

Scilab Textbook Companion for  
Engineering Mechanics  
by A. K. Tayal <sup>1</sup>

Created by  
Danish Ansari Sadruddin  
Engineering  
Civil Engineering  
University of Pune  
College Teacher  
None  
Cross-Checked by  
Spandana

July 31, 2019

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

# Book Description

**Title:** Engineering Mechanics

**Author:** A. K. Tayal

**Publisher:** Umesh Publications

**Edition:** 14

**Year:** 2010

**ISBN:** 978-9380117386

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 Parallel Forces in a Plane	5
3 Parallel forces in a plane	15
4 Centroid Centre of mass and Centre of Gravity	24
5 General Case of Forces in a plane	27
6 Friction	31
7 Application of friction	35
8 Simple Lifting Machines	43
9 Analysis of Plane Trusses and Frames	48
10 Uniform Flexible Suspension Cables	60
12 Moment of Inertia	66
13 Principle of Virtual Work	74
14 Rectilinear Motion of a particle	76

15 Curvilinear motion of a particle	91
16 Kinetics of a Particle Work and Energy	103
17 Kinetics of a Particle Impulse and Momentum	110
18 Impact Collision of Elastic Bodies	116
19 Relative Motion	127
20 Motion of Projectile	132
21 Kinematics of rigid body	141
22 Kinetics of Rigid Body Force and Acceleration	150
23 Kinetics of Rigid Body Work and Energy	154
24 Mechanical Vibrations	155
25 Shear Force and Bending Moment	160
26 Appendix	163

# List of Scilab Codes

Exa 2.1	CFP . . . . .	5
Exa 2.2	addition of concurrent forces . . . . .	5
Exa 2.4	Equilibrium equations . . . . .	6
Exa 2.8	equilibrium of a body subjected to two forces	7
Exa 2.9	equilibrium of a body subjected to three forces	7
Exa 2.10	equilibrium of a body subjected to three forces	8
Exa 2.11	reaction at the hinge . . . . .	8
Exa 2.12	reaction at the hinge . . . . .	9
Exa 2.13	reaction at the hinge . . . . .	10
Exa 2.15	equilibrium of a body . . . . .	10
Exa 2.16	Equilibrium of a Body . . . . .	10
Exa 2.17	Equilibrium of a body . . . . .	11
Exa 2.18	Equilibrium of a body . . . . .	12
Exa 2.19	Equilibrium of a body . . . . .	12
Exa 2.20	Equilibrium of a body . . . . .	13
Exa 2.21	Equilibrium of a body . . . . .	13
Exa 2.23	Equilibrium of a body . . . . .	14
Exa 3.1	Resultant of Forces in a Plane . . . . .	15
Exa 3.2	Resultant of forces in a Plane . . . . .	15
Exa 3.3	Resultant of Forces in a Plane . . . . .	16
Exa 3.5	Resultant of Forces in a Plane . . . . .	17
Exa 3.6	Resultant of forces in a plane . . . . .	17
Exa 3.7	Resultant of forces in a plane . . . . .	18
Exa 3.8	Resultant of Forces in a Plane . . . . .	18
Exa 3.9	Resultant of Forces in a Plane . . . . .	19
Exa 3.10	Distributed Force in a Plane . . . . .	20
Exa 3.11	Distributed force in a plane . . . . .	20
Exa 3.12	Distributed force in a plane . . . . .	21

Exa 3.13	Distributed forces in a plane . . . . .	22
Exa 3.14	Distributed forces in a plane . . . . .	23
Exa 4.8	Centroid of a composite plane figure . . . . .	24
Exa 4.9	Centroid of a composite plane figure . . . . .	25
Exa 5.2	Equations of equilibrium . . . . .	27
Exa 5.3	Equations of equilibrium . . . . .	28
Exa 5.4	Equations of Equilibrium . . . . .	28
Exa 5.5	Equations of Equilibrium . . . . .	29
Exa 5.6	Equations of Equilibrium . . . . .	30
Exa 6.1	Friction . . . . .	31
Exa 6.4	Friction . . . . .	32
Exa 6.7	Friction . . . . .	32
Exa 6.9	Friction . . . . .	33
Exa 6.13	Friction . . . . .	34
Exa 7.1	Application of Friction . . . . .	35
Exa 7.2	Application of Friction . . . . .	36
Exa 7.4	Belt friction . . . . .	36
Exa 7.5	Belt friction . . . . .	37
Exa 7.6	belt friction . . . . .	38
Exa 7.7	belt friction . . . . .	38
Exa 7.8	belt friction . . . . .	39
Exa 7.9	belt friction . . . . .	40
Exa 7.10	Friction in a square threaded screw . . . . .	41
Exa 7.11	Application of Friction . . . . .	42
Exa 8.1	Simple Machine . . . . .	43
Exa 8.2	Simple machines performance . . . . .	44
Exa 8.3	Performance of Machine . . . . .	45
Exa 8.4	Performance of a machine . . . . .	46
Exa 8.5	Simple screw jack . . . . .	46
Exa 9.1	Axial forces in members of Truss . . . . .	48
Exa 9.2	axial forces in members of truss . . . . .	49
Exa 9.3	Axial Forces in members of the truss . . . . .	50
Exa 9.5	Axial Forces in members of the truss . . . . .	51
Exa 9.6	Axial Forces in members of the truss . . . . .	53
Exa 9.10	Axial Forces in members of the truss . . . . .	54
Exa 9.12	Axial Forces in members of the truss . . . . .	55
Exa 9.13	Axial Forces in members of the truss . . . . .	56
Exa 9.14	Axial Forces in members of the truss . . . . .	57

Exa 9.15	Axial Forces in members of the truss . . . . .	58
Exa 10.1	Cable subjected to concentrated loads . . . . .	60
Exa 10.2	Cables subjected to concentrated loads . . . . .	61
Exa 10.3	Cables uniformly loaded per unit horizontal distance . . . . .	62
Exa 10.4	Cables uniformly loaded per unit horizontal distance . . . . .	63
Exa 10.5	Cables uniformly loaded per unit horizontal distance . . . . .	64
Exa 10.6	Catenary Cables . . . . .	65
Exa 12.7	Moment of Inertia of an area of a plane figure with respect to an axis in its plane . . . . .	66
Exa 12.8	Moment of Inertia of a Composite area or hollow section . . . . .	67
Exa 12.9	Moment of Inertia of a Composite area or hollow section . . . . .	68
Exa 12.10	Moment of Inertia of a Composite area or hollow section . . . . .	69
Exa 12.14	Product of Inertia . . . . .	71
Exa 12.15	Principal Moment of Inertia . . . . .	71
Exa 13.1	Application of Principle of Virtual Work . . . . .	74
Exa 13.7	Application of Principle of Virtual Work . . . . .	74
Exa 14.3	Displacement velocity and acceleration of connected bodies . . . . .	76
Exa 14.4	Velocity time relationship . . . . .	77
Exa 14.5	Displacement time relationship . . . . .	77
Exa 14.6	Displacement time relationship . . . . .	78
Exa 14.7	Displacement time relationship . . . . .	79
Exa 14.9	Displacement time relationship . . . . .	80
Exa 14.10	Distance travelled by a particle in the $n$ th second . . . . .	80
Exa 14.11	Variable acceleration . . . . .	81
Exa 14.12	Variable acceleration . . . . .	82
Exa 14.15	D Alemberts Principle . . . . .	82
Exa 14.16	D Alemberts Principle . . . . .	83
Exa 14.17	D Alamberts Principle . . . . .	83
Exa 14.18	Motion of two Bodies . . . . .	84
Exa 14.19	Motion of two Bodies . . . . .	85



Exa 14.20	Motion of two Bodies . . . . .	85
Exa 14.21	Motion of two Bodies . . . . .	86
Exa 14.22	Motion of two Bodies . . . . .	87
Exa 14.23	Motion of two Bodies . . . . .	87
Exa 14.24	Motion of two Bodies . . . . .	88
Exa 14.25	Motion of two Bodies . . . . .	89
Exa 14.30	Motion of two Bodies . . . . .	89
Exa 15.2	Components of Acceleration . . . . .	91
Exa 15.3	Components of Acceleration . . . . .	92
Exa 15.5	Motion of a particle on a curved frictionless path . . . . .	92
Exa 15.6	Motion of a particle on a curved frictionless path . . . . .	94
Exa 15.7	Components of motion . . . . .	95
Exa 15.9	Equations of dynamic equilibrium . . . . .	96
Exa 15.10	Equations of dynamic equilibrium . . . . .	97
Exa 15.12	Motion of particle on curved frictionless path . . . . .	97
Exa 15.13	motion of a particle on curved frictionless path . . . . .	98
Exa 15.15	Motion of a particle in a curved frictionless path . . . . .	98
Exa 15.16	Motion of a particle in a curved frictionless path . . . . .	99
Exa 15.18	Motion of vehicles on leveled and banked roads . . . . .	100
Exa 15.19	Motion of vehicles on leveled and banked roads . . . . .	100
Exa 15.20	Motion of vehicles on leveled and banked roads . . . . .	101
Exa 15.21	Motion of vehicles on leveled and banked roads . . . . .	102
Exa 16.1	Work of the Force of Spring . . . . .	103
Exa 16.3	Work and energy principle for a system of particles . . . . .	104
Exa 16.4	Power . . . . .	104
Exa 16.5	Principle of conservation of energy . . . . .	105
Exa 16.6	Principle of conservation of energy . . . . .	106
Exa 16.7	Principle of conservation of energy . . . . .	107
Exa 16.9	Principle of conservation of energy . . . . .	107
Exa 16.10	Principle of work and energy . . . . .	108
Exa 16.11	Principle of conservation of energy . . . . .	109
Exa 17.1	Principle of impulse and momentum . . . . .	110
Exa 17.2	Principle of impulse and momentum . . . . .	111

Exa 17.3	Principle of impulse and momentum . . . . .	111
Exa 17.5	Principle of impulse and momentum . . . . .	112
Exa 17.6	Principle of impulse and momentum . . . . .	113
Exa 17.8	Principle of conservation of angular momentum . . . . .	114
Exa 18.1	Principle of conservation of Momentum . . . . .	116
Exa 18.2	Principle of conservation of Momentum . . . . .	117
Exa 18.3	Principle of conservation of Momentum . . . . .	118
Exa 18.4	Principle of conservation of Momentum . . . . .	118
Exa 18.5	Motion of ball . . . . .	120
Exa 18.6	Principle of conservation of Energy . . . . .	120
Exa 18.7	Principle of conservation of Energy . . . . .	121
Exa 18.8	Principle of conservation of Energy . . . . .	121
Exa 18.9	Principle of conservation of Momentum . . . . .	123
Exa 18.10	Principle of conservation of Momentum . . . . .	124
Exa 18.11	Principle of conservation of Energy Momentum and work and energy . . . . .	125
Exa 19.1	Relative Velocity . . . . .	127
Exa 19.2	Relative Velocity . . . . .	127
Exa 19.3	Relative Velocity . . . . .	128
Exa 19.4	Relative Velocity . . . . .	129
Exa 20.1	Motion of Projectile . . . . .	132
Exa 20.2	Motion of Projectile . . . . .	133
Exa 20.3	Motion of Projectile . . . . .	134
Exa 20.5	Motion of Projectile . . . . .	134
Exa 20.6	Motion of Projectile . . . . .	135
Exa 20.7	Motion of Projectile . . . . .	136
Exa 20.8	Motion of Projectile . . . . .	137
Exa 20.9	Motion of Projectile . . . . .	138
Exa 20.10	Motion of Projectile . . . . .	139
Exa 21.1	Linear and angular velocity linear and angular acceleration in rotation . . . . .	141
Exa 21.2	Absolute and relative velocity in plane motion . . . . .	142
Exa 21.3	Absolute and relative velocity in plane motion . . . . .	142
Exa 21.4	Absolute and relative velocity in plane motion . . . . .	143
Exa 21.5	Absolute and relative velocity in plane motion . . . . .	144
Exa 21.6	Instantaneous Centre of rotation in plane motion . . . . .	145

Exa 21.7	Instantaneous Centre of rotation in plane motion . . . . .	145
Exa 21.8	Instantaneous Centre of rotation in plane motion . . . . .	146
Exa 21.9	Instantaneous Centre of rotation in plane motion . . . . .	147
Exa 21.11	Instantaneous Centre of rotation in plane motion . . . . .	147
Exa 21.12	Instantaneous Centre of rotation in plane motion . . . . .	148
Exa 21.13	Instantaneous Centre of rotation in plane motion . . . . .	149
Exa 22.1	Relation between the translatory motion and rotary motion of a body in plane motion . .	150
Exa 22.2	Relation between the translatory motion and rotary motion of a body in plane motion . .	151
Exa 22.5	Relation between the translatory motion and rotary motion of a body in plane motion . .	151
Exa 22.8	Relation between the translatory motion and rotary motion of a body in plane motion . .	152
Exa 22.10	Relation between the translatory motion and rotary motion of a body in plane motion . .	153
Exa 23.2	Principle of work and energy for a rigid body	154
Exa 24.1	Simple Harmonic Motion . . . . .	155
Exa 24.2	Simple Harmonic Motion . . . . .	156
Exa 24.5	Equivalent spring constant . . . . .	157
Exa 24.10	Pendulum Motion . . . . .	157
Exa 24.11	Pendulum Motion . . . . .	158
Exa 24.12	Pendulum Motion . . . . .	159
Exa 25.5	Shear Force and Bending Moment . . . . .	160
Exa 25.7	Shear Force and Bending Moment . . . . .	161
Exa 26.1	Appendix . . . . .	163
Exa 26.2	Appendix . . . . .	164
Exa 26.3	Appendix . . . . .	165
Exa 26.4	Appendix . . . . .	165
Exa 26.5	Appendix . . . . .	166
Exa 26.6	Appendix . . . . .	166
Exa 26.7	Appendix . . . . .	167

Exa 26.8	Appendix . . . . .	167
Exa 26.9	Appendix . . . . .	168
Exa 26.10	Appendix . . . . .	169
Exa 26.11	Appendix . . . . .	170
Exa 26.12	Appendix . . . . .	170
Exa 26.13	Appendix . . . . .	171
Exa 26.14	Appendix . . . . .	172
Exa 26.15	Appendix . . . . .	173
Exa 26.16	Appendix . . . . .	175
Exa 26.17	Appendix . . . . .	175
Exa 26.18	Appendix . . . . .	176
Exa 26.19	Appendix . . . . .	177

## Chapter 2

# Parallel Forces in a Plane

Scilab code Exa 2.1 CFP

```
1 //Initialization of variables
2 P=50 //N
3 Q=100 //N
4 beta=150 //degree // angle between P & the
   horizontal
5 //Calculations
6 R=sqrt(P^2+Q^2-(2*P*Q*cosd(beta))) // using the
   Trigonometric solution
7 Alpha=asind((sind(beta)*Q)/R)+15
8 //Result
9 clc
10 printf('The magnitude of resultant is %f Newton (N)
   \n',R)
11 printf('The direction of resultant is %f degree \n',
   Alpha)
```

---

Scilab code Exa 2.2 addition of concurrent forces

```

1 //Initilization of variables
2 P=50 //N
3 Q=100 //N
4 beta=15 //degree // angle between P& the horizontal
5 theta=45 //degree // angle between the resultant (R)
   & the horizontal
6 //Calculations
7 Rx=P*cosd(beta)+Q*cosd(theta) //N
8 Ry=P*sind(beta)+Q*sind(theta) //N
9 R=sqrt((Rx^2)+(Ry^2)) //N
10 alpha=atand(Ry/Rx) //degree
11 //Results
12 clc
13 printf('The magnitude of the resultant is %f N \n',R
   )
14 printf('The ange of the resultant with x-axis is %f
   degree \n',alpha)

```

---

#### Scilab code Exa 2.4 Equilibrium equations

```

1 //Initilization of variables
2 Tac=3.5 //kN
3 Tbc=3.5 //kN
4 alpha=20 //degree //angle made by Tac with -ve X
   axis
5 beta=50 //degree //angle made by Tbc with +ve X axis
6 //Calculations
7 theta=atand(((Tac*sind(alpha))+(Tbc*sind(beta)))/((
   Tac*cosd(alpha))-(Tbc*cosd(beta)))) //degree
8 P=Tac*(cosd(alpha)-cosd(beta))/(cosd(theta)) //kN //
   from eq'n 1
9 //Results
10 clc
11 printf('The maximum force that can be applied is %f
   kN \n',P)

```

```
12 printf('The direction of applied force is %f degree
        \n',theta)
```

---

**Scilab code Exa 2.8** equilibrium of a body subjected to two forces

```
1 //Initialization of variables
2 lAB=0.4 //m
3 lBC=0.3 //m
4 //Calculations
5 alpha=atand(lAB/lBC) //degree
6 //Results
7 clc
8 printf('The angle wich the force should make with
        the horizontal to keep the edge AB of the body
        vertical %f degree \n',alpha) //here alpha=theta
```

---

**Scilab code Exa 2.9** equilibrium of a body subjected to three forces

```
1 //Initialization of variables
2 F=1000 //N
3 lAB=0.5 //m
4 lDC=0.25 //m //length of the perpendicular drawn
        from point C to AB
5 //Calculations
6 lAC=sqrt((0.3)^2+(0.25)^2) //m
7 lBC=sqrt((0.20)^2+(0.25)^2) //m
8 Sac=(lAC*F)/(lAB) //N //by law of concurrent forces
9 Sbc=(lBC*F)/(lAB) //N //by law of concurrent forces
10 //Results
11 clc
12 printf('The axial force in the bar AC(by aw of
        concurrent forces) is %f N \n',Sac)
```

```
13 printf('The axial force in the bar BC(by aw of
    concurrent forces) is %f N \n',Sbc)
```

---

**Scilab code Exa 2.10** equilibrium of a body subjected to three forces

```
1 //Initilization of variables
2 F3=500 //N
3 alpha=60 //degree //angle made by F3 with F2
4 beta=40 //degree //angle made by F1 with F3
5 theta=80 //degree //angle made by F1 with F2
6 //Calculations
7 // Solving by using law of sines
8 F1=(F3*sind(alpha)/sind(theta)) //N //by law of
    sines
9 F2=(F3*sind(beta)/sind(theta)) //N //by law of sines
10 //Resuts
11 clc
12 printf('The force F1 is %f N \n',F1)
13 printf('The force F2 is %f N \n',F2)
```

---

**Scilab code Exa 2.11** reaction at the hinge

```
1 //Initilization of variables
2 P=5000 //N
3 lAB=5 //m
4 lOB=1.443 // m
5 alpha=30 //degree //angle made by force P with the
    beam
6 //Calculations
7 theta=atand(lOB/lAB) // degree // eq'n 1
8 Xa=(P*cosd(alpha)) //N //using eq'n 4
9 Ya=Xa*tand(theta) //N // from eq'n 3 & 4
```



```

10 Rb=P*sind(alpha)-Ya // N from eq'n 5// substuting
    value of Ya in eq'n 5
11 Ra=sqrt((Xa^2)+(Ya^2)) //N
12 //Results
13 clc
14 printf('The X component of reaction at A is %f N \n',
    ,Xa)
15 printf('The Y component of reaction at A is %f N \n',
    ,Ya)
16 printf('The reaction at support A is %f N \n',Ra)
17 printf('The reaction at support B is %f N \n',Rb)
18 // The decimal point error might cause a small
    discrepancy in the answers

```

---

#### Scilab code Exa 2.12 reaction at the hinge

```

1 //Initilization of variables
2 W=1000 //N
3 OD=0.4 //m
4 AD=0.3 //m
5 AO=0.5 //m //AO=sqrt((0.4)^2+(0.3)^2)
6 //Calculations
7 Ra=W*AO/OD //N // The answer of Ra in the textbook
    is incorrect
8 Rc=W*AD/OD //N
9 alpha=atand(OD/AD) //degree
10 //Results
11 clc
12 printf('The reaction at support A is %f N \n',Ra)
13 printf('The reaction at support B is %f N \n',Rc)
14 printf('The angle that Rc makes with horizontal %f
    degree \n',alpha)

```

---

### Scilab code Exa 2.13 reaction at the hinge

```
1 //Initialization of variables
2 W=2500 //N //This load acts at point B and C.
3 alpha=30 //degree // angle made by T1 with +ve y-
    axis & T2 with +ve x-axis
4 //Calculations
5 T2=W-(((cosd(alpha))^2/(sind(alpha)))-(sind(alpha)))
    // N // substiting eq'n 1 in 2
6 T1=(T2*cosd(30))/(sind(30)) //N // using eq'n 1
7 T3=T2 //N // By equilibrium eq'n at point C(sumFx=0)
8 //Results
9 clc
10 printf('Tension in portion AB is %f N \n',T1)
11 printf('Tension in portion BC is %f N \n',T2)
12 printf('Tension in portion CD is %f N \n',T3)
```

---

### Scilab code Exa 2.15 equilibrium of a body

```
1 //Initialization of variables
2 d=0.6 //m //diameter of the wheel
3 r=0.3 //m //radius of the wheel
4 W=1000 //N //weight of the wheel
5 h=0.15 //m //height of rectangular block
6 //Calculations
7 theta=atand((sqrt(h))/(sqrt(d-h)))
8 P=(W*tand(theta)) //N // dividing eq'n 1 & 2
9 //Results
10 clc
11 printf('The force P so that the wheel is just to
    roll over the block is %f N \n',P)
```

