lens
15 d=1/((1/vdash)-(1/f))//maximum distance for clear
    vision.... by using the lens formula
16
17 printf('the power of the lens is %3.1f D',P)
18 printf('\nthe maximum distance upto which,the person
    will be able to see clearly is %d cm',round(abs(
    d*10^2)))

```

Scilab code Exa 19.10w calculation of the near point and the distance of the retina

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 19.10w
5 //calculation of the near point and the distance of
    the retina from the lens
6
7 //given data
8 P1=50//power1(in D) of the lens
9 P2=60//power2(in D) of the lens
10
11 //calculation
12 //for the eye in fully relaxed condition ,the focal
    length is the largest.
13 //larger the focal length ,smaller is the power of
    lens
14 if(P1<P2)
15     P=P1
16 else
17     P=P2
18 end

```

```

19 f=1/P//distance of the retina from lens ,equal to
    the focal length
20 //for eye focused at near point the power is maximum
21 if(P1>P2)
22     Pdash=P1
23 else
24     Pdash=P2
25 end
26 fdash=1/Pdash//focal length
27 v=abs(f)//image is formed at the retina
28 u=1/((1/v)-(1/fdash))//near point.....using the
    lens formula
29
30 printf('the distance of the retina from the lens is
    %d cm',f*10^2)
31 printf('\nthe near point is at %d cm',abs(u*10^2))

```

Chapter 20

Dispersion and Spectra

Scilab code Exa 20.1 calculation of the dispersive power of the flint glass

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 20.1
5 //calculation of the dispersive power of the flint
   glass
6
7 //given data
8 mur=1.613//refractive index of flint glass for the
   red light
9 mu=1.620//refractive index of flint glass for the
   yellow light
10 muv=1.632//refractive index of flint glass for the
   violet light
11
12 //calculation
13 w=(muv-mur)/(mu-1)//definition of the dispersive
   power
14
15 printf('the dispersive power of the flint glass is
   %3.4f ',w)
```

Scilab code Exa 20.1w calculation of the angular dispersion produced by a thin prism

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 20.1w
5 //calculation of the angular dispersion produced by
   a thin prism of the flint glass
6
7 //given data
8
9 mur=1.613//refractive index of flint glass for the
   red light
10 muv=1.632//refractive index of flint glass for the
   violet light
11 A=5//refracting angle(in degree)
12
13 //calculation
14 delta=(muv-mur)*A//angular dispersion
15
16 printf('the angular dispersion produced by the thin
   prism of the flint glass is %3.3f degree',delta)
```

Scilab code Exa 20.2 calculation of the dispersive power of the material of the lens

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 20.2
5 //calculation of the dispersive power of the
   material of the lens
```

```

6
7 //given data
8 fr=90//focal length(in cm) for the red light
9 fv=86.4//focal length(in cm) for the violet light
10
11 //calculation
12 //(1/f) = (mu-1) * ((1/R1) - (1/R2))
13 //muv - 1 =K/fv ..... and ..... mur - 1 = K/fr
14 //let m = muv - mur and K = 1
15 m=((1/fv)-(1/fr))
16 //muy - 1 = ((muv + mur)/2) - 1 = (K/2)*((1/fv) -
      (1/fr))
17 //let n = muy -1 and K = 1
18 n=(1/2)*((1/fv)+(1/fr))
19 //w = (muv-mur)/(mu-1) ..... definition of the
      dispersive power
20 w=m/n
21
22 printf('the dispersive power of the material of the
      lens is %3.3f',w)

```

Scilab code Exa 20.2w calculation of the angle of flint glass prism and angular d

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 20.2w
5 //calculation of the angle of flint glass prism and
      angular dispersion produced by the combination
6
7 //given data
8 A=5//angle of crown glass prism(in degree)
9 mur=1.514//refractive index of crown glass for the
      red light
10 mu=1.517//refractive index of crown glass for the

```

```

        yellow light
11  muv=1.523//refractive index of crown glass for the
        violet light
12  murdash=1.613//refractive index of flint glass for
        the red light
13  mudash=1.620//refractive index of flint glass for
        the yellow light
14  muvdash=1.632//refractive index of flint glass for
        the violet light
15
16  //calculation
17  //delta = (mu - 1) * A..... deviation produced by
        the prism
18  //D = ((mu - 1)*A) - ((mudash - 1)*Adash).... net
        deviation
19  //net deviation for the mean ray is equal to zero
20  Adash=((mu-1)/(mudash-1))*A//angle of flint glass
        prism
21  //deltav - deltar = (muv - mur)*A
        ..... for crown glass prism
22  //deltavdash - deltardash = (muvdash - murdash)*
        Adash...for flint glass prism
23  delta=((muv-mur)*A)-((muvdash-murdash)*Adash)//net
        angular dispersion
24
25  printf('the angle of flint glass prism needed is %3
        .1f degree ',Adash)
26  printf('\nthe angular dispersion produced by the
        combination is %3.4f degree ',abs(delta))

```

Scilab code Exa 20.3w calculation of the refracting angles of the two prisms

```

1  //developed in windows XP operating system 32bit
2  //platform Scilab 5.4.1
3  clc;clear;

```



```

4 //example 20.3w
5 //calculation of the refracting angles of the two
  prisms
6
7 //given data
8 w=0.03//dispersive power of crown glass
9 wdash=0.05//dispersive power of flint glass
10 delta=1//deviation(in degree) produced
11 mu=1.517//refractive index for crown glass
12 mudash=1.621//refractive index for flint glass
13
14 //calculation
15 //w = (muv - mur)/(mu - 1)..... dispersive power
16 //(muv - mur)*A = (mu-1)*w*A..... angular dispersion
17 m=((mu-1)/(mudash-1))*(w/wdash)
18 //Adash = A*m.....(1)
19 //net deviation produced is delta
20 A=delta/((mu-1)-((mudash-1)*m))//refracting angle of
  crown glass
21 Adash=A*m//refracting angle of flint glass
22
23 printf('the refracting angle of the crown prism is
  %3.1f degree',A)
24 printf('\nthe refracting angle of the flint prism is
  %3.1f degree',Adash)