---

### Scilab code Exa 2.16 Equilibrium of a Body

```

1 //Initilization of variables
2 Soa=1000 //N (tension)
3 alpha=45 //degree //where alpha=(360/8)
4 theta=67.5 //degree //angle made by bar AO with AB &
  AH
5 //Calculutions
6 Sab=Soa*(sind(theta)/sind(alpha)) // N // Using law
  of sines
7 Sah=Sab //N
8 Sob=(Sab*sind(180-2*(theta)))/sind(theta) //N
9 //Results
10 clc
11 printf('The axial force in the bar AB is %f N \n',
  Sab) //Compression
12 printf('The axial force in the bar OB is %f N \n',
  Sob) //Tension

```

---

#### Scilab code Exa 2.17 Equilibrium of a body

```

1 //Initilization of variables
2 W=500 //N //weight of cylinder
3 alpha=25 //degree //angle made by OA with horizontal
4 beta=65 //degree //angle made by OB with horizontal
5 theta=90 //degree // theta=(alpha+beta)
6 //Calculutions
7 Ra=(W*sind(beta))/sind(theta) //N //from equilibrium
  eq'n
8 Rb=(W*sind(alpha))/sind(theta) //N //from
  equilibrium eqn's
9 //Results
10 clc
11 printf('The reaction at A is %f N \n',Ra)
12 printf('The reaction at B is %f N \n',Rb)

```

---

### Scilab code Exa 2.18 Equilibrium of a body

```
1 //Initialization of variables
2 Wa=1000 //N //weight of sphere A
3 Wb=400 //N //weight of sphere B
4 Ra=0.09 //m //radius of sphere A
5 Rb=0.05 //m //radius of sphere B
6 theta=33.86 //degree //angle made by Rq with Wb
7 alpha=60 //degree //angle made by Rl with horizontal
8 //Calculations
9 Rq=Wb/cosd(theta) //N //using sum Fy=0 for sphere B
10 Rp=Rq*sind(theta) //N //using sum Fx=0 for sphere B
11 Rl=(Rq*sind(theta))/sind(alpha) //N //using sum Fx=0
    for sphere A
12 Rn=((Wa)+(Rq*cosd(theta))-(Rl*cosd(alpha))) //N
13 //Results
14 clc
15 printf('The reaction at point P is %f N \n',Rp)
16 printf('The reaction at point L is %f N \n',Rl)
17 printf('The reaction at point N is %f N \n',Rn)
```

---

### Scilab code Exa 2.19 Equilibrium of a body

```
1 //Initialization of variables
2 P=50 //N
3 Q=100 //N
4 alpha=30 //degree //angle made by Rq with +ve Y-axis
5 //Calculations
6 theta=atand((P*cotd(alpha)-Q*tand(alpha))/(P+Q)) //
    degree
7 T=Q/(cosd(theta)*cotd(alpha)-sind(theta)) //N
8 //Results
```

```

9  clc
10 printf('The tension in the string is %f N \n',T)
11 printf('The angle wich the string makes with the
    horizontal when the system is in equilibrium is
    %f N \n',theta)

```

---

### Scilab code Exa 2.20 Equilibrium of a body

```

1  //Initilization of variables
2  theta1=50.5 //degree //is the angle made between BC
    & and BE
3  theta2=36.87 //degree //is te angle ade between BA &
    BE
4  g=9.81 //m/s^2
5  Wa=15*g //N
6  Wb=40*g //N
7  Wc=20*g //N
8  //Calculations
9  R2=Wc/(sind(theta1)) //N //from F.B.D of cylinder C(
    sum Fy=0)
10 R4=(Wb+R2*sind(theta1))/sind(theta2) //N //from F.B.
    D of cylinder B(sum Fy=0)
11 R6=R4*cosd(theta2) //N //from F.B.D of cylinder A(
    sum Fx=0)
12 //Results
13 clc
14 printf('The reaction between the cylinder A and the
    wall of the channel is %f N \n',R6)

```

---

### Scilab code Exa 2.21 Equilibrium of a body

```

1  //Initilazation of variables
2  F=1000 //N

```

```

3 theta=30 //degree //angle made by the force with the
   beam AB
4 Lab=3 //m
5 Lae=2 //m
6 Lce=1 //m
7 //Calculations
8 Re=(F*Lab*sind(theta))/Lae //N //Taking moment at A
9 Rd=(Re*Lce)/(Lab*sind(theta)) //N //Taking moment
   about C
10 //Results
11 clc
12 printf('The reaction at D due to force of 1000 N
   acting at B is %f N \n',Rd)

```

---

#### Scilab code Exa 2.23 Equilibrium of a body

```

1 //Initilization of variables
2 W=1000 //N
3 r=0.30 //m //radius of the wheel
4 h=0.15 //m //height of the obstacle
5 //Calculations
6 theta=asind(1) //degree //P is mini when sin(theta)
   =1 from eq'n of P
7 Pmini=(W*sqrt((2*r*h)-(h^2)))/(r*sind(theta)) //N
8 //Results
9 clc
10 printf('The least force required to just turn the
   wheel over the block is %f N \n',Pmini)
11 printf('The angle wich should be made by Pmini with
   AC is %f degree \n',theta)

```

---

# Chapter 3

## Parallel forces in a plane

Scilab code Exa 3.1 Resultant of Forces in a Plane

```
1 //Initilization of variables
2 W=1000 //N
3 Lab=1 //m
4 Lac=0.6 //m
5 theta=60 //degree //angle made by the beam with the
   horizontal
6 //Calculations
7 Q=(W*Lac*cosd(theta))/(Lab*cosd(theta)) //N // from
   eq'n 2
8 P=W-Q //N // from eq'n 1
9 //Results
10 clc
11 printf('The load taken by man P is %f N \n',P)
12 printf('The load taken by man Q is %f N \n',Q)
```

---

Scilab code Exa 3.2 Resultant of forces in a Plane

```
1 //Initilization of variables
```

```

2 F=1000 //N
3 Lab=1 //m
4 Lbc=0.25 //m
5 Lac=1.25 //m
6 //Calculations
7 Rb=(F*Lac)/Lab //N // from eq'n 2
8 Ra=Rb-F //N // fom eq'n 1
9 //Results
10 clc
11 printf('The reaction (downwards)at support A is %f N
        \n',Ra)
12 printf('The reaction (upwards)at support B is %f N \
        n',Rb)

```

---

### Scilab code Exa 3.3 Resultant of Forces in a Plane

```

1 //Inilitization of variables
2 Lab=12 //m
3 Mc=40 //kN-m
4 Md=10 //kN-m
5 Me=20 //kN-m
6 Fe=20 //kN //force acting at point E
7 //Calculations
8 Xa=-(Fe) //kN //take sum Fx=0
9 Rb=(Md+Me-Mc)/Lab //N //take moment at A
10 Ya=-Rb //N //take sum Fy=0
11 //Results
12 clc
13 printf('The vertical reaction (upwards) at A is %f
        kN \n',Ya)
14 printf('The horizontal reaction (towards A) is %f kN
        \n',Xa)
15 printf('The reaction (downwards) at B is %f kN \n',
        Rb)

```

---



### Scilab code Exa 3.5 Resultant of Forces in a Plane

```
1 // Initialization of variables
2 W=1000 //N
3 Lad=7.5 //m
4 Lae=1.5 //m
5 La1=3.75 //m //distance of 1st 1000N load from pt A
6 La2=5 //m //distance of 2nd 1000N load from pt A
7 La3=6 //m // distance of 3rd 1000N load from pt A
8 // Calculations (part1)
9 //using matrix to solve the given eqn's 1 & 2
10 A=[1 -2.5;3.5 -5]
11 B=[1000;7250]
12 C=inv(A)*B
13 //Calculations (part 2)
14 //Consider combined F.B.D of beams AB,BC &CD. Take
    moment at A
15 Re=((W*La1)+(W*La2)+(W*La3)+(C(2)*Lad)-(C(1)*La3))/
    Lae //N
16 Ra=C(2)-Re-C(1)+(3*W) //N //Taking sum of forces in
    Y direction
17 //Results
18 clc
19 printf('The reaction at F i.e Rf is %f N \n',C(1))
20 printf('The reaction at D i.e Rd is %f N \n',C(2))
21 printf('The reaction at pt E i.e Re is %f N \n',Re)
22 printf('The reaction at pt A i.e Ra is %f N \n',Ra)
    //acting vertically downwards
```

---

### Scilab code Exa 3.6 Resultant of forces in a plane

```
1 // Initialization of variables
```

```

2 W=100 // N //force acting at D
3 AB=50 // N // weight of bar ab
4 CD=50 // N // weight of bar cd
5 // Calculations
6 // From the derived expression the value of the
   angle is given as,
7 theta=atand(5/17.5) //degrees
8 // Results
9 clc
10 printf('The angle theta is %f degrees \n',theta)

```

---

### Scilab code Exa 3.7 Resultant of forces in a plane

```

1 //Initialization of variables
2 Ws=2 //kN //weight of scooter
3 Wd=0.5 //kN //weight of driver
4 Lab=1 //m
5 Led=0.8 //m
6 Leg=0.1 //m
7 //Calculations
8 Rc=((2*Leg)+(Wd*Led))/Lab //kN //take moment at E
9 Ra=(2+Wd-Rc)/2 // kN // as Ra=Rb,(Ra+Rb=2*Ra)
10 Rb=Ra // kN
11 //Results
12 clc
13 printf('The reaction at wheel A is %f kN \n',Ra)
14 printf('The reaction at wheel B is %f kN \n',Rb)
15 printf('The reaction at wheel C is %f kN \n',Rc)

```

---

### Scilab code Exa 3.8 Resultant of Forces in a Plane

```

1 //Initialization of variables
2 W1=15 //N //up

```

```

3 W2=60 //N //down
4 W3=10 //N //up
5 W4=25 //N //down
6 Lab=1.2 //m
7 Lac=0.4 //m
8 Lcd=0.3 //m
9 Ldb=0.5 //m
10 Lad=0.7 //m
11 Leb=0.417 //m //Leb=Lab-x
12 // Calculations
13 //(a) A single force
14 Ry=W1-W2+W3-W4 //N //take sum Fy=0
15 x=(-W2*Lac)+(W3*Lad)-(W4*Lab)/(Ry) //m
16 //(b) Single force moment at A
17 Ma=(Ry*x) //N-m
18 // Single force moment at B
19 Mb=W2*Leb //N-m
20 // Results
21 clc
22 printf('The reaction for single force (a) is %f N \n
        ',Ry)
23 printf('The distance of Ry from A is %f m \n',x)
24 printf('The moment at A is %f N-m \n',Ma)
25 printf('The moment at B is %f N-m \n',Mb)

```

---

### Scilab code Exa 3.9 Resultant of Forces in a Plane

```

1 // Initialization of variables
2 Ra=5000 //N
3 Ma=10000 //Nm
4 alpha=60 //degree //angle made by T1 with the pole
5 beta=45 //degree //angle made by T2 with the pole
6 theta=30 //degree //angle made by T3 with the pole
7 Lab=6 //m
8 Lac=1.5 //m

```

```

9 Lcb=4.5 //m
10 // Calculations
11 T3=Ma/(4.5*sind(theta)) //N //take moment at B
12 // Now we use matrix to solve eqn's 1 & 2
    simultaneously ,
13 A=[-0.707 0.8666;0.707 0.5]
14 B=[2222.2;8848.8]
15 C=inv(A)*B
16 // Results
17 clc
18 printf('Tension in wire 1 i.e T1 is %f N \n',C(2))
19 printf('Tension in wire 2 i.e T2 is %f N \n',C(1))
20 printf('Tension in wire 3 i.e T3 is %f N \n',T3)

```

---

#### Scilab code Exa 3.10 Distributed Force in a Plane

```

1 // Initialization of variables
2 w=2000 //N/m
3 Lab=3 //m
4 // Calculations
5 W=w*Lab/2 //N// Area under the curve
6 Lac=(2/3)*Lab //m//centroid of the triangular load
    system
7 Rb=(W*Lac)/Lab //N //sum of moment at A
8 Ra=W-Rb //N
9 // Results
10 clc
11 printf('The resultant of the distributed load lies at
    %f m \n',Lac)
12 printf('The reaction at support A is %f N \n',Ra)
13 printf('The reaction at support B is %f N \n',Rb)

```

---

#### Scilab code Exa 3.11 Distributed force in a plane

```

1 //Initiization of variables
2 w=1500 //N/m
3 x=4 //m
4 L=4 //m
5 //Calculations
6 k=x^2/w //m^3/N
7 //Solving the intergral we get
8 W=L^3/(3*k) //N
9 x_bar=L^4/(4*k*W) //m
10 //Result
11 clc
12 printf("The resultant is %f N and the line of action
        of the force is %f m",W,x_bar)

```

---

### Scilab code Exa 3.12 Distributed force in a plane

```

1 // Initalization of variables
2 w1=1.5 //kN/m // intensity of varying load at the
    starting point of the beam
3 w2=4.5 //kN/m // intensity of varying load at the
    end of the beam
4 l=6 //m // ength of the beam
5 // Calculations
6 // The varying load distribution is divided into a
    rectangle and a right angled triangle
7 W1=w1*l //kN // where W1 is the area of the load
    diagram(rectangle ABED)
8 x1=l/2 //m // centroid of the rectangular load
    system
9 W2=(w2-w1)*l/2 //kN // where W1 is the area of the
    load diagram(triangle DCE)
10 x2=2*l/3 //m // centroid of the triangular load
    system
11 W=W1+W2 //kN // W is the resultant
12 x=((W1*x1)+(W2*x2))/W //m // where x is the distance

```

```

        where the resultant lies
13 //Results
14 clc
15 printf('The resultant of the distributed load system
        is %f kN \n',W)
16 printf('The line of action of the resulting load is
        %f m \n',x)

```

---

### Scilab code Exa 3.13 Distributed forces in a plane

```

1 // Initiization of variables
2 W1=10 //kN //point load acting at D
3 W2=20 //kN // point load acting at C at an angle of
    30 degree
4 W3=5 //kN/m // intensity of udl acting on span EB of
    4m
5 W4=10 //kN/m // intensity of varying load acting on
    span BC of 3m
6 M=25 //kN-m // moment acting at E
7 theta=30 //degree // angle made by 20 kN load with
    the beam
8 Lad=2 //m
9 Leb=4 //m
10 Laf=6 //m //distance between the resultant of W3 &
    point A
11 Lac=11 //m
12 Lag=9 //m //distance between the resultant of W4 and
    point A
13 Lbc=3 //m
14 Lab=8 //m
15 // Calculations
16 Xa=20*cosd(theta) //kN // sum Fx=0
17 Rb=((W1*Lad)+(-M)+(W3*Leb*Laf)+(W2*sind(theta)*Lac)
    +((W4*Lbc*Lag)/2))/Lab //kN // taking moment at A
18 Ya=W1+(W2*sind(theta))+(W3*Leb)+(W4*Lbc/2)-Rb //kN

```

```

    // sum Fy=0
19 Ra=sqrt(Xa^2+Ya^2) //kN // resultant at A
20 //Results
21 clc
22 printf('The horizontal reaction at A i.e Xa is %f kN
    \n',Xa)
23 printf('The vertical reaction at A i.e Ya is %f kN \
    n',Ya)
24 printf('The reaction at A i.e Ra is %f kN \n',Ra)
25 printf('The reaction at B i.e Rb is %f kN \n',Rb)

```

---

#### Scilab code Exa 3.14 Distributed forces in a plane

```

1 // Initalization of variables
2 h=4 //m //height of the dam wall
3 rho_w=1000 // kg/m^3 // density of water
4 rho_c=2400 // kg/m^3 // density of concrete
5 g=9.81 // m/s^2
6 // Calculations
7 P=(rho_w*g*h^2)/2 // The resultant force due to
    water pressure per unit length of the dam
8 x=(2/3)*h //m // distance at which the resutant of
    the triangular load acts
9 b=sqrt((2*P*h)/(3*h*rho_c*g)) // m // eq'n required
    to find the minimum width of the dam
10 // Results
11 clc
12 printf('The minimum width which is to be provided to
    the dam to prevent overturning about point B is
    %f m \n',b)

```

---

## Chapter 4

# Centroid Centre of mass and Centre of Gravity

Scilab code Exa 4.8 Centroid of a composite plane figure

```
1 // Initialization of variables
2 b1=20 //cm // width of top flange
3 t1=5 //cm // thickness of top flange
4 b2=5 //cm // width of web
5 t2=15 //cm // thickness or height of the web
6 b3=30 //cm // width of bottom flange
7 t3=5 //cm // thickness of bottom flange
8 // Calculations
9 A1=b1*t1 //cm^2 // area of bottom flange
10 A2=b2*t2 //cm^2 // area of the web
11 A3=b3*t3 //cm^2 // area of top flange
12 y1=t3+t2+(t1/2) //cm // y co-ordinate of the
    centroid of top flange
13 y2=t3+(t2/2) //cm // y co-ordinate of the centroid
    of the web
14 y3=t3/2 //cm // y co-ordinate of the centroid of
    the bottom flange
15 y_c=((A1*y1)+(A2*y2)+(A3*y3))/(A1+A2+A3) //cm //
    where y_c is the centroid of the un-symmetrical I
```



```

    -section
16 // Results
17 clc
18 printf('The centroid of the un equal I-section is %f
        cm \n',y_c)

```

---

#### Scilab code Exa 4.9 Centroid of a composite plane figure

```

1 // Initalization of variables
2 // The given section is Z-section which is un-
  symmetrycal about both the axis
3 b1=20 //cm // width of bottom flange
4 t1=5 //cm // thickness of the bottom flange
5 b2=2.5 //cm // thickness of the web of the flange
6 t2=15 //cm // depth of the web
7 b3=10 //cm // width of the top flange
8 t3=2.5 //cm // thickness of the top flange
9 // Calculations
10 // Respective areas
11 A1=b1*t1 // cm^2 // area of the bottom flange
12 A2=b2*t2 // cm^2 // area of the web
13 A3=b3*t3 // cm^2 // area of the top-flange
14 // first we calculate the x co-ordinate of the
  centroid
15 x1=b3-b2+(b1/2) //cm // for the bottom flange
16 x2=b3-(b2/2) //cm // for the web
17 x3=b3/2 //cm // for the top flange
18 x_c=((A1*x1)+(A2*x2)+(A3*x3))/(A1+A2+A3) //cm
19 // secondly we calculate the y co-ordinate of the
  centroid
20 y1=t1/2 //cm // for the bottom flange
21 y2=t1+(t2/2) //cm // for the web
22 y3=t1+t2+(t3/2) //cm // for the top flange
23 y_c=((A1*y1)+(A2*y2)+(A3*y3))/(A1+A2+A3) // cm
24 // Results

```

```
25 clc
26 printf('The centroid of the cross-sectional area of
    a Z-section about x-axis is %f cm \n',x_c)
27 printf('The centroid of the cross-sectional area of
    a Z-section about y-axis is %f cm \n',y_c)
```

---

# Chapter 5

## General Case of Forces in a plane

Scilab code Exa 5.2 Equations of equilibrium

```
1 //Inilization of variables
2 W=2000 //N
3 Lab=2 //m //length of the member from the vertical
   to the 1st load of 2000 N
4 Lac=5 //m //length of the member from the vertical
   to the 2nd load of 2000 N
5 Lpq=3.5 //m
6 //Calculations
7 Rq=((W*Lab)+(W*Lac))/Lpq //N //take moment abt. pt P
8 Xp=Rq //N //sum Fx=0
9 Yp=2*W //N //sum Fy=0
10 Rp=sqrt(Xp^2+Yp^2) //N
11 //Resuts
12 clc
13 printf('The reaction at P is %f N \n',Rp)
14 printf('The reaction at Q is %f N \n',Rq)
```

---

### Scilab code Exa 5.3 Equations of equilibrium

```
1 //Initialization of variables
2 W=25 //N // self weight of the ladder
3 M=75 //N // weight of the man standing o the ladder
4 theta=63.43 //degree // angle which the ladder makes
   with the horizontal
5 alpha=30 //degree // angle made by the string with
   the horizontal
6 Loa=2 //m // spacing between the wall and the ladder
7 Lob=4 //m //length from the horizontal to the top of
   the ladder touching the wall(vertical)
8 //Calculations
9 //Using matrix to solve the simultaneous eqn's 3 & 4
10 A=[2 -4;1 -0.577]
11 B=[100;100]
12 C=inv(A)*B
13 //Results
14 clc
15 printf('The reaction at A i.e Ra is %f N \n',C(1))
16 printf('The reaction at B i.e Rb is %f N \n',C(2))
17 //Calculations
18 T=C(2)/cosd(alpha) //N // from (eqn 1)
19 //Results
20 printf('The required tension in the string is %f N \
   n',T)
```

---

### Scilab code Exa 5.4 Equations of Equilibrium

```
1 //Initialization of variables
2 W=100 //N
3 theta=60 //degree //angle made by the ladder with
   the horizontal
4 alpha=30 //degree //angle made by the ladder with
   the vertical wall
```

```

5 Lob=4 //m // length from the horizontal to the top
   of the ladder touching the wall(vertical)
6 Lcd=2 //m // length from the horizontal to the
   centre of the ladder where the man stands
7 //Calculations
8 Lab=Lob*secd(alpha) //m //length of the ladder
9 Lad=Lcd*tand(alpha) //m
10 Rb=(W*Lad)/Lab //N //take moment at A
11 Xa=Rb*sind(theta) //N // From eq'n 1
12 Ya=W+Rb*cosd(theta) //N From eq'n 2
13 //Results
14 clc
15 printf('The reaction at B i.e Rb is %f N \n',Rb)
16 printf('The horizontal reaction at A i.e Xa is %f N
   \n',Xa)
17 printf('The vertical reaction at A i.e Ya is %f N \n
   ',Ya)

```

---

### Scilab code Exa 5.5 Equations of Equilibrium

```

1 //Initilization of variables
2 W=100 //N //self weight of the man
3 alpha=30 //degree // angle made by the ladder with
   the wall
4 Lob=4 //m // length from the horizontal to the top
   of the ladder touching the wall(vertical)
5 Lcd=2 //m
6 //Calculations
7 // using the equilbirium equations
8 Ya=W //N // From eq'n 2
9 Lad=Lcd*tand(alpha) //m //Lad is the distance fom pt
   A to the point where the line from the cg
   intersects the horizontal
10 Rb=(W*Lad)/Lob //N // Taking sum of moment abt A
11 Xa=Rb //N // From eq'n 1

```

```

12 //Results
13 clc
14 printf('The horizontal reaction at A i.e Xa is %f N
        \n',Xa)
15 printf('The vertical reaction at A i.e Ya is %f N \n
        ',Ya)
16 printf('The reaction at B i.e Rb is %f N \n',Rb)

```

---

### Scilab code Exa 5.6 Equations of Equilibrium

```

1 //Initilization of variables
2 d=0.09 //m //diametre of the right circular cylinder
3 h=0.12 //m //height of the cyinder
4 W=10 //N // self weight of the bar
5 l=0.24 //m //length of the bar
6 //Calculations
7 theta=atand(h/d) // angle which the bar makes with
        the horizontal
8 Lad=sqrt(d^2+h^2) //m // Lad is the length of the
        bar from point A to point B
9 Rd=(W*h*cosd(theta))/Lad //N // Taking moment at A
10 Xa=Rd*sind(theta) //N // sum Fx=0.... From eq'n 1
11 Ya=W-(Rd*cosd(theta)) //N // sum Fy=0..... From eq'n
        2
12 Ra=sqrt(Xa^2+Ya^2) //resultant of Xa & Ya
13 //Results
14 clc
15 printf('The horizontal reaction at A i.e Xa is %f N
        \n',Xa)
16 printf('The vertical reaction at A i.e Ya is %f N \n
        ',Ya)
17 printf('Therefore the reaction at A i.e Ra is %f N \
        n',Ra)
18 printf('The reaction at D i.e Rd is %f N \n',Rd)

```

---

# Chapter 6

## Friction

Scilab code Exa 6.1 Friction

```
1 // Initialization of variables
2 m=5 //kg // mass of the bock
3 g=9.81 // m/s^2 // acceleration due to gravity
4 theta=15 // degree // angle made by the forces (P1 &
   P2) with the horizontal of the block
5 mu=0.4 //coefficient of static friction
6 //Calculations
7 // Case 1. Where P1 is the force required to just
   pull the bock
8 // Solving eqn's 1 & 2 using matrix
9 A=[cosd(theta) -mu;sind(theta) 1]
10 B=[0;(m*g)]
11 C=inv(A)*B
12 // Calculations
13 // Case 2. Where P2 is the force required to push
   the block
14 // Solving eqn's 1 & 2 using matrix
15 P=[-cosd(theta) mu;-sind(theta) 1]
16 Q=[0;(m*g)]
17 R=inv(P)*Q
18 // Results
```

```
19 clc
20 printf('The required pull force P1 is %f N \n',C(1))
21 printf('The required push force P2 is %f N \n',R(1))
```

---

#### Scilab code Exa 6.4 Friction

```
1 // Initialization of variables
2 W1=50 // N // weight of the first block
3 W2=50 // N // weight of the second block
4 mu_1=0.3 // coefficient of friction between the
   inclined plane and W1
5 mu_2=0.2 // coefficient of friction between the
   inclined plane and W2
6 // Calculations
7 // On adding eq'ns 1&3 and substituting the values of
   N1 & N2 from eqn's 2&4 in this and on solving for
   alpha we get ,
8 alpha=atan(((mu_1*W1)+(mu_2*W2))/(W1+W2)) //
   degrees
9 // Results
10 clc
11 printf('The inclination of the plane is %f degree \n
   ',alpha)
```

---

#### Scilab code Exa 6.7 Friction

```
1 // Initialization of variables
2 M=2000 // kg // mass of the car
3 mu=0.3 // coefficient of static friction between the
   tyre and the road
4 g=9.81 // m/s^2 // acc. due to gravity
5 // Calculations
6 // Divide eqn 1 by eqn 2, We get
```



```

7 theta=atand(mu) //degree
8 // Results
9 clc
10 printf('The angle of inclination is %f degree \n',
        theta)

```

---

### Scilab code Exa 6.9 Friction

```

1 // Initalization of variabes
2 Wa=1000 //N // weight of block A
3 Wb=500 //N // weight of block B
4 theta=15 // degree // angle of the wedge
5 mu=0.2 // coefficient of friction between the
    surfaces in contact
6 phi=7.5 // degrees // used in case 2
7 pie=3.14
8 // Caculations
9 // CASE (a)
10 // consider the equilibrium of upper block A
11 // rearranging eq'ns 1 &2 and solving them using
    matrix for N1 & N2
12 A=[1 -0.4522;-0.2 0.914]
13 B=[0;1000]
14 C=inv(A)*B
15 // Now consider the equilibrium of lower block B
16 // From eq'n 4
17 N3=Wb+(C(2)*cosd(theta))-(mu*C(2)*sind(theta)) //N
18 // Now from eq'n 3
19 P=(mu*N3)+(mu*C(2)*cosd(theta))+(C(2)*sind(theta))
    // N
20 // CASE (b)
21 // The eq'n for required coefficient for the wedge
    to be self locking is ,
22 mu_req=(theta*pie)/(360) // multiplying with (pie
    /180) to convert it into radians

```

```

23 // Results
24 clc
25 printf('The minimum horizontal force (P) which
        should be applied to raise the block is %f N \n',
        P)
26 printf('The required coefficient for the wedge to be
        self locking is %f \n',mu_req)

```

---

### Scilab code Exa 6.13 Friction

```

1 // Initialization of variables
2 P=100 //N // force acting at 0.2 m from A
3 Q=200 //N // force acting at any distance x from B
4 l=1 //m // length of the bar
5 theta=45 //degree //angle made by the normal
        reaction at A&B with horizontal
6 // Calculations
7 // solving eqn's 1 & 2 using matrix for Ra & Rb,
8 A=[1 -1;sind(theta) sind(theta)]
9 B=[0;(P+Q)]
10 C=inv(A)*B
11 // Now take moment about B
12 x=((C(1)*l*sind(theta))-(P*(1-0.2)))/200 //m // here
        0.2 is the distance where 100 N load lies from A
13 // Results
14 clc
15 printf('The minimum value of x at which the load Q
        =200 N may be applied before slipping impends is
        %f m \n',x)

```

---

# Chapter 7

## Application of friction

Scilab code Exa 7.1 Application of Friction

```
1 // Initalization of variables
2 d1=24 // cm // diameter of larger pulley
3 d2=12 // cm // diameter of smaller pulley
4 d=30 //cm // seperation between 1st & the 2nd pulley
5 pie=3.14
6 // Calcuations
7 r1=d1/2 //cm // radius of 1st pulley
8 r2=d2/2 //cm // radius of 2nd pulley
9 theta=asind((r1-r2)/d) //degrees
10 // Angle of lap
11 beta_1=180+(2*theta) //degree // for larger pulley
12 beta_2=180-(2*theta) //degree //for smaller pulley
13 L=pie*(r1+r2)+(2*d)+((r1-r2)^2/d) //cm // Length of
    the belt
14 // Results
15 clc
16 printf('The angle of lap for the larger pulley is %f
    degree \n',beta_1)
17 printf('The angle of lap for the smaller pulley is
    %f degree \n',beta_2)
18 printf('The length of pulley required is %f cm \n',L
```

)

---

### Scilab code Exa 7.2 Application of Friction

```
1 // Initialization of variables
2 d1=0.6 //m // diameter of larger pulley
3 d2=0.3 //m // diameter of smaller pulley
4 d=3.5 //m // separation between the pulleys
5 pie=3.14
6 // Calculations
7 r1=d1/2 //m // radius of larger pulley
8 r2=d2/2 //m // radius of smaller pulley
9 theta=asind((r1+r2)/d) //degree
10 // Angle of lap for both the pulleys is same, i.e
11 beta=180+(2*theta) // degree
12 L=((pie*(r1+r2))+(2*d)+((r1-r2)^2/d)) //cm // Length
    of the belt
13 // Results
14 clc
15 printf('The angle of lap for the pulley is %f degree
    \n',beta)
16 printf('The length of pulley required is %f m \n',L)
```

---

### Scilab code Exa 7.4 Belt friction

```
1 // Initialization of variables
2 W1=1000 //N
3 mu=0.25 //coefficient of friction
4 pie=3.14
5 beta=pie
6 T1=W1 // Tension in the 1st belt carrying W1
7 e=2.718 //constant
8 // Calculations
```

```

 9 T2=T1/(e^(mu*beta)) //N // Tension in the 2nd belt
10 W2=T2 //N
11 // Results
12 clc
13 printf('The minimum weight W2 to keep W1 in
        equilibrium is %f N \n',W2)

```

---

### Scilab code Exa 7.5 Belt friction

```

1 // Initalization of variables
2 mu=0.5 // coefficient of friction between the belt
        and the wheel
3 W=100 //N
4 theta=45 //degree
5 e=2.718
6 Lac=0.75 //m // ength of the lever
7 Lab=0.25 //m
8 Lbc=0.50 //m
9 r=0.25 //m
10 pie=3.14 // constant
11 // Calculations
12 beta=((180+theta)*pie)/180 // radian // angle of lap
13 // from eq'n 2
14 T1=(W*Lbc)/Lab //N
15 T2=T1/(e^(mu*beta)) //N // from eq'n 1
16 // consider the F.B.D of the pulley and take moment
        about its center, we get Braking Moment (M)
17 M=r*(T1-T2) //N-m
18 // Results
19 clc
20 printf('The braking moment (M) exerted by the
        vertical weight W is %f N-m \n',M)

```

---

### Scilab code Exa 7.6 belt friction

```
1 // Initilization of variables
2 W= 1000 //N // or 1kN
3 mu=0.3 // coefficient of friction between the rope
    and the cylinder
4 e=2.718 // constant
5 pie=3.14 // constant
6 alpha=90 // degree // since 2*alpha=180 degree
7 // Calculations
8 beta=2*pie*3 // radian // for 3 turn of the rope
9 // Here T1 is the larger tension in that section of
    the rope which is about to slip
10 T1=W //N
11 F=W/e^(mu*(1/(sind(alpha)))*(beta)) //N // Here T2=F
12 // Results
13 clc
14 printf('The force required to suport the weight of
    1000 N i.e 1kN is %f N \n',F)
```

---

### Scilab code Exa 7.7 belt friction

```
1 // Initilization of variables
2 Pw=50 //kW
3 T_max=1500 //N
4 v=10 // m/s // velocity of rope
5 w=4 // N/m // weight of rope
6 mu=0.2 // coefficient of friction
7 g=9.81 // m/s^2 // acceleration due to gravity
8 e=2.718 // constant
9 pie=3.14 // constant
10 alpha=30 // degree // since 2*alpha=60
11 // Calculations
12 beta=pie // radian // angle of lap
13 T_e=(w*v^2)/g // N // where T_e is the centrifugal
```

```

    tension
14 T1=(T_max)-(T_e) //N
15 T2=T1/(e^(mu*(1/sind(alpha))*(beta))) //N // From eq
    'n T1/T2=e^(mu*cosec(alpha)*beta)
16 P=(T1-T2)*v*(10^-3) //kW // power transmitted by a
    single rope
17 N=Pw/P // Number of ropes required
18 // Results
19 clc
20 printf('The number of ropes required to transmit 50
    kW is %f Nos \n',N)
21 // approx no of ropes is 5

```

---

#### Scilab code Exa 7.8 belt friction

```

1 // Initialization of variables
2 d1=0.45 //m // diameter of larger pulley
3 d2=0.20 //m // diameter of smaller pulley
4 d=1.95 //m // separation between the pulley's
5 T_max=1000 //N // or 1kN which is the maximum
    permissible tension
6 mu=0.20 // coefficient of friction
7 N=100 // r.p.m // speed of larger pulley
8 e=2.718 // constant
9 pie=3.14 // constant
10 T_e=0 //N // as the data for calculating T_e is not
    given we assume T_e=0
11 // Calculations
12 r1=d1/2 //m // radius of larger pulley
13 r2=d2/2 //m // radius of smaller pulley
14 theta=asind((r1+r2)/d) // degree
15 // for cross drive the angle of lap for both the
    pulleys is same
16 beta=((180+(2*(theta)))*pie)/180 //radian
17 T1=T_max-T_e //N

```

```

18 T2=T1/(e^(mu*(beta))) //N // from formulae , T1/T2=e
    ^ (mu*beta)
19 v=((2*pie)*N*r1)/60 // m/s // where v=velocity of
    belt which is given as , v=wr=2*pie*N*r/60
20 P=(T1-T2)*v*(10^-3) //kW // Power
21 // Results
22 clc
23 printf('The power transmitted by the cross belt
    drive is %f kW \n',P)
24 // the approx answer is 1.3 kW The answer given in
    the book (i.e 1.81kW) is wrong.

```

---

#### Scilab code Exa 7.9 belt friction

```

1 // Initalization of variabes
2 b=0.1 //m //width of the belt
3 t=0.008 //m //thickness of the belt
4 v=26.67 // m/s // belt speed
5 pie=3.14 // constant
6 beta=165 // radian // angle of lap for the smaller
    belt
7 mu=0.3 // coefficient of friction
8 sigma_max=2 // MN/m^2 // maximum permissible stress
    in the belt
9 m=0.9 // kg/m // mass of the belt
10 g=9.81 // m/s^2
11 e=2.718 // constant
12 // Calculations
13 A=b*t // m^2 // cross-sectional area of the belt
14 T_e=m*v^2 // N // where T_e is the Centrifugal
    tension
15 T_max=(sigma_max)*(A)*(10^6) // N // maximum tension
    in the belt
16 T1=(T_max)-(T_e) // N
17 T2=T1/(e^((mu*pie*beta)/180)) //N // from formulae

```



```

    T1/T2=e^(mu*beta)
18 P=(T1-T2)*v*(10^-3) //kW // Power transmitted
19 T_o=(T1+T2)/2 // N // Initial tension
20 // Now calculations to transmit maximum power
21 Te=T_max/3 // N // max tension
22 u=sqrt(T_max/(3*m)) // m/s // belt speed for max
    power
23 T_1=T_max-Te // N // T1 for case 2
24 T_2=T_1/(e^((mu*pie*beta)/180)) // N
25 P_max=(T_1-T_2)*u*(10^-3) // kW // Max power
    transmitted
26 // Results
27 clc
28 printf('The initial power transmitted is %f kW \n',P
    )
29 printf('The initial tension in the belt is %f N \n',
    T_o)
30 printf('The maximum power that can be transmitted is
    %f kW \n',P_max)
31 printf('The maximum power is transmitted at a belt
    speed of %f m/s \n',u)

```

---

#### Scilab code Exa 7.10 Friction in a square threaded screw

```

1 // Initalization of variables
2 p=0.0125 // m // pitch of screw
3 d=0.1 //m // diameter of the screw
4 r=0.05 //m // radius of the screw
5 l=0.5 //m // length of the lever
6 W=50 //kN // load on the lever
7 mu=0.20 // coefficient of friction
8 pie=3.14 //constant
9 // Calculations
10 theta=atand(p/(2*pie*r)) //degree // theta is the
    Helix angle

```

```

11 phi=atand(mu) // degree // phi is the angle of
    friction
12 // Taking the leverage due to handle into account,
    force F1 required is,
13 F1=(W*(tand(theta+phi)))*(r/l) //kN
14 // To lower the load
15 F2=(W*(tand(theta-phi)))*(r/l) //kN // -ve sign of
    F2 indicates force required is in opposite sense
16 E=(tand(theta)/tand(theta+phi))*100 // % // here E=
    eata=efficiency in %
17 // Results
18 clc
19 printf('The force required (i.e F1) to raise the
    weight is %f kN \n',F1)
20 printf('The force required (i.e F2) to lower the
    weight is %f kN \n',F2)
21 printf('The efficiency of the jack is %f percent \n'
    ,E)

```

---

### Scilab code Exa 7.11 Application of Friction

```

1 // Initilization of variabes
2 P=20000 //N //Weight of the shaft
3 D=0.30 //m //diameter of the shaft
4 R=0.15 //m //radius of the shaft
5 mu=0.12 // coefficient of friction
6 // Calculations
7 // Friction torque T is given by formulae ,
8 T=(2/3)*P*R*mu //N-m
9 M=T //N-m
10 // Results
11 clc
12 printf('The frictional torque is %f N-m \n',M)

```

---

# Chapter 8

## Simple Lifting Machines

Scilab code Exa 8.1 Simple Machine

```
1 // Initalization of variables
2 V.R=6 // Velocity ratio
3 P=20 //N // Effort
4 W=100 //N // Load lifted
5 // Calculations
6 //(a)
7 P_actual=P //N
8 W_actual=W //N
9 M.A=W/P // where , M.A= Mechanical advantage
10 E=(M.A/V.R)*100 //% // Where E= efficiency
11 //(b)
12 // Now ideal effort required is ,
13 P_ideal=W/V.R //N
14 // Effort loss in friction is , (Le)
15 Le=P_actual-P_ideal //N // Effort loss in friction
16 //(c)
17 // Ideal load lifted is ,(W_ideal)
18 W_ideal=P*V.R //N
19 // Frictional load/resistance ,
20 F=W_ideal-W_actual // N
21 // Results
```

```

22 clc
23 printf('(a) The efficiency of the machine is %f
    percent \n',E)
24 printf('(b) The effort loss in friction of the
    machine is %f N \n',Le)
25 printf('(c) The Frictional load of the machine is %f
    N \n',F)

```

---

### Scilab code Exa 8.2 Simple machines performance

```

1 // Initilization of variables
2 V_r=20 // Velocity ratio
3 // Values from the table // Variables have been
    assumed
4 // Values of W in N
5 W=[30;40;50;60;70;80;90;100]
6 // P in N
7 P=[7;8.5;10;11.5;13.5;14.5;16;17.5]
8 M.A=[W(1)/P(1);W(2)/P(2);W(3)/P(3);W(4)/P(4);W(5)/P
    (5);W(6)/P(6);W(7)/P(7);W(8)/P(8)]
9 // Efficiency (n)
10 n=(V_r^-1)*[M.A(1);M.A(2);M.A(3);M.A(4);M.A(5);M.A
    (6);M.A(7);M.A(8)]*100 // %
11 // Calculations
12 // Part (a)– Realtionship between W & P
13 // Here part a cannot be solved as it has variables
    which cannot be defined in Scilab. Ref.textbook
    for the solution
14 // Part (b)– Graph between W & efficiency n(eta)
15 x=[0;W(1);W(2);W(3);W(4);W(5);W(6);W(7);W(8)] //
    values for W // N
16 y=[0;n(1);n(2);n(3);n(4);n(5);n(6);n(7);n(8)] //
    values for efficiency n (eta) // %
17 subplot(221)
18 xlabel("W (N)")

```

```

19 ylabel(" efficiency n (%)")
20 plot(x,y)
21 // Results
22 clc
23 printf('The graph is the solution')
24 // The value of m is found by drawing straight line
    on the graph and by taking its slope. Ref
    textbook for the solution
25 // The curve of the graph may differ from textbook
    because of the graphical calculation.