```

Chapter 22

Photometry

Scilab code Exa 22.1 calculation of the luminous flux

```
1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 22.1
5 //calculation of the luminous flux
6
7 //given data
8 lambda=600//wavelength(in nm) given
9 P=10//wattage(in W) of source
10 rellum=.6//relative luminosity
11
12 //calculation
13 //1 W source of 555 nm = 685 lumen
14 lumflux=P*685*rellum//luminous flux
15
16 printf('the luminous flux is %d lumen',lumflux)
```

Scilab code Exa 22.1w calculation of the total radiant flux total luminous flux and

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 22.1w
5 //calculation of the total radiant flux ,total
   luminous flux and the luminous efficiency
6
7 //given data
8 E1=12//energy(in J) emitted by the source
9 lambda1=620*10-9//wavelength(in m) of the light1
10 E2=8//energy(in J) emitted by the source
11 lambda2=580*10-9//wavelength(in m) of the light2
12 rellum1=.35//relative luminosity of the light1
13 rellum2=.80//relative luminosity of the light2
14
15 //calculation
16 radflux=E1+E2//total radiant flux
17 lumflux1=E1*685*rellum1//luminous flux corresponding
   to the 12 W
18 lumflux2=E2*685*rellum2//luminous flux corresponding
   to the 8 W
19 lumflux=lumflux1+lumflux2//total luminous flux
20 lumeff=lumflux/radflux//luminous efficiency
21
22 printf('the total radiant flux is %d W',radflux)
23 printf('\\nthe total luminous flux is %d lumen',
   lumflux)
24 printf('\\nthe luminous efficiency is %d lumen W-1',
   lumeff)

```

Scilab code Exa 22.2w calculation of the total luminous flux emitted by the source

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;

```

```

4 //example 22.2w
5 //calculation of the total luminous flux emitted by
   the source and the total luminous intensity of
   the source
6
7 //given data
8 r=1*10^-2//radius(in m) of the circular area
9 d=2//distance(in m) from the point source
10 lumflux=2*10^-3//luminous flux(in lumen)
11
12 //calculation
13 deltaw=(%pi*r*r)/(d*d)//solid angle subtended by the
   area on the point source
14 F=(4*%pi*lumflux)/(deltaw)//total luminous flux
15 lumint=lumflux/deltaw//luminous intensity
16
17 printf('the total luminous flux emitted by the
   source is %d lumen',round(F))
18 printf('\nthe total luminous intensity of the source
   is %d cd',lumint)

```

Scilab code Exa 22.3w calculation of the luminous flux falling on a plane

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 22.3w
5 //calculation of the luminous flux falling on a
   plane
6
7 //given data
8 P=100//power(in W) input of the bulb
9 lumeff=25//luminous efficiency (in lumen W^-1)
10 A=1*10^-4//area(in m^2)
11 d=50*10^-2//distance(in m) of the area from the lamp

```

```

12
13 //calculation
14 deltaF=lumeff*P//luminous flux emitted by the bulb
15 I=deltaF/(2*%pi)
16 deltaw=A/d^2//solid angle(in sr) subtended by the
    object on the lamp
17 //I = deltaF/deltaw.....luminous intensity
18 deltaF=I*deltaw//luminous flux emitted in the solid
    angle
19
20 printf('the luminous flux falling on the plane is %3
    .2f lumen',deltaF)

```

Scilab code Exa 22.4w calculation of the illuminance at a small surface area of t

```

1 //developed in windows XP operating system 32 bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 22.4w
5 //calculation of the illuminance at a small surface
    area of the table-top
6
7 //given data
8 d=.50//distance(in m) of the point source above the
    table-top
9 lumflux=1570//luminous flux(in lumen) of the source
10 d1=.8//distance(in m)from the source
11
12 //calculation
13 I=lumflux/(4*%pi)//luminous intensity of the source
    in any direction
14
15 //E=I*cosd(theta)/r ^ 2..... illuminance
16 r=d//for point A
17 theta=0//for point A

```

```

18 EA=I*cosd(theta)/r^2//illuminance at point A
19
20 r1=d1//for point B
21 theta1=acosd(d/d1)//for point B
22 EB=I*cosd(theta1)/r1^2//illuminance at point B
23
24 printf('the illuminance at a small surface area of
    the table-top directly below the surface is %d
    lux',round(EA))
25 printf('\nthe illuminance at a small surface area of
    the table-top at a distance 0.80 m from the
    source is %d lux',EB)

```

Scilab code Exa 22.5w calculation of the luminous flux emitted into a cone of given

```

1 //developed in windows XP operating system 32bit
2 //platform Scilab 5.4.1
3 clc;clear;
4 //example 22.5w
5 //calculation of the luminous flux emitted into a
    cone of given solid angle
6
7 //given data
8 I0=160//luminous intensity(in candela) of small
    plane source
9 deltaw=0.02//solid angle(in sr)
10 theta=60//angle(in degree) made by the centre line
    of the cone with the forward normal
11
12 //calculation
13 I=I0*cosd(theta)//by using Lambert's cosine law
14 deltaF=I*deltaw//luminous flux
15
16 printf('the luminous flux emitted into a cone of
    solid angle 0.02 sr around a line making an angle

```

of 60 degree with the forward normal is %3.1f
lumen',deltaF)