```

---

### Scilab code Exa 8.3 Performance of Machine

```

1 // Calculations
2 W_actual=1360 //N //Load lifted
3 P_actual=100 //N // Effort
4 n=4 // no of pulleys
5 // Calculations
6 // for 1st system of pulleys having 4 movable
    pulleys, Velocity ratio is
7 V.R=2^(n) // Velocity Ratio
8 // If the machine were to be ideal(frictionless)
9 M.A=V.R // Here, M.A= mechanical advantage
10 // For a load of 1360 N, ideal effort required is
11 P_ideal=W_actual/V.R //N
12 // Effort loss in friction is,
13 P_friction=P_actual-P_ideal //N
14 // For a effort of 100 N, ideal load lifted is,
15 W_ideal=V.R*100 //N
16 // Load lost in friction is,
17 W_friction=W_ideal-W_actual // N
18 // Results
19 clc
20 printf('(a) The effort wasted in friction is %f N \n
    ',P_friction)

```

```
21 printf('(b) The load wasted in friction is %f N \n',  
    W_friction)
```

---

#### Scilab code Exa 8.4 Performance of a machine

```
1 // Initalization of variables  
2 W=1000 //N // Load to be lifted  
3 n=5 // no. of pulleys  
4 E=75 //% // Efficiency  
5 // Calculations  
6 // Velocity Ratio is given as,  
7 V.R=n  
8 // Mechanical Advantage (M.A) is ,  
9 M.A=(E/100)*V.R // from formulae, Efficiency=E=M.A/V  
    .R  
10 P=W/M.A //N // Effort required  
11 // Results  
12 clc  
13 printf('The effort required to lift the load of 1000  
    N is %f N \n',P)
```

---

#### Scilab code Exa 8.5 Simple screw jack

```
1 // Initalization of variables  
2 W=2000 //N // Load to be raised  
3 l=0.70 //m // length of the handle  
4 d=0.05 //m // diameter of the screw  
5 p=0.01 //m // pitch of the screw  
6 mu=0.15 // coefficient of friction at the screw  
    thread  
7 pie=3.14 //constant  
8 E=1 // efficiency  
9 // Calculations
```

```

10 phi=atand(mu) //degree
11 theta=atand(p/(pie*d)) //degree // where theta is
    the Helix angle
12 // Force required at the circumference of the screw
    is ,
13 P=W*tand(theta+phi) // N //
14 // Force required at the end of the handle is ,
15 F=(P*(d/2))/l //N
16 // Force required (Ideal case)
17 V.R=2*pie*l/p
18 M.A=E*V.R // from formulae E=M.A/V,R
19 P_ideal=W/M.A //N // From formulae , M.A=W/P
20 // Results
21 clc
22 printf('The force required at the end of the handle
    is %f N \n',F)
23 printf('The force required if the screw jack is
    considered to be an ideal machine is %f N \n',
    P_ideal)

```

---

## Chapter 9

# Analysis of Plane Trusses and Frames

Scilab code Exa 9.1 Axial forces in members of Truss

```
1 // Initialization of variables
2 W1=2000 //N // load at joint D of the truss
3 W2=4000 //N // load at joint E of the truss
4 Lac=6 //m // length of the tie
5 Lab=3 //m
6 Lbc=3 //m
7 theta=60 //degree // interior angles of the truss
8 // Calculations
9 // Here A is simply supported & B is roller support
  . Now the SUPPORT REACTIONS are given as,
10 Rc=((W1*(Lab/2))+(W2*(Lab+(Lbc/2))))/Lac //N //
    Taking moment at A
11 Ra=W1+W2-Rc //N // Take sum Fy=0
12 // ANALYSIS OF TRUSS BY METHOD OF JOINT
13 // ASSUMPTION- we consider the,(1) Forces moving
    towards each other as +ve i.e TENSILE (T) & (2)
    Forces moving away from each other as -ve i.e
    COMPRESSIVE (C)
14 // (1) JOINT A
```



```

15 Fad=Ra/(sind(theta)) //N //(C) // Using eq'n 2
16 Fab=Fad*cosd(theta) //N // (T) // Using eq'n 1
17 // (2) JOINT C
18 Fce=Rc/(sind(theta)) //N // (C) // Using eq'n 4
19 Fcb=Fce*cosd(theta) //N // (T) // Using eq'n 3
20 // (3) JOINT D
21 Fdb=((Fad*sind(theta))-(W1))/sind(theta) //N // (T)
    // Using eq'n 6
22 Fde=(Fdb*cosd(theta))+(Fad*cosd(theta)) //N // (C)
    // Using eq'n 5
23 // (4) JOINT E
24 Feb=((Fce*cosd(theta))-(Fde))/cosd(theta) //N // (C)
    // Using eq'n 7
25 // Results
26 clc
27 printf('The Axial Force in member AD (Fad) is %f N \
    n',Fad)
28 printf('The Axial Force in member AB (Fab) is %f N \
    n',Fab)
29 printf('The Axial Force in member CE (Fce) is %f N \
    n',Fce)
30 printf('The Axial Force in member CB (Fcb) is %f N \
    n',Fcb)
31 printf('The Axial Force in member DB (Fdb) is %f N \
    n',Fdb)
32 printf('The Axial Force in member DE (Fde) is %f N \
    n',Fde)
33 printf('The Axial Force in member EB (Feb) is %f N \
    n',Feb)

```

---

### Scilab code Exa 9.2 axial forces in members of truss

```

1 // Initialization of variables
2 W1=2000 //N (or 2 kN)// load at joint D of the truss
3 W2=4000 //N (or 4 kN)// load at joint E of the truss

```

```

4 Lac=6 //m // length of the tie
5 Lab=3 //m
6 Lbc=3 //m
7 theta=60 //degree // interior angles of the truss
8 // Calculations
9 // Here A is simply supported & B is roller support
. Now the SUPPORT REACTIONS are given as,
10 Rc=((W1*(Lab/2))+(W2*(Lab+(Lbc/2))))/Lac //N //
    Taking moment at A
11 Ra=W1+W2-Rc //N // Take sum Fy=0
12 // Calculations
13 // Calculating the axial forces in the respective
    members by METHOD OF SECTION
14 // A section is drawn passing through member DE such
    that it cuts the respective member. Now consider
    the equilibrium of the left hand portion of the
    truss. The three unknown forces are Fde, Fdb, &
    Fab
15 // Take moment about B
16 Fde=((3*Ra)-(W1*Lab*sind(30)))/(3*cosd(30)) //N // (
    T)
17 // Results
18 clc
19 printf('The axial force in the member DE (Fde)is %f
    N \n',Fde)

```

---

### Scilab code Exa 9.3 Axial Forces in members of the truss

```

1 // Initalization of variables
2 W=1 //kN // load on the truss at joint D
3 theta=45 //degree // angle made by the members AC &
    BD with the horizontal
4 Lab=1 //m
5 Lcd=1 //m // here Lcd= the distance from B to the
    line of extension drawn from 1kN force on the

```

```

        horizontal
6 // Calculations
7 // (1) JOINT E
8 // Here the joint E is in equilibrium under two
    forces Fec & Fed which are non-collinear. Hence
    they must be 0. i.e Fec=Fed=0
9 Fec=0
10 Fed=0
11 // (2) JOINT D
12 Fdb=W/sind(theta) // kN // (C)// sum Fy=0
13 Fdc=Fdb*cosd(theta) // kN // (T) // sum Fx=0
14 // (3) JOINT C
15 Fca=Fdc/sind(theta) // kN // (T) // sum Fx=0
16 Fcb=-(Fca*sind(theta)) // kN // (C) // sum Fy=0
17 // Results
18 clc
19 printf('The axial force in the member DC (Fdc) is %f
    kN \n',Fdc)
20 printf('The axial force in the member DB (Fdb) is %f
    kN \n',Fdb)
21 printf('The axial force in the member CA (Fca) is %f
    kN \n',Fca)
22 printf('The axial force in the member CB (Fcb) is %f
    kN \n',Fcb)
23 printf('The axial force in the member EC (Fec) is %f
    kN \n',Fec)
24 printf('The axial force in the member ED (Fed) is %f
    kN \n',Fed)
25 // Here -ve sign indicates COMPRESSIVE force & +ve
    indicates TENSILE force

```

---

#### Scilab code Exa 9.5 Axial Forces in members of the truss

```

1 // Initialization of variables
2 W1=1000 //N // Load acting at the end pannels and

```

```

    the ridge
3  W2=2000 //N // Load acting at the intermediate
    pannels
4  Laf=1 //m
5  Lgf=1 //m
6  Lag=2 //m
7  Lbg=1 //m
8  Lab=3 //m
9  theta=30 //degree // angle made by the principal
    rafter with the tie beam
10 beta=60 //degree // angle made by the slings (i.e
    members CF & CG) with the tie beam
11 // Calculations
12 // consider the equilibrium of the entire truss as a
    F.B.D
13 Xa=2*(W1*sind(theta))+(W2*sind(theta)) //N // sum Fx
    =0
14 Rb=((W2*Laf*cosd(theta))+(W1*Lag*cosd(theta)))/Lab
    // N // Moment at A=0
15 Ya=2*(W1*cosd(theta))+(W2*cosd(theta))-(Rb) //N //
    sum Fy=0
16 // Now pass a section through the truss such that it
    cuts the members CE,CG & FG. Now consider the
    equilibrium of the right hand side of the truss
17 // Take moment about C
18 Ffg=(Rb*(Lbg+0.5))/(0.5*tand(beta)) // N // (T) //
    Here 0.5 is the half distance of Lgf
19 // Take moment about G
20 Fce=(-Rb*Lbg)/(Lbg*sind(theta)) // N // (C)
21 // Take moment about B
22 Fcg=0/(Lbg*sind(beta)) // N
23 // Results
24 clc
25 printf('The axial force in the member FG (Ffg) is %f
    N \n',Ffg)
26 printf('The axial force in the member CE (Fce) is %f
    N \n',Fce)
27 printf('The axial force in the member CG (Fcg) is %f

```

N \n', Fcg)

---

**Scilab code Exa 9.6** Axial Forces in members of the truss

```
1 // Initalization of variables
2 W1=100 //N // load acting at pt. C vertically
3 W2=50 //N // load acting at point B horizontaly
4 L=2 //m // length of each bar in the hexagonal truss
5 theta=60 //degree // internal angle of the truss
6 // Calculations
7 // We calculate the values of different members of
  the truss
8 HG=L*sind(theta)
9 AF=L
10 // Support A is hinged whereas support F is a roller
  support. Firstly we find the support reactios as
  follows ,
11 Rf=(W2*HG)/AF //N // moment at F
12 Xa=W2 //N // sum Fx=0
13 Ya=W1-Rf //N // sum Fy=0
14 // Now pass a section through the truss cutting the
  members CD,GD,GE & GF and consider equilibrium of
  right hand portion of the truss
15 Fcd=(Rf*(L/2))/(L*sind(theta)) // N (C) // Taking
  moment about G
16 // Now pass a scetion pq cutting the members CB,GB &
  GA
17 Fga=((Rf*(L+(L/2)))-(W1*(L/2)))/(L*sind(theta)) // N
  (T) // Taking moment about B
18 // take moment about G
19 Fcb=((W1*(L/2))+(Rf*(L/2)))/(L*sind(theta)) // N (C)
20 Fgb=(Fcb*cosd(theta))-(Fga*cosd(theta)) // N (T) //
  sum Fx=0
21 // Results
22 clc
```

```

23 printf('The axial force in the member CD (Fcd) is %f
      N \n',Fcd)
24 printf('The axial force in the member GB (Fgb) is %f
      N \n',Fgb)

```

---

### Scilab code Exa 9.10 Axial Forces in members of the truss

```

1 // Initilization of variables
2 W=24 // kN // Load acting at pt C
3 Laf=12 // m // length of the tie beam
4 l=4 // m// length of each member in the tie
5 h=3 // m // height of the slings
6 Lae=8 // m
7 // Calculations
8 s=sqrt((l^2)+(h^2)) // m // sloping length
9 // From triangle BCD,
10 theta=acosd(h/s)
11 // SUPPORT REACTIONS
12 Rf=(W*l)/Laf // kN // take moment at A
13 Ra=W-Rf // kN // sum Fy=0
14 // now pass a section through the truss and
      consider te equilibrium of the left hand portion
15 Fce=(Ra*l)/h // kN (T) // Take moment at B
16 Fbd=((W*l)-(Ra*Lae))/h // kN (C) // take moment at E
17 Fbe=(Ra-W)/cosd(theta) // kN
18 Fbd=(-Ra*l)/h // kN // take moment at C
19 Fce=((Ra*Lae)-(W*l))/h // kN (T) // take moment at D
20 Fcd=(W-Ra)/cosd(theta) // kN (T) // sum Fy=0
21 // Results
22 clc
23 printf('(1) The axial force in the bar CE (Fce) is
      %f kN \n',Fce)
24 printf('(2) The axial force in the bar BD (Fbd) is
      %f kN \n',Fbd)
25 printf('(3) The axial force in the bar BE (Fbe) is

```

```

    %f kN \n',Fbe)
26 printf('(4) The axial force in the bar CD (Fcd) is
    %f kN \n',Fcd)

```

---

**Scilab code Exa 9.12 Axial Forces in members of the truss**

```

1 // Initialization of variables
2 W1=4 // kN // load acting at a distance of 5 m from
   C
3 W2=3 // kN // load acting at a distance of 7.5 m
   from C
4 L=30 //m // distance AB
5 L1=15 // dist AC
6 L2=15 //m //dist BC
7 l1=10 //m // distance between A and 4 kN load
8 l2=22.5 //m // distance between A and 3 kN load
9 // Calculations
10 // (1) Reactions
11 Yb=((W1*l1)+(W2*l2))/L // kN // Take moment at A
12 Ya=W1+W2-Yb // kN // sum Fy=0
13 // Xa=Xb.....(eq'n 1) // sum Fx=0
14 // (2) Dismember
15 // Member AC. Consider equilibrium of member AC
16 // Xa=Xc ... Consider thus as eq'n 2 // sum Fx=0
17 Yc=W1-Ya // kN // sum Fy=0
18 // Take moment about A
19 Xc=((W1*l1)-(Yc*L1))/L1 // kN
20 // now from eq'n 1 & 2
21 Xa=Xc // kN
22 Xb=Xa // kN
23 // The components of reactions at A & B are ,
24 Ra=sqrt(Xa^2+Ya^2) // kN
25 Rb=sqrt(Xb^2+Yb^2) // kN
26 // Results
27 clc

```

```

28 printf('The reaction at A ( Ra) is %f kN \n',Ra)
29 printf('The reaction at B ( Rb) is %f kN \n',Rb)

```

---

**Scilab code Exa 9.13 Axial Forces in members of the truss**

```

1 // Initialization of variables
2 W1=2 // kN // load acting at a distance of 1m from
   point A
3 W2=1 // kN // load acting at a distance of 1m from
   point B
4 theta=30 // degree
5 L=4 // m // length of the tie beam
6 l=1 //m // length of each member in the tie
7 // Calculations
8 // (a) Reactions
9 Yb=((W1*1)+(W2*3*1))/L // kN // Taking moment about
   A
10 Ya=W1+W2-Yb // kN // sum Fy=0
11 // (b) Dismember
12 // MEMBER AB
13 // Xa=Xb..... (eq'n 1) // sum Fx=0
14 // MEMBER AC
15 // Xa=Xc.....(eq'n 2) // sum Fx=0
16 Yc=W1-Ya // kN // sum Fy=0
17 // Taking moment about A
18 Xc=((W1*1)-(Yc*2*1))/(2*tand(theta)) // kN
19 // From eq'n 1 & 2
20 Xa=Xc // kN
21 Xb=Xa // kN
22 // Results
23 clc
24 printf('The force in tie bar AB is %f kN \n',Xb)

```

---



Scilab code Exa 9.14 Axial Forces in members of the truss

```

1 // Initialization of variables
2 W=1000 // N
3 r=0.25 // radius of pulley at E
4 Lab=2 //m
5 Lad=1 // m
6 Lbd=1 // m
7 Ldc=0.75 // m
8 l1=0.5 //m // c/c distance between bar AB and point
   E
9 l2=1.25 // m // dist between rigid support and the
   weight
10 // Calculations
11 // (a) Reactions
12 Xa=W // N // sum Fx=0
13 Yb=((W*l1)+(W*l2))/Lab // N // Take moment about A
14 Ya=W-Yb // N // sum Fy=0
15 // Dismember
16 // MEMBER ADB
17 // consider triangle BCD to find theta, where s=
   length of bar BC,
18 s=sqrt(Lbd^2+Ldc^2) // m
19 theta=acosd(Lbd/s) // degree
20 // equilibrium eq'n of member ADB
21 Yd=(Ya*Lab)/Lad // take moment about B
22 Fbc=(Yb+Ya-Yd)/sind(theta) // N // sum Fy=0
23 Xd=(Fbc*cosd(theta))+Xa // N // sum Fx=0
24 // PIN D
25 Rd=sqrt(Xd^2+Yd^2) // N // shear force on the pin
26 // Results
27 clc
28 printf('The compressive force in bar BC (Fbc) is %f
   N \n',Fbc)
29 printf('The shear force on the pin is %f N \n',Rd)

```

---

Scilab code Exa 9.15 Axial Forces in members of the truss

```
1 // Initilization of variables
2 P=5000 // N
3 theta=45 // degree // angle made by Rd & Re with the
   horizontal
4 Lab=3 // m
5 Lac=3 // m
6 Lbd=2 // m
7 Lce=2 // m
8 l=1.5 // m // dist of load P from B
9 // Calculations (BEAM AB )
10 // Consider the equilibrium of beams
11 // We are using matrix to solve the simultaneous eqn
   's
12 A=[(Lbd*sind(theta)) Lab;(Lce*sind(theta)) -Lac]
13 B=[(P*l) 0]
14 C=B*inv(A)
15 // Calculations (BEAM AC)
16 Re=C(1) // N (C) // from eq'n 1
17 Ya=(Re*Lce*sind(theta))/Lac // N // from eq'n 7
18 Xa=C(1)*cosd(theta) // N // from eq'n 2
19 Ra=sqrt(Xa^2+Ya^2) // N (C)
20 Yb=P-Ya-(C(1)*sind(theta)) // N (C) // eq'n 3
21 Yc=Ya-(Re*sind(theta)) // N (T)
22 // Results
23 clc
24 printf('(1) The value of axial force (Rd) in bar 2
   is %f N \n',C(1))
25 printf('(2) The value of axial force (Re) in bar 3
   is %f N \n',Re)
26 printf('(3) The value of axial force (Yb) in bar 1
   is %f N \n',Yb)
27 printf('(4) The value of axial force (Yc) in bar 4
```

```
is %f N \n', Yc)
28 // here consider Yc as +ve coz the assumed direction
    of the force was compressive which is to be
    reversed
```

---

# Chapter 10

## Uniform Flexible Suspension Cables

Scilab code Exa 10.1 Cable subjected to concentrated loads

```
1 // Initialization of variables
2 W1=400 // N // vertical load at pt C
3 W2=600 // N // vertical load at pt D
4 W3=400 // N // vertical load at pt E
5 l=2 // m // l= Lac=Lcd=Lde=Leb
6 h=2.25 // m // distance of the cable from top
7 L=2 // m // dist of A from top
8 // Calculations
9 // Solving eqn's 1&2 using MATRIX for Xb & Yb
10 A=[-L 4*1;-h 2*1]
11 B=[((W1*1)+(W2*2*1)+(W1*3*1));(W1*1)]
12 C=inv(A)*B
13 // Now consider the F.B.D of BE, Take moment at E
14 y_e=(C(2)*1)/C(1) // m / here y_e is the distance
    between E and the top
15 theta_1=atand(y_e/l) // degree // where theta_1 is
    the angle between BE and the horizontal
16 T_BE=C(1)/cosd(theta_1) // N (T_BE=T_max)
17 // Now consider the F.B.D of portion BEDC
```

```

18 // Take moment at C
19 y_c=((C(2)*6)-(W3*4)-(W2*2))/(C(1)) // m
20 theta_4=atand(((y_c)-(1))/(1)) // degree
21 T_CA=C(1)/cosd(theta_4) // N // Tension in CA
22 // Results
23 clc
24 printf('(i) The horizontal reaction at B (Xb) is %f
      N \n',C(1))
25 printf('(i) The vertical reaction at B (Yb) is %f N
      \n',C(2))
26 printf('(ii) The sag at point E (y_e) is %f m \n',
      y_e)
27 printf('(iii) The tension in portion CA (T_CA) is %f
      N \n',T_CA)
28 printf('(iv) The max tension in the cable (T_max) is
      %f N \n',T_BE)
29 printf('(iv) The max slope (theta_1) in the cable is
      %f degree \n',theta_1)

```

---

### Scilab code Exa 10.2 Cables subjected to concentrated loads

```

1 // Initiiization of variables
2 W1=100 // N // Pt load at C
3 W2=150 // N // Pt load at D
4 W3=200 // N // Pt load at E
5 l=1 // m // l=Lac=Lcd=Lde=Leb
6 h=2 // m // dist between Rb & top
7 Xa=200 // N
8 Xb=200 // N
9 // Calculations
10 // consider the F.B.D of entire cable
11 // Take moment at A
12 Yb=((W1*1)+(W2*2*1)+(W3*3*1)-(Xb*h))/(4*1) // N
13 Ya=W1+W2+W3-Yb // N // sum Fy=0
14 // Now consider the F.B.D of AC

```

```

15 // Take moment at C,
16 y_c=(Ya*1)/Xa // m
17 theta_1=atand(y_c/1) // degree
18 T_AC=Xa/cosd(theta_1) // N // T_AC*cosd(theta_1)=
    horizontal component of tension in the cable
19 // here, T_AC=T_max
20 T_max=T_AC // N
21 // Now consider the F.B.D of portion ACD
22 y_d=((Ya*2*1)-(W1*1))/(Xa) // m // taking moment at
    D
23 theta_2=atand(((y_d)-(y_c))/(1)) // degree
24 T_CD=Xa/(cosd(theta_2)) // N
25 // Results
26 clc
27 printf('(i) The component of support reaction at A (
    Ya) is %f N \n',Ya)
28 printf('(i) The component of support reaction at B (
    Yb) is %f N \n',Yb)
29 printf('(ii) The tension in portion AC (T_AC) of the
    cable is %f N \n',T_AC)
30 printf('(ii) The tension in portion CD (T_CD) of the
    cable is %f N \n',T_CD)
31 printf('(iii) The max tension in the cable is %f N \
    n',T_max)

```

---

**Scilab code Exa 10.3** Cables uniformly loaded per unit horizontal distance

```

1 // Initialization of variables
2 w=75 // kg/m // mass per unit length of thw pipe
3 l=20 // m // dist between A & B
4 g=9.81 // m/s^2 // acc due to gravity
5 y=2 // m // position of C below B
6 // Calculations
7 // Let x_b be the distance of point C from B
8 // In eq'n  $x_b^2+32*x_b-320=0$ 

```

```

9 a=1
10 b=32
11 c=-320
12 x_b=(-b+sqrt(b^2-(4*a*c)))/(2*a) // m // we get x_b
    by equating eqn's 1&2
13 // Now tension T_0
14 T_0=((w*g*x_b^2)/(2*y))*(10^-3) //kN // from eq'n 1
15 // Now the max tension occurs at point A,hence x is
    given as,
16 x=20-x_b // m
17 w_x=w*g*x*10^(-3) // kN
18 T_max=sqrt((T_0)^2+(w_x)^2) // kN // Maximum Tension
19 // Results
20 clc
21 printf('The lowest point C which is situated at a
    distance (x_b) from support B is %f m \n',x_b)
22 printf('The maximum tension (T_max) in the cable is
    %f kN \n',T_max)
23 printf('The minimum tension (T_0) in the cable is %f
    kN \n',T_0)

```

---

#### Scilab code Exa 10.4 Cables uniformly loaded per unit horizontal distance

```

1 // Initalization of variables
2 m=0.5 // kg/m // mass of the cable per unit length
3 g=9.81 // m/s^2
4 x=30 // m // length AB
5 y=0.5 // m // dist between C & the horizontal
6 x_b=15 // m // dist of horizontal from C to B
7 // Calculations
8 w=m*g // N/m // weight of the cable per unit length
9 T_0=(w*x_b^2)/(2*y) // N // From eq'n 1
10 T_B=sqrt((T_0)^2+(w*x/2)^2) // N // Tension in the
    cable at point B
11 W=T_B // N // As pulley is frictionless the tension

```

```

    in the pulley on each side is same, so  $W=T_B$ 
12 // Slope of the cable at B,
13 theta=acosd(T_0/T_B) // degree
14 // Now length of the cable between C & B is ,
15 S_cb=x_b(1+((2/3)*(y/x_b)^2)) // m
16 // Now total length of the cable AB is ,
17 S_ab=2*S_cb // m
18 // Results
19 clc
20 printf('(i) The magnitude of load W is %f N \n',W)
21 printf('(ii) The angle of the cable with the
    horizontal at B is %f degree \n',theta)
22 printf('(iii) The total length of the cable AB is %f
    m \n',S_ab)

```

---

#### Scilab code Exa 10.5 Cables uniformly loaded per unit horizontal distance

```

1 // Initialization of variables
2 x=30 // m // distance between two electric poles
3 Tmax=400 // N // Max Pull or tension
4 w=3 // N/m // weight per unit length of the cable
5 // Calculations
6 // The cable is assumed to be parabolic in shape ,
    its eq'n is  $y=wx^2/2*T_0$  .....(eq'n 1).
    Substituting the co-ordinates of point B (l/2,h),
    where h is the sag in the cable.This gives ,  $T_0=($ 
     $w*(l/2)^2)/(2*h)=wl^2/8*h$ 
7 // Now the maximum pull or tension occurs at B,
8 T_B=Tmax // N
9 // Hence  $T_B=Tmax=\sqrt{T_0^2+(w*l/2)^2}$ . On
    simplyfyingthis eq'n we get ,
10 h=sqrt(x^2/(16*(((Tmax*2)/(w*x))^2-(1)))) // m
11 // Results
12 clc
13 printf('The smallest value of the sag in the cable

```



is %f m \n',h)

---

### Scilab code Exa 10.6 Catenary Cables

```
1 // Initalization of variables
2 l=200 // m // length of the cable
3 m=1000 // kg // mass of the cable
4 S=50 // m // sag in the cable
5 s=l/2 // m
6 g=9.81 // m/s^2
7 // Calculations
8 w=(m*g)/l // N/m // mass per unit length of the
   cable
9 // Substuting the values s=l/2 & y=c+S in eq'n 1 to
   get the value of c,
10 c=7500/100 // m
11 Tmax=sqrt((w*c)^2+(w*s)^2) // N // Maximum Tension
12 // To determine the span (2*x) let us use the eq'n
   of catenary,  $y=c*\cosh(x/c)$ , where  $y=c+50$ . On
   simplifying we get  $y/c=\cosh(x/c)$ , here let  $y/c=A$ 
13 y=c+50
14 A=y/c
15 x=c*(acosh(A)) // m
16 L=2*x // m // where L= span
17 // Results
18 clc
19 printf('The horizontal distance between the supports
   and the max Tension (L) is %f m \n',L)
```

---

# Chapter 12

## Moment of Inertia

Scilab code Exa 12.7 Moment of Inertia of an area of a plane figure with respect to

```
1 // Initialization of variables
2 A= 50 // cm^2 // area of the shaded portion
3 J_A=22.5*10^2 // cm^4 // polar moment of inertia of
   the shaded portion
4 d=6 // cm
5 // Calculations
6 J_c=J_A-(A*d^2)
7 // substituting the value of I_x from eq'n 2 in eq'n 1
   we get ,
8 I_y=J_c/3 // cm^4 // M.O.I about Y-axis
9 // Now from eq'n 2,
10 I_x=2*I_y // cm^4 // M.O.I about X-axis
11 // Results
12 clc
13 printf('The centroidal moment of inertia about X-
   axis (I_x) is %f cm^4 \n',I_x)
14 printf('The centroidal moment of inertia about Y-
   axis (I_y) is %f cm^4 \n',I_y)
```

---

Scilab code Exa 12.8 Moment of Inertia of a Composite area or hollow section

```
1 // Initialization of variables
2 b=20 // cm // width of the pate
3 d=30 // cm // depth of the plate
4 r=15 // cm // radius of the circular hole
5 h=20 // cm // distance between the centre of the
   circle & the x-axis
6 // Calculations
7 // (a) Location of the centroid of the composite
   area
8 A_1=b*d // cm^2 // area of the plate
9 y_1=d/2 // cm // y-coordinate of the centroid
10 A_2=(%pi*r^2)/4 // cm^2 // area of the circle
   removed (negative)
11 y_2=h // cm // y-coordinate of the centroid
12 y_c=((A_1*y_1)-(A_2*y_2))/(A_1-A_2) // cm // from
   the bottom edge
13 // (b) Moment of Inertia of the composite area about
   the centroidal x-axis
14 // Area (A_1) M.I of area A_1 about x-axis
15 I_x1=(b*(d^3))/12 // cm^4
16 // M.I of the area A_1 about the centroidal x-axis
   of the composite area (By parallel-axis theorem)
17 OC_1=15 // cm // from the bottom edge
18 OC_2=20 // cm
19 OC=12.9 // cm // from the bottom edge
20 d_1=OC_1-OC // cm
21 d_2=OC_2-OC // cm
22 I_X1=(I_x1)+(A_1*d_1^2) // cm^4
23 // Area(A_2) M.I of area A_2 about x-axis
24 I_x2=(%pi*r^4)/64 // cm^2
25 // M.I of the area A_2 about the centroidal x-axis
   of the composite area (By parallel-axis theorem)
26 I_X2=(I_x2)+(A_2*d_2^2) // cm^4
27 // COMPOSITE AREA:M.O.I of the composite area about
   the centroidal x-axis
28 I_x=(I_X1)-(I_X2) // cm^4
```

```

29 // Results
30 clc
31 printf('The M.O.I of the composite area about the
        centroidal x-axis is %f cm^4 \n',I_x)
32 // There may be a small error in the answer due to
        decimal point discrepancy

```

---

**Scilab code Exa 12.9** Moment of Inertia of a Composite area or hollow section

```

1 // Initalization of variables
2 b1=80 // mm // width of the flange pate
3 d1=20 // mm // depth of the flange plate
4 b2=40 // mm // width/thickness of the web
5 d2=60 // mm // depth of the web
6 // Calculations
7 // (a) Location of the centroid of the composite
        area
8 A_1=b1*d1 // mm^2 // area of the flange plate
9 y_1=d2+(d1/2) // mm // y-coordinate of the centroid
10 A_2=b2*d2 // mm^2 // area of the web
11 y_2=d2/2 // mm // y-coordinate of the centroid
12 y_c=((A_1*y_1)+(A_2*y_2))/(A_1+A_2) // mm // from
        the bottom edge
13 // (b) Moment of Inertia of the composite area about
        the centroidal x-axis
14 // Area (A_1) M.I of area A_1 about x-axis
15 I_x1=(b1*(d1^3))/12 // mm^4
16 // M.I of the area A_1 about the centroidal x-axis
        of the composite area (By parallel-axis theorem)
17 OC_1=70 // mm // from the bottom edge
18 OC_2=30 // mm // from the bottom edge
19 OC=y_c // mm // from the bottom edge
20 d_1=(d2-y_c)+(d1/2) // mm
21 d_2=y_c-OC_2 // mm
22 I_X1=(I_x1)+(A_1*d_1^2) // mm^4

```

```

23 // Area(A_2) M.I of area A_2 about x-axis
24 I_x2=(b2*d2^3)/12 // mm^4
25 // M.I of the area A_2 about the centroidal x-axis
    of the composite area (By parallel-axis theorem)
26 I_X2=(I_x2)+(A_2*d_2^2) // mm^4
27 // COMPOSITE AREA:M.O.I of the composite area about
    the centroidal x-axis
28 I_x=(I_X1)+(I_X2) // mm^4
29 // Results
30 clc
31 printf('The M.O.I of the composite area about the
    centroidal x-axis is %f mm^4 \n',I_x)
32 // NOTE: The answer given in the text book is
    2.31*10^3 insted of 2.31*10^6.

```

---

**Scilab code Exa 12.10** Moment of Inertia of a Composite area or hollow section

```

1 // Initalization of variables
2 b1=120 // mm // width of the flange pate of L-
    section
3 d1=20 // mm // depth of the flange plate
4 b2=20 // mm // width/thickness of the web
5 d2=130 // mm // depth of the web
6 // Calculations
7 // (a) Location of the centroid of the composite
    area
8 A_1=b1*d1 // mm^2 // area of the flange plate
9 A_2=b2*d2 // mm^2 // area of the web
10 y_1=d2+(d1/2) // mm // y-coordinate of the centroid
11 y_2=d2/2 // mm // y-coordinate of the centroid
12 x_1=60 // mm // x-coordinate of the centroid
13 x_2=110 // mm // x-coordinate of the centroid
14 y_c=((A_1*y_1)+(A_2*y_2))/(A_1+A_2) // mm // from
    the bottom edge
15 x_c=((A_1*x_1)+(A_2*x_2))/(A_1+A_2) // mm // from

```

```

    the bottom edge
16 // (b) Moment of Inertia of the composite area about
    the centroidal x-axis
17 // Area (A_1) M.I of area A_1 about x-axis
18 I_x1=(b1*(d1^3))/12 // mm^4
19 // M.I of the area A_1 about the centroidal x-axis
    of the composite area (By parallel-axis theorem)
20 OC_1=d2+(d1/2) // mm // from the bottom edge
21 OC_2=d2/2 // mm // from the bottom edge
22 OC=y_c // mm // from the bottom edge
23 d_1=(d2-y_c)+(d1/2) // mm
24 d_2=y_c-OC_2 // mm
25 I_X1=(I_x1)+(A_1*d_1^2) // mm^4
26 // Area(A_2) M.I of area A_2 about x-axis
27 I_x2=(b2*d2^3)/12 // mm^4
28 // M.I of the area A_2 about the centroidal x-axis
    of the composite area (By parallel-axis theorem)
29 I_X2=(I_x2)+(A_2*d_2^2) // mm^4
30 // COMPOSITE AREA:M.O.I of the composite area about
    the centroidal x-axis
31 I_x=(I_X1)+(I_X2) // mm^4
32 // (c) Moment of Inertia of the composite area about
    the centroidal y-axis
33 // Area (A_1) M.I of area A_1 about y-axis
34 I_y1=(d1*(b1^3))/12 // mm^4
35 // M.I of the area A_1 about the centroidal y-axis
    of the composite area (By parallel-axis theorem)
36 d_3=x_c-(b1/2) // mm // distance between c &c1 along
    x axis
37 I_Y1=(I_y1)+(A_1*d_3^2) // mm^4
38 // Area(A_2) M.I of area A_2 about y-axis
39 I_y2=(d2*b2^3)/12 // mm^4
40 // M.I of the area A_2 about the centroidal y-axis
    of the composite area (By parallel-axis theorem)
41 d_4=b1-x_c-(b2/2) // mm // distance between c &c2
    along x axis
42 I_Y2=(I_y2)+(A_2*d_4^2) // mm^4
43 // COMPOSITE AREA:M.O.I of the composite area about

```

```

    the centroidal y-axis
44 I_y=(I_Y1)+(I_Y2) // mm^4
45 // Results
46 clc
47 printf('The M.O.I of the composite area about the
    centroidal x-axis is %f mm^4 \n',I_x)
48 printf('The M.O.I of the composite area about the
    centroidal Y-axis is %f mm^4 \n',I_y)
49 // NOTE: The answer for I_x given in text book is
    0.76*10^6 insted of 10.76*10^6

```

---

#### Scilab code Exa 12.14 Product of Inertia

```

1 // Initalization of variables
2 b=1 // cm // smaller side of the L-section
3 h=4 // cm // larger side of the L-section
4 // Calculations
5 // (A) RECTANGLE A_1: Using the paralel axis theorem
6 Ixy=0
7 I_xy1=(Ixy)+((h*b)*(b/2)*(h/2)) // cm^4
8 // (B) RECTANGLE A_2: Using the paralel axis theorem
9 I_xy2=(Ixy)+((b*(h-1))*(1+(3/2))*(b/2)) // cm^4
10 // Product of inertia of the total area
11 I_xy=I_xy1+I_xy2 // cm^4
12 // Calculations
13 clc
14 printf('The Product of inertia of the L-section is
    %f cm^4 \n',I_xy)

```

---

#### Scilab code Exa 12.15 Principal Moment of Inertia

```

1 // Initalization of variables

```

```

2 I_x=1548 // cm^4 // M.O.I of the Z-section about X-
  axis
3 I_y=2668 // cm^4 // M.O.I of the Z-section about Y-
  axis
4 b=12 // cm // width of flange of the Z-section
5 d=3 // cm // depth of flange of the Z-section
6 t=2 // cm // thickness of the web of the Z-section
7 h=6 // cm // depth of the web of the Z-section
8 // Calculations
9 A_1=b*d // cm^2 // area of top flange
10 x_1=-5 // cm // distance of the centroid from X-axis
    for top flange
11 y_1=4.5 // cm // distance of the centroid from Y-
    axis for top flange
12 A_2=t*h // cm^2 // area of web
13 x_2=0 // cm // distance of the centroid from X-axis
    for the web
14 y_2=0 // cm // distance of the centroid from Y-axis
    for the web
15 A_3=b*d // cm^2 // area of bottom flange
16 x_3=5 // cm // distance of the centroid from X-axis
    for top flange
17 y_3=-4.5 // cm // distance of the centroid from Y-
    axis for top flange
18 // Product of Inertia of the total area is ,
19 I_xy=((A_1*x_1*y_1)+(A_3*x_3*y_3)) // cm^4
20 // The direction of the principal axes is ,
21 theta_m=(atand((2*I_xy)/(I_y-I_x)))/2 // degree
22 // Principa M.O.I
23 I_max=((I_x+I_y)/2)+(sqrt(((I_x-I_y)/2)^2+(I_xy)^2))
    // cm^4
24 I_mini=((I_x+I_y)/2)-(sqrt(((I_x-I_y)/2)^2+(I_xy)^2)
    ) // cm^4
25 // Results
26 clc
27 printf('The principal axes of the section about O is
    %f degree \n',theta_m)
28 printf('The Maximum value of principal M.O.I is %f

```



```
    cm^4 \n',I_max)  
29 printf('The Minimum value of principal M.O.I is %f  
    cm^4 \n',I_mini)
```

---

# Chapter 13

## Principle of Virtual Work

Scilab code Exa 13.1 Application of Principle of Virtual Work

```
1 // Initialization of variables
2 W=1000 // N // weight to be raised
3 // Calculations
4 // From the Principle of virtual work,
5 P=W/2 // N
6 // Results
7 clc
8 printf('The value of force (i.e P) that can hold the
        system in equilibrium is %f N \n',P)
```

---

Scilab code Exa 13.7 Application of Principle of Virtual Work

```
1 // Initialization of variables
2 P=1000 // N // Force acting at the hinge of the 1st
   square
3 Q=1000 // N // Force acting at the hinge of the 2nd
   square
4 // Calculations
```

```

5 // Choosing the co-ordinate system with origin at A,
   we can write ,
6 theta=45 // degree
7 // Forces that do work are P,Q & X_B. Applying the
   principle of virtual work & Simplifying and
   solving for X_B,
8 X_B=((2*P)/6)*(cosd(theta)/sind(theta)) // N
9 // Now give a virtual angular displacement to the
   whole frame about end A such that line AB turns
   by an angle delta_phi.
10 // The force doing work are P,Q&Y_B. Applying the
   principle of virtual work & Simplifying this eq'n
   and solving for Y_B,
11 Y_B=((3*Q)+P)/6 // N
12 // Simply by removing the support at A & replacing
   it by the reactions X_A & Y_A we can obtain ,
13 X_A=X_B // N
14 Y_A=P+Q-Y_B // N
15 // Results
16 clc
17 printf('The Horizontal component of reaction at A (
   X_A) is %f N \n',X_A)
18 printf('The Vertical component of reaction at A (Y_A
   ) is %f N \n',Y_A)
19 printf('The Horizontal component of reaction at B (
   X_B) is %f N \n',X_B)
20 printf('The Vertical component of reaction at B (Y_B
   ) is %f N \n',Y_B)

```

---

# Chapter 14

## Rectilinear Motion of a particle

Scilab code Exa 14.3 Displacement velocity and acceleration of connected bodies

```
1 // Initialization of variables
2 a_T=0.18 // m/s^2 // acc of trolley
3 // Calculations
4 a_B=-a_T/3 // m/s^2 // from eq'n 4
5 t=4 // seconds
6 v_T=a_T*t // m/s // velocity of trolley after 4
   seconds
7 v_B=-v_T/3 // m/s // from eq'n 3
8 S_T=(1/2)*a_T*t^2 // m // distance moved by trolley
   in 4 sec
9 S_B=-S_T/3 // m // from eq'n 2
10 // Results
11 clc
12 printf('The acceleration of block B is %f m/s^2 \n',
   a_B)
13 printf('The velocity of trolley & the block after 4
   sec is %f m/s & %f m/s \n',v_T,v_B)
14 printf('The distance moved by the trolley & the
   block is %f m & %f m \n',S_T,S_B)
15 // The -ve sign indicates that the velocity or the
   distance travelled is in opposite direction.
```

---

**Scilab code Exa 14.4 Velocity time relationship**

```
1 // Initilization of variables
2 v_B=12 // cm/s // velocity of block B
3 u=0
4 s=24 // cm // distance travelled by bock B
5 t=5 // seconds
6 // Calculations
7 a_B=v_B^2/(2*s) // cm/s^2 // using eq'n v^2-u^2=28*a
   *s for block B. Here u=0
8 a_A=(3/2)*a_B // cm/s^2 // from eq'n 4 // Here a_A
   is negative which means acceleration is in
   opposite direction. However we consider +ve
   values for further calculations
9 v_A=u+(a_A*t) // m/s // using eq'n v=u+(a*t)
10 S_A=(u*t)+((1/2)*a_A*t^2) // m // using eq'n S=(u*t)
   +((1/2)*a*t^2)
11 // Results
12 clc
13 printf('The acceleration of block A (a_A) is %f cm/s
   ^2 \n',a_A)
14 printf('The acceleration of block B (a_B) is %f cm/s
   ^2 \n',a_B)
15 printf('The velocity of block A (v_A) after 5
   seconds is %f m/s \n',v_A)
16 printf('The position of block A (S_A) after 5
   seconds is %f m \n',S_A)
```

---

**Scilab code Exa 14.5 Displacement time relationship**

```
1 // Initilization of variables
```

```

2 u=72*(1000/(60*60)) // km/hr // speed of the vehicle
3 s=300 // m // distance where the light is turning is
  red
4 t=20 // s // traffic light timed to remain red
5 // Calculations
6 // Now to find the acceleration we use the eq'n s=u*
  t+(1/2)*a*t^2
7 a=((s)-(u*t))*2/t^2 // m/s^2 (Deceleration)
8 v=(u+(a*t))*((60*60)/1000) // km/hr // here we
  multiply with (60*60)/1000 to convert m/s to km/
  hr
9 // Results
10 clc
11 printf('(a) The required uniform acceleration of the
  car is %f m/s^2 \n',a)
12 printf('(b) The speed at which the motorist crosses
  the traffic light is %f km/hr \n',v)

```

---

#### Scilab code Exa 14.6 Displacement time relationship

```

1 // Initialization of variables
2 S=50 // m // height of the tower
3 v=25 // m/s // velocity at which the stone is thrown
  up from the foot of the tower
4 g=9.81 // m/s^2 // acc due to graity
5 // Calculations
6 // The equation of time for the two stones to cross
  each other is given as,
7 t=S/v // seconds
8 S_1=(1/2)*g*t^2 // m // from the top
9 // Results
10 clc
11 printf('The time (t) at which the two stones cross
  each other is %f seconds \n',t)
12 printf('The two stones cross each other (from top)

```

at a distance of %f m \n',S\_1)

---

#### Scilab code Exa 14.7 Displacement time relationship

```
1 // Intilization of variables
2 acc=0.5 // m/s^2 // acceleration of the elevator
3 s=25 // m // distance travelled by the elevator from
   the top
4 u=0 // m/s
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // Using eq'n the eq'n  $v^2-u^2=2*a*s$ , solving for v
   we get ,
8 v=sqrt((2*acc*s)+(u^2)) // m/s
9 // Now solving eq'n 1 & 2 for t we get,  $(4.655*t^2)
   -(5*t)+(25)=0$ 
10 // Find the roots of the eq'n using the eq'n,  $t=(-b+
   sqrt(b^2-(4*a*c)))/(2*a)$ . In this eq'n the values
   of a,b & c are ,
11 a=4.655
12 b=-5
13 c=-25
14 t=(-b+sqrt((b^2)-(4*a*c)))/(2*a) // seconds
15 // Let S_1 be the distance travelled by the elevator
   after it travels 25 m from top when the stone
   hits the elevator, This disance S_1 is given as ,
16 S_1=(v*t)+((1/2)*acc*t^2) // m
17 // Let S be the total dist from top when the stone
   hits the elevator ,
18 S=S_1+s // m
19 // Results
20 clc
21 printf('The time taken by the stone to hit the
   elevator is %f seconds \n',t)
22 printf('The distance (S)travelled by the elevator at
```

the time of impact is %f m \n',S)

---

#### Scilab code Exa 14.9 Displacement time relationship

```
1 // Initialization of variables
2 v=60 // km/hr // velocity of the train
3 d1=15 // km // Distance travelled by the local train
   // from the velocity-time graph (here d1= Area OED)
4 d2=12 // km // from the velocity-time graph (here d2
   // = Area OABB')
5 d3=3 // km // from the velocity-time graph (here d3=
   // Area BB'C)
6 // Calculations
7 t_1=d2/v // hr // time of travel for first 12 kms
8 t_2=(2*d3)/v // hr // time of for next 3 kms
9 // Total time of travel for passenger train is given
   // by eq'n
10 t=t_1+t_2 // hr
11 // Now time of travel of the local train (let it be
   // T) is given as,
12 T=2*t // hr
13 V_max=(2*d1)/T // km/hr
14 // Results
15 clc
16 printf('The maximum speed of the local train is %f
   // km/hr \n',V_max)
```

---

#### Scilab code Exa 14.10 Distance travelled by a particle in the n th second

```
1 // Initialization of variables
2 a=10 // m/s^2 // acceleration of the particle
3 S_5th=50 // m // distance travelled by the particle
   // during the 5th second
```



```

4 t=5 // seconds
5 // Calculations
6 // The distance travelled by the particle in time t
  is given by,  $S=(u*t)+(1/2)*a*t^2$ .....(consider
  this as eq'n 1)
7 // Here, The distance travelled by the particle in
  the 5th second=The distance travelled in 5
  seconds – The distance travelled in 4 seconds
  ..... (consider eq'n 2)
8 // Using eq'n 1:  $S_{(0-5)}=(5*u)+(1/2)*10*5^2 = 5*u$ 
  +125.....(consider eq'n 3)
9 // again,  $S_{(0-4)}=(4*u)+(1/2)*10*4^2 = 4*u+80$ ....(
  consider eq'n 4)
10 // Now,put eq'n 3&4 in eq'n 2 and solve for u. We
  get,  $50=[(5*u+125)-(4*u+80)]$  i.e  $50=u+45$ 
11  $u=(S_{5th})-45$  // m/s
12 // Calculations
13 clc
14 printf('The initial velocity of the particle is %f m
  /s \n',u)

```

---

#### Scilab code Exa 14.11 Variable acceleration

```

1 // Initalization of variables
2 // Conditions given are
3 t=1 // s
4 x=14.75 // m
5 v=6.33 // m/s
6 // Calculations
7 // We use expression 1,2 & 3 to find distance ,
  velocity & acceleration of the particle after 2
  sec
8 T=2 // sec
9  $X=(T^4/12)-(T^3/3)+(T^2)+(5*T)+9$  // m // eq'n 3
10  $V=(T^3/3)-(T^2)+(2*T)+5$  // m/s

```

```

11 a=(T^2)-(2*T)+2 // m/s^2
12 // Results
13 clc
14 printf('The distance travelled by the particle is %f
        m \n',X)
15 printf('The velocity of the particle is %f m/s \n',V
        )
16 printf('The acceleration of the particle is %f m/s^2
        \n',a)
17 // The answer may vary due to decimal point error

```

---

#### Scilab code Exa 14.12 Variable acceleration

```

1 // Calculations
2 // From eq'n 2 it is clear that velocity of the
  particle becomes zero at t=3 sec
3 t=3 // sec .. from eq'n 2
4 // Position of particle at t=3 sec
5 x=(t^3)-(3*t^2)-(9*t)+12 // m // from eq'n 1
6 // Acc of particle at t=3 sec
7 a=6*(t-1) // m/s^2 // from eq'n 3
8 // Results
9 clc
10 printf('The time at which the velocity of the
        particle becomes zero is %f sec \n',t)
11 printf('The position of the particle at t=3 sec is %f
        m \n',x)
12 printf('The acceleration of the particle is %f m/s^2
        \n',a)
13 // Ref textbook for the graphs

```

---

#### Scilab code Exa 14.15 D AlembertsPrinciple

```

1 // Initalization of variables
2 F=250 // N // Force acting on a body
3 m=100 // kg // mass of the body
4 // Calculations
5 // Using the eq'n of motion
6 a=F/m // m/s^2
7 // Results
8 clc
9 printf('The acceleration of the body is %f m/s^2 \n',
        ,a)

```

---

#### Scilab code Exa 14.16 D AlembertsPrinciple

```

1 // Initalization of variables
2 a=1 // m/s^2 // downward/upward acceleration of the
    elevator
3 W=500 // N // Weight of man
4 g=9.81 // m/s^2 // acceleration due to gravity
5 // Calculations
6 // (a) Downward Motion
7 R_1=W*(1-(a/g)) // N // (Assume pressure as R_1)
8 // (b) Upward Motion
9 R_2=W*(1+(a/g)) // N // (Assume pressure as R_2)
10 // Results
11 clc
12 printf('(a) The pressure transmitted to the floor by
    the man for Downward motion of the elevator is
    %f N \n',R_1)
13 printf('(b) The pressure transmitted to the floor by
    the man for Upward motion of the elevator is %f
    N \n',R_2)

```

---

#### Scilab code Exa 14.17 D Alamberts Principle

```

1 // Initalization of variables
2 W=5000 // N // Total weight of the elevator
3 u=0 // m/s
4 v=2 // m/s // velocity of the elevator
5 s=2 // m // distance traveled by the elevator
6 t=2 // seconds // time to stop the lift
7 w=600 // N // weight of the man
8 g=9.81 // m/s^2 // acc due to gravity
9 // Calculations
10 // Acceleration acquired by the elevator after
    travelling 2 m is given by,
11 a=sqrt((v^2-u^2)/(2*s)) // m/s^2
12 // (a) Let T be the the tension in the cable which
    is given by eq'n,
13 T=W*(1+(a/g)) // N
14 // (b) Motion of man
15 // Let R be the pressure experinced by the man.Then
    from the Eq'n of motion of man pressure is given
    as ,
16 R=w*(1-(a/g)) // N
17 // Results
18 clc
19 printf('(a) The Tensile force in the cable is %f N \
    n',T)
20 printf('(b) The pressure transmitted to the floor by
    the man is %f N \n',R)

```

---

#### Scilab code Exa 14.18 Motion of two Bodies

```

1 // Initalization of variables
2 M_1=10 // kg // mass of the 1st block
3 M_2=5 // kg // mass of the 2nd block
4 mu=0.25 // coefficient of friction between the
    blocks and the surface
5 g=9.81 // m/s^2 // acc due to gravity

```

```

6 // Calculations
7 a=g*(M_2-(mu*M_1))/(M_1+M_2) // m/s^2 // from eq'n 5
8 T=M_1*M_2*g*(1+mu)/(M_1+M_2) // N // from eq'n 6
9 // Results
10 clc
11 printf('The acceleration of the masses is %f m/s^2 \
n',a)
12 printf('The tension in the string is %f N \n',T)

```

---

#### Scilab code Exa 14.19 Motion of two Bodies

```

1 // Initialization of variables
2 M_1=150 // kg // mass of the 1st block
3 M_2=100 // kg // mass of the 2nd block
4 mu=0.2 // coefficient of friction between the blocks
and the inclined plane
5 g=9.81 // m/s^2 // acc due to gravity
6 theta=45 // degree // inclination of the surface
7 // Calculations
8 // substituting the value of eq'n 3 in eq'n 1 &
solving for T,we get value of T as,
9 T=((M_1*M_2*g)*(sind(theta)+2-(mu*cosd(theta))))
/((4*M_1)+(M_2)) // N
10 // Results
11 clc
12 printf('The tension in the string during the motion
of the system is %f N \n',T)

```

---

#### Scilab code Exa 14.20 Motion of two Bodies

```

1 // Initialization of variables
2 M_1=5 // kg // mass of the 1st block
3 theta_1=30 // degree // inclination of the 1st plane

```

```

4 M_2=10 // kg // mass of the 2nd block
5 theta_2=60 // degree // inclination of the 2nd plane
6 mu=0.33 // coefficient of friction between the
    blocks and the inclined plane
7 g=9.81 // m/s^2 // acc due to gravity
8 // Calculations
9 // solving eq'n 1 & 2 for a we get ,
10 a((((M_2*(sind(theta_2)-(mu*cosd(theta_2)))))-(M_1*(
    sind(theta_1)+(mu*cosd(theta_1))))))*g)/(M_1+M_2)
    // m/s^2
11 // Results
12 clc
13 printf('The acceleration of the masses is %f m/s^2 \
    n',a)

```

---

#### Scilab code Exa 14.21 Motion of two Bodies

```

1 // Initalization of variables
2 S=5 // m // distance between block A&B
3 mu_A=0.2 // coefficient of friction between the
    block A and the inclined plane
4 mu_B=0.1 // coefficient of friction between the
    block B and the inclined plane
5 theta=20 // degree // inclination of the pane
6 g=9.81 // m/s^2 // acc due to gravity
7 // Calculatio//
8 // EQUATION OF MOTION OF BLOCK A:
9 // Let a_A & a_B be the acceleration of block A & B.
10 a_A=(g*sind(theta))-(mu_A*g*cosd(theta)) // m/s^2 //
    from eq'n 1 & eq'n 2
11 // EQUATION OF MOTION OF BLOCK B:
12 a_B=g*((sind(theta))-(mu_B*cosd(theta))) // m/s^2 //
    from eq'n 3 & Rb
13 // Now the eq'n for time of collision of the blocks
    is given as ,

```

```

14 t=sqrt((S*2)/(a_B-a_A)) // seconds
15 S_A=(1/2)*a_A*t^2 // m // distance travelled by
    block A
16 S_B=(1/2)*a_B*t^2 // m // distance travelled by
    block B
17 // Results
18 clc
19 printf('The time before collision is %f seconds \n',
    t)
20 printf('The distance travelled by block A before
    collision is %f m \n',S_A)
21 printf('The distance travelled by block B before
    collision is %f m \n',S_B)

```

---

#### Scilab code Exa 14.22 Motion of two Bodies

```

1 // Initialization of variables
2 P=50 // N // Weight of the car
3 Q=100 // N // Weight of the rectangular block
4 g=9.81 // m/s^2 // acc due to gravity
5 b=25 // cm // width of the rectangular block
6 d=50 // cm // depth of the block
7 // Calculations
8 a=(Q*g)/(4*P+2*Q) // m/s^2 // from eq'n 4
9 W=(Q*(P+Q))/(4*P+Q) // N // from eq'n 6
10 // Results
11 clc
12 printf('The maximum value of weight (W) by which the
    car can be accelerated is %f N \n',W)
13 printf('The acceleration is %f m/s^2 \n',a)

```

---

#### Scilab code Exa 14.23 Motion of two Bodies

```

1 // Initalization of variables
2 P=40 // N // weight on puley r_1
3 Q=60 // N // weight on pulley r_2
4 g=9.81 // m/s^2 // acc due to gravity
5 // Calculations
6 // The eq'n for acceleration of pulley P i.e a_p is ,
7 a_p=((2*P)-(Q))/((4*P)+(Q))*2*g // m/s^2
8 // Results
9 clc
10 printf('The downward acceleration of P is %f m/s^2 \
n',a_p)

```

---

#### Scilab code Exa 14.24 Motion of two Bodies

```

1 // Initalization of variables
2 M=15 // kg // mass of the wedge
3 m=6 // kg // mass of the block
4 theta=30 // degree // angle of the wedge
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 a_A=((m*g*cosd(theta)*sind(theta))/((M)+(m*(sind(
theta))^2)))/(g) // g // By eliminating R_1 from
eq'n 1&3.
8 // Here, assume a_r is the acceleration of block B
relative to wedge A which is given by substuting
a_A in eq'n 2
9 a_r=((g*sind(theta))*(m+M))/((M)+(m*(sind(theta))
^2)))/(g) // g
10 // Results
11 clc
12 printf('(a) The acceleration of the wedge is %f g \n
',a_A)
13 printf('(b) The acceleration of the bock relative to
the wedge is %f g \n',a_r)

```

---



### Scilab code Exa 14.25 Motion of two Bodies

```
1 // Initialization of variables
2 P=30 // N // weight on pulley A
3 Q=20 // N // weight on pulley B
4 R=10 // N // weight on puey B
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // Solving eqn's 6 & 7 using matrix for a & a_1, we
  get
8 A=[70 -40;-10 30]
9 B=[10;-10]
10 C=inv(A)*B
11 // Acceleration of P is given as ,
12 P=C(1) // m/s^2
13 // Acceleration of Q is given as ,
14 Q=C(2)-C(1) // m/s^2
15 // Acceleration of R is given as ,
16 R=-(C(2)+C(1)) // m/s^2 // as R is taken to be +ve
17 // Results
18 clc
19 printf('The acceleration of P is %f g \n',P)
20 printf('The acceleration of Q is %f g \n',Q)
21 printf('The acceleration of R is %f g \n',R)
22 // Here the -ve sign indicates deceleration or
  backward/downward acceleation.
```

---

### Scilab code Exa 14.30 Motion of two Bodies

```
1 // Initialization of variables
2 W=1 // kg/m // weight of the bar
3 L_AB=0.6 // m // length of segment AB
```

```

4 L_BC=0.30 // m // length of segment BC
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // Consider the respective F.B.D.
8 theta_1=atand(5/12) // slope of bar AB // here
   theta_1= atan(theta)
9 theta_2=asind(5/13) // theta_2=asin(theta)
10 theta_3=acosd(12/13) // theta_3=acos(theta)
11 M_AB=L_AB*W // kg acting at D // Mass of segment AB
12 M_BC=L_BC*W // kg acting at E // Mass of segment BC
13 // The various forces acting on the bar are:
14 // Writing the eqn's of dynamic equilibrium
15 Y_A=(L_AB*g)+(L_BC*g) // N // sum F_y=0
16 // Using moment eq'n Sum M_A=0:Here, in this eq'n the
   values are as follows ,
17 AF=L_BC*cosd(theta_3)
18 DF=L_BC*sind(theta_2)
19 AH=(L_AB*cosd(theta_3))+((L_BC/2)*sind(theta_2))
20 IG=(L_AB*sind(theta_2))-((L_BC/2)*cosd(theta_3))
21 // On simplifying and solving moment eq'n we get a
   as ,
22 a=((2*L_AB*L_BC*g*sind(theta_2))-(L_BC*g*(L_BC/2)*
   cosd(theta_3)))/((2*L_AB*L_BC*cosd(theta_3))+
   L_BC*(L_BC/2)*sind(theta_2)) // m/s^2
23 X_A=0.9*a //N // from eq'n of dynamic equilibrium
24 R_A=sqrt(X_A^2+Y_A^2) // N // Resultant of R_A
25 alpha=atand(Y_A/X_A) // degree
26 // Results
27 clc
28 printf('The acceleration is %f m/s^2 \n',a)
29 printf('The reaction at A (R_A) is %f N \n',R_A)
30 printf('The angle made by the resultant is %f degree
   \n',alpha)

```

---

# Chapter 15

## Curvilinear motion of a particle

Scilab code Exa 15.2 Components of Acceleration

```
1 // Initialization of variables
2 r=200 // m // radius of the curved road
3 v_1=72*(1000/3600) // m/s // initial speed of the
   car
4 v_2=36*(1000/3600) // m/s // speed of the car after
   10 seconds
5 t=10 // seconds
6 // Calculations
7 A_n=v_1^2/r // m/s^2 // normal component of
   acceleration
8 A_t=0 // since dv/dt=0 // tangential component of
   acceleration
9 delv=v_1-v_2
10 delt=t-0
11 a_t=delv/delt // m/s^2 // tangential component of
   deceleration after the brakes are applied
12 a_n=v_1^2/r // m/s^2 // normal component of
   deceleration after the brakes are applied
13 // Results
14 clc
15 printf('The normal component of acceleration is %f m
```

```

    /s^2 \n',A_n)
16 printf('The tangential component of acceleration is
    %f m/s^2 \n',A_t)
17 printf('The normal component of deceleration is %f m
    /s^2 \n',a_n)
18 printf('The tangential component of deceleration is
    %f m/s^2 \n',a_t)

```

---

### Scilab code Exa 15.3 Components of Acceleration

```

1 // Initialization of variables
2 r=250 // m // radius of the curved road
3 a_t=0.6 // m/s^2 // tangential acceleration
4 a=0.75 // m/s^2 // total acceleration attained by
    the car
5 // Calculations
6 a_n=sqrt(a^2-a_t^2) // m/s^2
7 v=sqrt(a_n*r) // m/s
8 // Using v=u+a*t
9 u=0
10 t=v/a_t // seconds
11 // Now using v^2-u^2=2*a*s
12 s=v^2/(2*a_t) // m
13 // Results
14 clc
15 printf('The distance traveled by the car is %f m \n
    ',s)
16 printf('The time for which the car travels is %f
    seconds \n',t)

```

---

### Scilab code Exa 15.5 Motion of a particle on a curved frictionless path

```

1 // Initialization of variables

```

```

2 v=10 // m/s // speed of the car
3 r=200 // m // radius of the road
4 t=15 // seconds
5 // Calculations
6 omega=(v/r) // radian/seconds // angular velocity of
   the car
7 // Velocity in x & y direction is given by eq'n
8 v_x=omega*r*sind(omega*(180/%pi)*t) // m/s // value
   of v_x is -ve but we consider it to be +ve for
   calculations
9 v_y=omega*r*cosd(omega*(180/%pi)*t) // m/s
10 // Acceleration in x & y direction is given by
11 a_x=omega^2*r*cosd(omega*(180/%pi)*t) // m/s^2 //
   value of a_x is -ve but we consider it to be +ve
   for calculations
12 a_y=omega^2*r*sind(omega*(180/3.14)*t) // m/s^2 //
   value of a_y is -ve but we consider it to be +ve
   for calculations
13 a=sqrt(a_x^2+a_y^2) // m/s^2 // total acc
14 phi=atand(a_y/a_x) // degrees // direction of
   acceleration
15 // Components in tangential and normal directions
16 // Velocity
17 v_n=0 // m/s
18 v_t=v // m/s
19 // Acceleration
20 a_n=v^2/r // m/s^2 // normal acc
21 a_t=0 // tangential acc
22 // angular position of the car after 15 sec
23 theta=omega*(180/%pi)*t // degrees
24 // Results
25 clc
26 printf('The component of velocity in X direction (
   v_x) is %f m/s \n',v_x)
27 printf('The component of velocity in Y direction (
   v_y) is %f m/s \n',v_y)
28 printf('The component of acceleration in X direction
   (a_x) is %f m/s^2 \n',a_x)

```

```

29 printf('The component of acceleration in Y direction
      (a_y) is %f m/s^2 \n',a_y)
30 printf('The total acceleration is %f m/s^2 and its
      direction is %f degrees \n',a,phi)
31 printf('The normal acceleration is %f m/s^2 and
      tangential acceleration is %f m/s^2 \n',a_n,a_t)

```

---

### Scilab code Exa 15.6 Motion of a particle on a curved frictionless path

```

1 // Initalization of variables
2 t=1 // seconds
3 pi=3.14
4 // Calculations
5 // From the equations of r and theta given we find 1
      st & 2nd derative and substitute t=1sec Here we
      consider the 1st derivate as r_1 & theta_1 and so
      on...
6 r=(1.25*t^2)-(0.9*t^3) // m
7 r_1=(1.25*(2*t))-(0.9*(3*t^2)) // m/s
8 r_2=2.5-(0.9*3*(2*t)) // m/s^2
9 theta=(pi/2)*(4*t-3*t^2) // radian
10 theta_1=(pi/2)*(4-(6*t)) // rad/second
11 theta_2=(pi/2)*(0-(6*t)) // rad/second^2
12 // Velocity of collar P
13 v_r=r_1 // m/s
14 v_theta=r*theta_1 // m/s
15 v=sqrt(v_r^2+v_theta^2) // m/s
16 alpha=atand(v_theta/v_r) // degree
17 // Acceleration of the collar P
18 a_r=r_2-(r*theta_1^2) // m/s^2
19 a_theta=(r*theta_2)+(2*r_1*theta_1) // m/s^2
20 a=sqrt(a_r^2+a_theta^2) // m/s^2
21 beta=atand(a_theta/a_r) // degree
22 // Acceleration of collar P relative to the rod. Let
      it be a_relative

```

```

23 a_relative=r_2 // m/s^2 // towards O
24 // Calculations
25 clc
26 printf('The velocity of the collar is %f m/s \n',v)
27 printf('The acceleration of the collar is %f m/s^2 \
n',a)
28 printf('The acceleration of the collar relative to
the rod is %f m/s^2 \n',a_relative)

```

---

### Scilab code Exa 15.7 Components of motion

```

1 // Consider the eq'ns of motion from the book
2 // The notations have been changed for the
   derivatives of r & theta
3 // (1) At t=0 s
4 theta_0=0
5 theta_1=2*pi // rad/s
6 theta_2=0
7 r_0=0
8 r_1=10 // cm/s
9 r_2=0
10 // At t=0.3 s
11 t=0.3 // sec
12 theta=2*pi*t // rad
13 theta1=2*pi // rad/s
14 theta2=0
15 r=10*t // cm
16 r1=10 // cm/s
17 r2=0
18 // (i)
19 // Velocity
20 v_r=r_1 // cm/s
21 v_theta=r_0*theta_1
22 v=sqrt(v_r^2+v_theta^2) // cm/s
23 // Acceleration

```

```

24 a_r=r_2-(r_0*theta_1^2) // cm/s^2
25 a_theta=(r_0*theta_2)+(2*r_1*theta_1) // cm/s^2
26 a=sqrt(a_r^2+a_theta^2) // cm/s^2
27 // (ii)
28 // Velocity
29 V_R=r1 // cm/s
30 V_theta=r*theta1 // cm/s
31 V=sqrt(V_R^2+V_theta^2) // cm/s
32 // Acceleration
33 A_r=r2-(r*theta1^2) // cm/s^2
34 A_theta=(r*theta2)+(2*r1*theta1) // cm/s^2
35 A=sqrt(A_r^2+A_theta^2) // cm/s^2
36 // Results
37 clc
38 printf('The velocity and the acceleration of the
    partice at t=0 s is %f cm/s & %f cm/s^2 \n',v,a)
39 printf('The velocity and the acceleration of the
    partice at t=0.3 s is %f cm/s & %f cm/s^2 \n',V,A
    )

```

---

#### Scilab code Exa 15.9 Equations of dynamic equilibrium

```

1 // Calculations
2 // Tension in the wire before it is cut
3 T_ab=1/((2.747*0.643)+(0.766)) // From eqn's 1 & 2..
    Here T_ab is multiplied with W (i.e weight of
    small ball)
4 T_AB=cosd(40) // Tension in the wire after the wire
    is cut. Again T_AB is multiplied with W.
5 // Results
6 clc
7 printf('The tension in the wire before and after it
    is cut is respectively %f W & %f W \n',T_ab,T_AB)

```

---



**Scilab code Exa 15.10** Equations of dynamic equilibrium

```
1 // Initialization of variables
2 mu_a=0.40 // coefficient of friction under block A
3 n=40 // r.p.m // speed of rotation of frame
4 W_A=120 // N // weight of block A
5 W_B=80 // N // weight of block B
6 r_1=1.2 // m // distance between W_A & axis of
   rotation
7 r_2=1.6 // m // distance between W_B & axis of
   rotation
8 g=9.81 // m/s^2
9 // Calculations
10 // Consider the F.B.D of block A
11 N=W_A // N // sum F_y=0
12 omega=(2*pi*n)/60 // rad/sec
13 a_n=omega^2*r_1 // m/s^2
14 T=((W_A/g)*a_n)-(mu_a*W_A) // N
15 // Now consider the F.B.D of block B
16 A_n=omega^2*r_2 // m/s^2
17 N_1=(W_B/g)*A_n // N // sum F_x=0
18 mu=(T-W_B)/N_1 // sum F_x=0
19 // Results
20 clc
21 printf('The coefficient of friction of block B is %f
   \n',mu)
```

---

**Scilab code Exa 15.12** Motion of particle on curved frictionless path

```
1 // Initialization of variables
2 W=10000 // N // Weight of the locomotive
3 // Calculations
```

```

4 // Consider the various derivations given in the
   textbook
5 R_max=W/20 // N // eq'n for max reaction
6 // The position of occurrence of maximum thrust
   cannot be defined here. Refer textbook for the
   answer
7 // Results
8 clc
9 printf('The maximum lateral thrust is %f N \n',R_max
   )

```

---

**Scilab code Exa 15.13** motion of a particle on curved frictionless path

```

1 // Initialization of variables
2 W=10 // N // Weight of the ball
3 // Calculations
4 // consider the eq'n derived to find the reaction ,
   given as
5 R=W*(1+((2*pi^2)/9)) // N
6 // Results
7 clc
8 printf('The value of the reaction is %f N \n',R)

```

---

**Scilab code Exa 15.15** Motion of a particle in a curved frictionless path

```

1 // Initialization of variables
2 P=50 // N // Weight of ball P
3 Q=50 // N // Weight of ball Q
4 R=100 // N // Weight of the governing device
5 l=0.3 // m // length of each side
6 theta=30 // degree
7 g=9.81 // m/s^2 // acc due to gravity
8 // Calculations

```

```

9 // Consider the respective F.B.D
10 r=l*sind(theta) // m // Radius of circe
11 // On solving eqn's 1,2 &3 we get the value of v as,
12 v=sqrt(((Q+R)*g*r)/((sqrt(3))*Q)) // m/s
13 // But the eq'n v=omega*r we get the value of N as,
14 N=(60*v)/(2*pi*r) // r.p.m
15 // Results
16 clc
17 printf('The speed of rotation is %f r.p.m \n',N)
18 // NOTE: In the text book (A.K. Tayal) this sum is
    numbered as 'EXAMPLE 15.14' which is incorrect.

```

---

**Scilab code Exa 15.16** Motion of a particle in a curved frictionless path

```

1 // Initalization of variables
2 Q=20 // N // Weight of the governor device
3 W=10 // N // Weight of the fly balls
4 theta=30 // degree // angle between the vertical
    shaft and the axis AB
5 l=0.2 // m // length of the shaft
6 g=9.81 // m/s^2 // acc due to gravity
7 // Calculations
8 // Consider the respective F.B.D
9 // Radius of the circle is given as,
10 r=Q*sind(theta)*(10^-2) // m
11 // Solving eq'n 1 & 2 for v. The eq'n for v is given
    as,
12 v=sqrt(((W*l*0.5)+(0.05*Q))/((W*0.2*sqrt(3))/(2*g*r)
    )) // m/s
13 // But, v=r*omega=2*pi*N*r/60. From this eq'n we get
    N as,
14 N=(v*60)/(2*pi*r) // r.p.m.
15 // Results
16 clc
17 printf('The speed of the fly-balls is %f r.p.m \n',N)

```

)

---

**Scilab code Exa 15.18** Motion of vehicles on leveled and banked roads

```
1 // Initialization of variables
2 r=50 // m // radius of the road
3 mu=0.15 // coefficient of friction between the
   wheels and the road
4 g=9.81 // m/s^2 // acc due to gravity
5 // Calculations
6 // The eq'n fo max speed of the vehicle without
   skidding is
7 v=sqrt(mu*g*r) // m/s
8 // The angle theta made with the vertical while
   negotiating the corner is
9 theta=atand(v^2/(g*r)) // degree
10 // Results
11 clc
12 printf('The maximum speed with which the vehicle can
   travel is %f m/s \n',v)
13 printf('The angle made with the vertical is %f
   degree \n',theta)
```

---

**Scilab code Exa 15.19** Motion of vehicles on leveled and banked roads

```
1 // Initialization of variables
2 v=100*(1000/3600) // m/s // or 100 km/hr
3 r=250 // m // radius of the road
4 g=9.81 // m/s^2 // acc due to gravity
5 // Calculations
6 // The angle of banking is given by eq'n,
7 theta=atand((v^2)/(g*r)) // degree
8 // Results
```

```

9  clc
10 printf('The angle of banking of the track is %f
    degree \n',theta)

```

---

**Scilab code Exa 15.20** Motion of vehicles on leveled and banked roads

```

1  // Initalization of variables
2  W=10000 // N // Weight of the car
3  r=100 // m // radius of the road
4  v=10 // m/s // speed of the car
5  h=1 // m // height of the C.G of the car above the
    ground
6  b=1.5 // m // distance between the wheels
7  g=9.81 // m/s^2 // acc due to gravity
8  // Calculations
9  // The reactions at the wheels are given by te eq'ns
    :
10 R_A=(W/2)*(1-((v^2*h)/(g*r*b))) // N // Reaction at
    A
11 R_B=(W/2)*(1+((v^2*h)/(g*r*b))) // N // Reaction at
    B
12 // The eq'n for max speed to avoid overturning on
    level ground is ,
13 v_max=sqrt((g*r*(b/2))/(h)) // m/s
14 // Results
15 clc
16 printf('The reaction at Wheel A (R_A) is %f N \n',
    R_A)
17 printf('The reaction at Wheel B (R_B) is %f N \n',
    R_B)
18 printf('The maximum speed at which the vehicle can
    travel without the fear of overturning is is %f m
    /s \n',v_max)

```

---

Scilab code Exa 15.21 Motion of vehicles on leveled and banked roads

```
1 // Initialization of variables
2 W=1 // N // Weight of the bob
3 theta=8 // degree // angle made by the bob with the
   vertical
4 r=100 // m // radius of the curve
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // from eq'n 1 & 2 we get v as,
8 v=(sqrt(g*r*tand(theta)))*(3600/1000) // km/hr
9 T=W/cosd(theta) // N // from eq'n 2
10 // Results
11 clc
12 printf('The speed of the carriage is %f km/hr \n',v)
13 printf('The tension in the chord is %f N \n',T)
```

---

## Chapter 16

# Kinetics of a Particle Work and Energy

Scilab code Exa 16.1 Work of the Force of Spring

```
1 // Initialization of variables
2 k=1000 // N/m // stiffness of spring
3 x_1=0.1 // m // distance upto which the spring is
   stretched
4 x_2=0.2 // m
5 x_0=0 // initial position of spring
6 // Calculations
7 // Work required to stretch the spring by 10 cm from
   undeformed position is given as,
8 U_10=-((k/2)*(x_1^2-x_0^2)) // N-m
9 // Work required to stretch from 10 cm to 20 cm is ,
10 U=-((1/2)*k*(x_2^2-x_1^2)) // N-m
11 // Results
12 clc
13 printf('The work of the spring force is %f N-m \n',
   U_10)
14 printf('The work required to stretch the spring by
   20 cm is %f N-m \n',U)
```

---

### Scilab code Exa 16.3 Work and energy principle for a system of particles

```
1 // Initialization of variables
2 M_A=100 // kg // mass of block A
3 M_B=150 // kg // mass of block B
4 mu=0.2 // coefficient of friction between the blocks
   and the surface
5 x=1 // m // distance by which block A moves
6 g=9.81 // m/s^2 // acc due to gravity
7 // Calculations
8 // Consider the respective F.B.D
9 // Applying the principle of work and energy to the
   system of blocks A&B and on simplifying we get
   the value of v as ,
10 v=sqrt((( -mu*M_A*g)+(M_B*g))/(125)) // m/s
11 // Results
12 clc
13 printf('The velocity of block A is %f m/s \n',v)
```

---

### Scilab code Exa 16.4 Power

```
1 // Initialization of variables
2 M=500*10^3 // kg // mass of the train
3 u=0 // m/s // initial speed
4 v=90*(1000/3600) // m/s // final speed
5 t=50 // seconds
6 F_r=15*10^3 // N // Frictional resistance to motion
7 // Calculations
8 // Acceleration is given as ,
9 a=v/t // m/s^2
10 // The total force required to accelerate the train
   is ,
```



```

11 F=M*a // N
12 // The maximum power required is at, t=50s & v=25 m/
    s
13 P=(F+F_r)*v*(10^-6) // MW
14 // At any time after 50 seconds, the force required
    only to overcome the frictional resistance of
    15*10^3 N is ,
15 P_req=F_r*v*(10^-3) // kW
16 // Results
17 clc
18 printf('(a) The maximum power required is %f MW \n',
    P)
19 printf('(b) The power required to maintain a speed
    of 90 km/hr is %f kW \n',P_req)

```

---

### Scilab code Exa 16.5 Principle of conservation of energy

```

1 // Initilization of variables
2 W=50 // N // Weight suspended on spring
3 k=10 // N/cm // stiffness of the spring
4 x_2=15 // cm // measured extensions
5 h=10 // cm // height for position 2
6 // Calculations
7 // Consider the required F.B.D.
8 // POSITION 1: The force exerted by the spring is ,
9 F_1=W // N
10 // Extension of spring from undeformed position is
    x_1 ,
11 x_1=F_1/k // cm
12 // POSITION 2: When pulled by 10 cm to the floor. P.
    E of weight is ,
13 P.E_g=-W*h // N-cm // (P.E_g= P.E_gravity)
14 // P.E of the spring with respect to position 1
15 P.E_s=(1/2)*k*(x_2^2-x_1^2) // N-cm // (P.E_s= P.E_
    spring)

```

```

16 // Total P.E of the system with respect to position
    1
17 P.E_t=P.E_g+P.E_s // N-cm // (P.E_t= P.E_total)
18 // Total energy of the system ,
19 E_2=P.E_t // N-cm
20 // Total energy of the system in position 3 w.r.t
    position 1 is:
21 x=-sqrt(100) // cm
22 x=+sqrt(100) // cm
23 // Results
24 clc
25 printf('The potential energy of the system is %f N-
    cm \n',E_2)
26 printf('The maximum height above the floor that the
    weight W will attain after release is %f cm \n',x
    )

```

---

### Scilab code Exa 16.6 Principle of conservation of energy

```

1 // Initalization of variables
2 m=5 // kg // mass of the ball
3 k=500 // N/m // stiffness of the spring
4 h=10 // cm // height of drop
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // Consider the respective F.B.D.
8 // In eq'n 1 substitute the respective values and
    simplify it further. In this eq'n of 2nd degree a
    =1 b=-0.1962 & c=-0.01962. Thus the roots of the
    eq'n is given as,
9 a=1
10 b=-0.1962
11 c=-0.01962
12 delta=((-b+(sqrt((b^2)-(4*a*c))))/(2*a))*(10^2) //
    cm // We consider the +ve value of delta

```

```

13 // Results
14 clc
15 printf('The maximum deflection of the spring is %f
        cm \n',delta)

```

---

**Scilab code Exa 16.7 Principle of conservation of energy**

```

1 // Initialization of variables
2 m=5 // kg // mass of the ball
3 k=500 // N/m // stiffness of the spring
4 h=0.1 // m // height of vertical fall
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // Consider the respective F.B.D
8 // On equating the total energies at position 1 & 2
  we get eq'n of delta as,
9 delta=sqrt((2*m*g*h)/(k)) // m
10 // Results
11 clc
12 printf('The maximum compression of the spring is %f
        m \n',delta)

```

---

**Scilab code Exa 16.9 Principle of conservation of energy**

```

1 // Initialization of variables
2 m=5 // kg // mass of the collar
3 k=500 // N/m // stiffness of the spring
4 AB=0.15 // m // Refer the F.B.D for AB
5 AC=0.2 // m // Refer the F.B.D for AC
6 g=9.81 // m/s^2 // acc due to gravity
7 // Calculations
8 // Consider the respective F.B.D
9 // POSITION 1:

```

```

10 P.E_1=m*g*(AB)+0
11 K.E_1=0
12 E_1=P.E_1+K.E_1 //
13 // POSITION 2 : Length of the spring in position 2
14 CB=sqrt(AB^2+AC^2) // m
15 // x is the extension in the spring
16 x=CB-AC // m
17 // On substituting and Equating equations of total
    energies for position1 & position 2 we get the
    value of v as,
18 v=sqrt(((E_1-((1/2)*k*x^2))*2)/m) // m/s
19 // Results
20 clc
21 printf('The velocity of the collar will be %f m/s \n
    ',v)
22 // The answer given in the text book (v=16.4 m/s) is
    wrong.

```

---

### Scilab code Exa 16.10 Principle of work and energy

```

1 // Initialization of variables
2 m=5 // kg // mass of the block
3 theta=30 // degree // inclination of the plane
4 x=0.5 // m // distance travelled by the block
5 k=1500 // N/m // stiffness of the spring
6 mu=0.2 // coefficient of friction between the block
    and the surface
7 g=9.81 // m/s^2 // acc due to gravity
8 // Calculations
9 // Consider the F.B.D of the block
10 // Applying the principle of work and energy between
    the positions 1 & 2 and on further
    simplification we get the generic eq'n for delta
    as,  $750*\delta^2 - 16.03*\delta - 8.015 = 0$ . From this eq
    'n e have values of a,b & c as,

```

```

11 a=750
12 b=-16.03
13 c=-8.015
14 // Thus the roots of the eq'n are given as,
15 delta=(-b+(sqrt(b^2-(4*a*c))))/(2*a) // m
16 // Results
17 clc
18 printf('The maximum compression of the spring is %f
        m \n',delta)

```

---

**Scilab code Exa 16.11** Principle of conservation of energy

```

1 // Initialization of variables
2 M=10 // kg // Here M=M_1=M_2
3 g=9.81 // m/s^2 // acc due to gravity
4 // Calculations
5 // Consider the respective F.B.D
6 // Applying the principle of conservation of energy
    and by equating the total energies at position 1
    & position 2 we get v as,
7 v=sqrt((M*g*4)/(25)) // m/s
8 // Results
9 clc
10 printf('The velocity of mass M_2 is %f m/s \n',v)

```

---

## Chapter 17

# Kinetics of a Particle Impulse and Momentum

Scilab code Exa 17.1 Principle of impulse and momentum

```
1 // Initialization of variables
2 m=0.1 // kg // mass of ball
3 // Calculations
4 // Consider the respective F.B.D.
5 // For component eq'n in x-direction
6 delta_t=0.015 // seconds // time for which the ball
   &the bat are in contact
7 v_x_1=-25 // m/s
8 v_x_2=40*cosd(40) // m/s
9 F_x_average=((m*(v_x_2))-(m*(v_x_1)))/(delta_t) // N
10 // For component eq'n in y-direction
11 delta_t=0.015 // seconds
12 v_y_1=0 // m/s
13 v_y_2=40*sind(40) // m/s
14 F_y_average=((m*v_y_2)-(m*(v_y_1)))/(delta_t) // N
15 F_average=sqrt(F_x_average^2+F_y_average^2) // N
16 // Results
17 clc
18 printf('The average impules force exerted by the bat
```

on the ball is %f N \n',F\_average)

---

### Scilab code Exa 17.2 Principle of impulse and momentum

```
1 // Initiation of variables
2 m_g=3000 // kg // mass of the gun
3 m_s=50 // kg // mass of the shell
4 v_s=300 // m/s // initial velocity of shell
5 s=0.6 // m // distance at which the gun is brought
   to rest
6 v=0 // m/s // initial velocity of gun
7 // Calculations
8 // On equating eq'n 1 & eq'n 2 we get v_g as ,
9 v_g=(m_s*v_s)/(-m_g) // m/s
10 // Using v^2-u^2=2*a*s to find acceleration ,
11 a=(v^2-v_g^2)/(2*s) // m/s^2
12 // Force required to stop the gun,
13 F=m_g*(-a) // N // here we make a +ve to find the
   Force
14 // Time required to stop the gun, using v=u+a*t:
15 t=(-v_g)/(-a) // seconds // we take -a to consider
   +ve value of acceleration
16 // Results
17 clc
18 printf('The recoil velocity of gun is %f m/s \n',v_g
   )
19 printf('The Force required to stop the gun is %f N \
   n',F)
20 printf('The time required to stop the gun is %f
   seconds \n',t)
```

---

### Scilab code Exa 17.3 Principle of impulse and momentum

```

1 // Initialization of variables
2 m_m=50 // kg // mass of man
3 m_b=250 // kg // mass of boat
4 s=5 // m // length of the boat
5 v_r=1 // m/s // here v_r=v_(m/b)= relative velocity
    of man with respect to boat
6 // Calculations
7 // Velocity of man is given by, v_m=(-v_r)+v_b
8 // Final momentum of the man and the boat=m_m*v_m+
    m_b*v_b. From this eq'n v_b is given as
9 v_b=(m_m*v_r)/(m_m+m_b) // m/s // this is the
    absolute velocity of the boat
10 // Time taken by man to move to the other end of the
    boat is ,
11 t=s/v_r // seconds
12 // The distance travelled by the boat in the same
    time is ,
13 s_b=v_b*t // m to right from O
14 // Results
15 clc
16 printf('(a) The velocity of boat as observed from
    the ground is %f m/s \n',v_b)
17 printf('(b) The distance by which the boat gets
    shifted is %f m \n',s_b)

```

---

#### Scilab code Exa 17.5 Principle of impulse and momentum

```

1 // Initialization of variables
2 M=250 // kg // mass of the boat
3 M_1=50 // kg // mass of the man
4 M_2=75 // kg // mass of the man
5 v=4 // m/s // relative velocity of man w.r.t boat
6 // Calculations
7 // (a)
8 // Let the increase in the velocity or the final

```



```

    velocity of the boat when TWO MEN DIVE
    SIMULTANEOUSLY is given by eq'n,
9  deltaV_1=((M_1+M_2)*v)/(M+(M_1+M_2)) // m/s
10 // (b) // The increase in the velocity or the final
    velocity of the boat when man of 75 kg dives 1st
    followed by man of 50 kg
11 // Man of 75 kg dives first , So let the final
    velocity is given as
12 deltaV_75=(M_2*v)/((M+M_1)+M_2) // m/s
13 // Now let the man of 50 kg jumps next , Here
14 deltaV_50=(M_1*v)/(M+M_1) // m/s
15 // Let final velocity of boat is ,
16 deltaV_2=0+deltaV_75+deltaV_50 // m/s
17 // (c)
18 // The man of 50 kg jumps first ,
19 delV_50=(M_1*v)/((M+M_2)+(M_1)) // m/s
20 // the man of 75 kg jumps next ,
21 delV_75=(M_2*v)/(M+M_2) // m/s
22 // Final velocity of boat is ,
23 deltaV_3=0+delV_50+delV_75 // m/s
24 // Results
25 clc
26 printf('(a) The Final velocity of boat when two men
    dive simultaneously is %f m/s \n',deltaV_1)
27 printf('(b) The Final velocity of boat when the man
    of 75 kg dives first and 50 kg dives second is %f
    m/s \n',deltaV_2)
28 printf('(3) The Final velocity of boat when the man
    of 50kg dives first followed by the man of 75 kg
    is %f m/s \n',deltaV_3)

```

---

### Scilab code Exa 17.6 Principle of impulse and momentum

```

1 // Initalization of variables
2 m_m=70 // kg // mass of man

```

```

3 m_c=35 // kg // mass of canoe
4 m=25/1000 // kg // mass of bullet
5 m_wb=2.25 // kg // mass of wodden block
6 V_b=5 // m/s // velocity of block
7 // Calculations
8 // Considering Initial Momentum of bullet=Final
  momentum of bullet & the block we have, Velocity
  of bullet (v) is given by eq'n,
9 v=(V_b*(m_wb+m))/(m) // m/s
10 // Considering, Momentum of the bullet=Momentum of
  the canoe & the man,the velocity on canoe is
  given by eq'n
11 V=(m*v)/(m_m+m_c) // m/s
12 // Results
13 clc
14 printf('The velocity of the canoe is %f m/s \n',V)

```

---

**Scilab code Exa 17.8** Principle of conservation of angular momentum

```

1 // Initalization of variables
2 m=2 // kg // mass of the particle
3 v_0=20 // m/s // speed of rotation of the mass
  attached to the string
4 r_0=1 // m // radius of the circle along which the
  particle is rotated
5 r_1=r_0/2 // m
6 // Calculations
7 // here, equating (H_0)_1=(H_0)_2 i.e (m*v_0)*r_0=(
  m*v_1)*r_1 (here, r_1=r_0/2). On solving we get
  v_1 as,
8 v_1=2*v_0 // m/s
9 // Tension is given by eq'n,
10 T=(m*v_1^2)/r_1 // N
11 // Results
12 clc

```

```
13 printf('The new speed of the particle is %f m/s \n',  
        v_1)  
14 printf('The tension in the string is %f N \n',T)
```

---

# Chapter 18

## Impact Collision of Elastic Bodies

Scilab code Exa 18.1 Principle of conservation of Momentum

```
1 // Initialization of variables
2 m_a=1 // kg // mass of the ball A
3 v_a=2 // m/s // velocity of ball A
4 m_b=2 // kg // mass of ball B
5 v_b=0 // m/s // ball B at rest
6 e=1/2 // coefficient of restitution
7 // Calculations
8 // Solving eqn's 1 & 2 using matrix for v'_a & v'_b,
9 A=[1 2;-1 1]
10 B=[2;1]
11 C=inv(A)*B
12 // Results
13 clc
14 printf('The velocity of ball A after impact is %f m/
    s \n',C(1))
15 printf('The velocity of ball B after impact is %f m/
    s \n',C(2))
```

---

## Scilab code Exa 18.2 Principle of conservation of Momentum

```
1 // Initialization of variables
2 m_a=2 // kg // mass of ball A
3 m_b=6 // kg // mass of ball B
4 m_c=12 // kg // mass of ball C
5 v_a=12 // m/s // velocity of ball A
6 v_b=4 // m/s // velocity of ball B
7 v_c=2 // m/s // velocity of ball C
8 e=1 // coefficient of restitution for perfectly
    elastic body
9 // Calculations
10 // (A)
11 // Solving eq'n 1 & 2 using matrix for v'_a & v'_b,
12 A=[2 6;-1 1]
13 B=[48;8]
14 C=inv(A)*B
15 // Calculations
16 // (B)
17 // Solving eq'ns 3 & 4 simultaneously using matrix
    for v'_b & v'_c
18 P=[1 2;-1 1]
19 Q=[12;6]
20 R=inv(P)*Q
21 // Results (A&B)
22 clc
23 printf('The velocity of ball A after impact on ball
    B is %f m/s \n',C(1)) // here the ball of mass 2
    kg is bought to rest
24 printf('The velocity of ball B after getting
    impacted by ball A is %f m/s \n',C(2))
25 printf('The final velocity of ball B is %f m/s \n',R
    (1)) // here the ball of mass 6 kg is bought to
    rest
```

```
26 printf('The velocity of ball C after getting
    impacted by ball B is %f m/s \n',R(2))
```

---

### Scilab code Exa 18.3 Principle of conservation of Momentum

```
1 // Initalization of variables
2 h_1=9 // m // height of first bounce
3 h_2=6 // m // height of second bounce
4 // Calculations
5 // From eq'n (5) we have, Coefficient of restitution
    between the glass and the floor is ,
6 e=sqrt(h_2/h_1)
7 // From eq'n 3 we get height of drop as ,
8 h=h_1/e^2 // m
9 // Results
10 clc
11 printf('The ball was dropped from a height of %f m \
    n',h)
12 printf('The coefficient of restitution between the
    glass and the floor is %f \n',e)
13 // Here we use h'='h_1 & h''='h_2 because h' & h''
    could not be defined in Scilab.
```

---

### Scilab code Exa 18.4 Principle of conservation of Momentum

```
1 // Initalization of variables
2 e=0.90 // coefficient o restitution
3 v_a=10 // m/s // velocity of ball A
4 v_b=15 // m/s // velocity of ball B
5 alpha_1=30 // degree // angle made by v_a with
    horizontal
6 alpha_2=60 // degree // angle made by v_b with
    horizontal
```

```

7 // Calculations
8 // The components of initial velocity of ball A:
9 v_a_x=v_a*cosd(alpha_1) // m/s
10 v_a_y=v_a*sind(alpha_1) // m/s
11 // The components of initial velocity of ball B:
12 v_b_x=-v_b*cosd(alpha_2) // m/s
13 v_b_y=v_b*sind(alpha_2) // m/s
14 // From eq'n 1 & 2 we get,
15 v_ay=v_a_y // m/s // Here, v_ay=(v'_a)_y
16 v_by=v_b_y // m/s // Here, v_by=(v'_b)_y
17 // On adding eq'n 3 & 4 we get,
18 v_bx=((v_a_x+v_b_x)+(-e*(v_b_x-v_a_x)))/2 // m/s //
    Here. v_bx=(v'_b)_x
19 // On substituting the value of v'_b_x in eq'n 3 we
    get,
20 v_ax=(v_a_x+v_b_x)-(v_bx) // m/s // here, v_ax=(v'_a
    )_x
21 // Now the eq'n for resultant velocities of balls A
    & B after impact are,
22 v_A=sqrt(v_ax^2+v_ay^2) // m/s
23 v_B=sqrt(v_bx^2+v_by^2) // m/s
24 // The direction of the ball after Impact is,
25 theta_1=atand(-(v_ay/v_ax)) // degree
26 theta_2=atand(v_by/v_bx) // degree
27 // Results
28 clc
29 printf('The velocity of ball A after impact is %f m/
    s \n',v_A)
30 printf('The velocity of ball B after impact is %f m/
    s \n',v_B)
31 printf('The direction of ball A after impact is %f
    degree \n',theta_1)
32 printf('The direction of ball B after impact is %f
    degree \n',theta_2)
33 // Her we use, (1) v'_a & v'_b as v_A & v_B.

```

---

### Scilab code Exa 18.5 Motion of ball

```
1 // Initiization of variables
2 theta=30 // degrees // ange made by the ball against
   the wall
3 e=0.50
4 // Calculations
5 // The notations have been changed
6 // Resolving the velocity v as,
7 v_x=cosd(theta)
8 v_y=sind(theta)
9 V_y=v_y
10 // from coefficient of restitution reation
11 V_x=-e*v_x
12 // Resultant velocity
13 V=sqrt(V_x^2+V_y^2)
14 theta=atand(V_y/(-V_x)) // taking +ve value for V_x
15 // NOTE: Here all the terms are multiplied with
   velocity i.e (v).
16 // Results
17 clc
18 printf('The velocity of the ball is %f v \n',V)
19 printf('The direction of the ball is %f degrees \n',
   theta)
```

---

### Scilab code Exa 18.6 Principle of conservation of Energy

```
1 // Initalization of variables
2 e=0.8 // coefficient of restitution
3 g=9.81 // m/s^2 // acc due to gravity
4 // Calcuations
```



```

5 // Squaring eqn's 1 &2 and Solving eqn's 1 & 2 using
   matrix for the value of h
6 A=[-1 (2*g);-1 -(1.28*g)]
7 B=[0.945^2;(-0.4*9.81)]
8 C=inv(A)*B // m
9 // Results
10 clc
11 printf('The height from which the ball A should be
   released is %f m \n',C(2))
12 // The answer given in the book i.e 0.104 is wrong.

```

---

**Scilab code Exa 18.7** Principle of conservation of Energy

```

1 // Initalization of variables
2 theta_a=60 // degree // angle made by sphere A with
   the verticle
3 e=1 // coefficient of restitution for elastic impact
4 // Calculations
5 // theta_b is given by the eq'n cosd*theta_b=0.875,
   hence theta_b is ,
6 theta_b=acosd(0.875) // degree
7 // Results
8 clc
9 printf('The angle through which the sphere B will
   swing after the impact is %f degree \n',theta_b)

```

---

**Scilab code Exa 18.8** Principle of conservation of Energy

```

1 // Initalization of variables
2 m_a=0.01 // kg // mass of bullet A
3 v_a=100 // m/s // velocity of bullet A
4 m_b=1 // kg // mass of the bob
5 v_b=0 // m/s // velocity of the bob

```

```

6 l=1 // m // length of the pendulum
7 v_r=-20 // m/s // velocity at which the bullet
  rebounds the surface of the bob // here the
  notation for v'_a is shown by v_r
8 v_e=20 // m/s // velocity at which the bullet
  escapes through the surface of the bob // here
  the notation for v_a is shown by v_e
9 g=9.81 // m/s^2 // acc due to gravity
10 // Calculations
11 // Momentum of the bullet & the bob before impact is
    ,
12 M=(m_a*v_a)+(m_b*v_b) // kg.m/s .....(eq'n 1)
13 // The common velocity v_c ( we use v_c insted of v'
  for notation of common velocity) is given by
  equating eq'n 1 & eq'n 2 as ,
14 // (a) When the bullet gets embedded into the bob
15 v_c=M/(m_a+m_b) // m/s
16 // The height h to which the bob rises is given by
  eq'n 3 as ,
17 h_1=(1/2)*(v_c^2/g) // m
18 // The angle (theta_1) by which the bob swings
  corresponding to the value of height h_1 is ,
19 theta_1=acosd((1-h_1)/l) // degree
20 // (b) When the bullet rebounds from the surface of
  the bob
21 // The velocity of the bob after the rebound of the
  bullet from its surface is given by equating eq'n
  1 & eq'n 4 as ,
22 v_bob_rebound=M-(m_a*v_r) // m/s // here
  v_bob_rebound=v'_b
23 // The equation for the height which the bob attains
  after impact is ,
24 h_2=(v_bob_rebound^2)/(2*g) // m
25 // The corresponding angle of swing
26 theta_2=acosd((1-h_2)/l) // degree
27 // (c) When the bullet pierces and escapes through
  the bob
28 // From eq'n 1 & 5 the velocity attained by the bob

```

```

    after impact is given as,
29 v_b_escape=M-(m_a*v_e) // m/s // here we use,
    v_b_escape insted of v'_b
30 // The equation for the height which the bob attains
    after impact is,
31 h_3=(v_b_escape^2)/(2*g) // m
32 // The corresponding angle of swing
33 theta_3=acosd((1-h_3)/(1)) // degree
34 // Results
35 clc
36 printf('(a) The maximum angle through which the
    pendulum swings when the bullet gets embedded into
    the bob is %f degree \n',theta_1)
37 printf('(b) The maximum angle through which the
    pendulum swings when the bullet rebounds from the
    surface of the bob is %f degree \n',theta_2)
38 printf('(c) The maximum angle through which the
    pendulum swings when the bullet escapes from
    other end of the bob the bob is %f degree \n',
    theta_3)
39 // IN THIS SUM WE HAVE USED DIFFERENT NOTATIONS
    CONSIDERING DIFFERENT CASES BECAUSE IN THE TEXT
    BOOK WE HAD 3 VARIABLES WITH SAME NOTATION BUT
    WITH A DIFFERENT VALUE WHICH COULD NOT BE
    EXECUTED INTO SCILAB.

```

---

### Scilab code Exa 18.9 Principle of conservation of Momentum

```

1 // Initalization of variables
2 W_a=50 // N // falling weight
3 W_b=50 // N // weight on which W_a falls
4 g=9.81 // m/s^2 // acc due to gravity
5 m_a=W_a/g // kg // mass of W_a
6 m_b=W_b/g // kg // mass of W_b
7 k=2*10^3 // N/m // stiffness of spring

```

```

8 h=0.075 // m // height through which W_a falls
9 // The velocity of weight W_a just before the impact
  and after falling from a height of h is given
  from the eq'n, ( Principle of conservation of
  energy)
10 v_a=sqrt(2*g*h) // m/s
11 // Let the mutual velocity after the impact be v_m (
  i.e v_m=v'), (by principle of conservation of
  momentum)
12 v_m=(m_a*v_a)/(m_a+m_b) // m/s
13 // Initial compression of the spring due to weight
  W_b is given by,
14 delta_st=(W_b/k)*(10^2) // cm
15 // Let the total compression of the spring be
  delta_t, Then delta_t is found by finding the
  roots from the eq'n..... delta_t^2-0.1*delta_t
  -0.000003=0. In this eq'n let ,
16 a=1
17 b=-0.1
18 c=-0.000003
19 delta_t=(-b+(sqrt(b^2-(4*a*c))))/2*a)*(10^2) // cm
  // we consider the -ve value
20 delta=delta_t-delta_st // cm
21 // Results
22 clc
23 printf('The compression of the spring over and above
  caused by the static action of weight W_a is %f
  cm \n',delta)

```

---

#### Scilab code Exa 18.10 Principle of conservation of Momentum

```

1 // Initilization of variables
2 v_a=600 // m/s // velocity of the bullet before
  impact
3 v_b=0 // m/s // velocity of the block before impact

```

```

4 w_b=0.25 // N // weight of the bullet
5 w_wb=50 // N // weight of wodden block
6 mu=0.5 // coefficient of friction between the floor
    and the block
7 g=9.81 // m/s^2 // acc due to gravity
8 // Calculations
9 m_a=w_b/g // kg // mass of the bullet
10 m_b=w_wb/g // kg // mass of the block
11 // Let the common velocity be v_c which is given by
    eq'n (Principle of conservation of momentum)
12 v_c=(w_b*v_a)/(w_wb+w_b) // m/s
13 // Let the distance through which the block is
    displaced be s, Then s is given by eq'n
14 s=v_c^2/(2*g*mu) // m
15 // Results
16 clc
17 printf('The distance through which the block is
    displaced from its initial position is %f m \n',s
    )

```

---

**Scilab code Exa 18.11** Principle of conservation of Energy Momentum and work and en

```

1 // Initalization of variables
2 M=750 // kg // mass of hammer
3 m=200 // kg // mass of the pile
4 h=1.2 // m // height of fall of the hammer
5 delta=0.1 // m // distance upto which the pile is
    driven into the ground
6 g=9.81 // m/s^2 // acc due to gravity
7 // Caculations
8 // The resistance to penetration to the pile is
    given by eq'n,
9 R=((M+m)*g)+((M^2*g*h)/((M+m)*delta))*(10^-3) //
    kN
10 // Results

```

```
11 clc
12 printf('The resistance to penetration to the pile is
    %f kN \n',R)
```

---

# Chapter 19

## Relative Motion

Scilab code Exa 19.1 Relative Velocity

```
1 // Initialization of variables
2 v_t=10 // m/s // velocity of the train
3 v_s=5 // m/s // velocity of the stone
4 // Calculations
5 // Let v_r be the relative velocity, which is given
   as, (from triangle law)
6 v_r=sqrt(v_t^2+v_s^2) // m/s
7 // The direction of the stone is,
8 theta=atand(v_s/v_t) // degree
9 // Results
10 clc
11 printf('The velocity at which the stone appears to
   hit the person travelling in the train is %f m/s
   \n',v_r)
12 printf('The direction of the stone is %f degree \n',
   theta)
```

---

Scilab code Exa 19.2 Relative Velocity

```

1 // Initalization of variables
2 v_A=5 // m/s // speed of ship A
3 v_B=2.5 // m/s // speed of ship B
4 theta=135 // degree // angle between the two ships
5 // Calculations
6 // Here,
7 OA=v_A // m/s
8 OB=v_B // m/s
9 // The magnitude of relative velocity is given by
   cosine law as,
10 AB=sqrt((OA^2)+(OB^2)-(2*OA*OB*cosd(theta))) // m/s
11 // where AB gives the relative velocity of ship B
   with respect to ship A
12 // Applying sine law to find the direction, Let
   alpha be the direction of the reative velocity,
   then
13 alpha=asind((OB*sind(theta))/(AB)) // degree
14 // Results
15 clc
16 printf('The magnitude of relative velocity of ship B
   with respect to ship A is %f m/s \n',AB)
17 printf('The direction of the relative velocity is %f
   degree \n',alpha)

```

---

### Scilab code Exa 19.3 Relative Velocity

```

1 // Initalization of variables
2 v_c=20 // km/hr // speed at which the cyclist is
   riding to west
3 theta_1=45 // degree // angle made by rain with the
   cyclist when he rides at 20 km/hr
4 V_c=12 // km/hr // changed speed
5 theta_2=30 // degree // changed angle when the
   cyclist rides at 12 km/hr
6 // Calculations

```



```

7 // Solving eq'ns 1 & 2 simultaneously to get the
  values of components(v_R_x & v_R_y) of absolute
  velocity v_R. We use matrix to solve eqn's 1 & 2.
8 A=[1 1;1 0.577]
9 B=[20;12]
10 C=inv(A)*B // km/hr
11 // The X component of relative velocity (v_R_x) is C
  (1)
12 // The Y component of relative velocity (v_R_y) is C
  (2)
13 // Calculations
14 // Relative velocity (v_R) is given as,
15 v_R=sqrt((C(1))^2+(C(2))^2) // km/hr
16 // And the direction of absolute velocity of rain is
  theta, is given as
17 theta=atand(C(2)/C(1)) // degree
18 // Results
19 clc
20 printf('The magnitude of absolute velocity is %f km/
  hr \n',v_R)
21 printf('The direction of absolute velocity is %f
  degree \n',theta)

```

---

#### Scilab code Exa 19.4 Relative Velocity

```

1 // Initiization of variables
2 a=1 // m/s^2 // acceleration of car A
3 u_B=36*(1000/3600) // m/s // velocity of car B
4 u=0 // m/s // initial velocity of car A
5 d=32.5 // m // position of car A from north of
  crossing
6 t=5 // seconds
7 // Calculations
8 // CAR A: Absolute motion using eq'n v=u+at we have,
9 v=u+(a*t) // m/s

```

```

10 // Now distance travelled by car A after 5 seconds
    is given by,  $s_A = u*t + (1/2)*a*t^2$ 
11  $s_A = (u*t) + ((1/2)*a*t^2)$ 
12 // Now, let the position of car A after 5 seconds be
     $y_A$ 
13  $y_A = d - s_A$  // m //
14 // CAR B:
15 // let  $a_B$  be the acceleration of car B
16  $a_B = 0$  // m/s
17 // Now position of car B is  $s_B$ 
18  $s_B = (u_B*t) + ((1/2)*a_B*t^2)$  // m
19  $x_B = s_B$  // m
20 // Let the Relative position of car A with respect
    to car B be BA & its direction be theta, then
    from fig. 19.9(b)
21  $OA = y_A$ 
22  $OB = x_B$ 
23  $BA = \text{sqrt}(OA^2 + OB^2)$  // m
24  $\text{theta} = \text{atand}(OA/OB)$  // degree
25 // Let the relative velocity of car A w.r.t. the car
    B be  $v_{AB}$  & the angle be phi. Then from fig
    19.9(c). Consider small alphabets
26  $oa = v$ 
27  $ob = u_B$ 
28  $v_{AB} = \text{sqrt}(oa^2 + ob^2)$  // m/s
29  $\text{phi} = \text{atand}(oa/ob)$  // degree
30 // Let the relative acceleration of car A w.r.t. car
    B be  $a_{A/B}$ . Then,
31  $a_{AB} = a - a_B$  //  $\text{m/s}^2$ 
32 // Results
33 clc
34 printf('The relative position of car A relative to
    car B is %f m \n',BA)
35 printf('The direction of car A w.r.t car B is %f
    degree \n',theta)
36 printf('The velocity of car A relative to car B is
    %f m/s \n',v_AB)
37 printf('The direction of car A w.r.t (for relative

```

```
    velocity)is %f degree \n',phi)
38 printf('The acceleration of car A relative to car B
    is %f m/s^2 \n',a_AB)
```

---

# Chapter 20

## Motion of Projectile

Scilab code Exa 20.1 Motion of Projectile

```
1 // Initialization of variables
2 v_o=500 // m/s // velocity of the projectile
3 alpha=30 // angle at which the projectile is fired
4 t=30 // seconds
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 v_x=v_o*cosd(alpha) // m/s // Initial velocity in
   the horizontal direction
8 v_y=v_o*sind(alpha) // m/s // Initial velocity in
   the vertical direction
9 // MOTION IN HORIZONTAL DIRECTION:
10 V_x=v_x // m/s // V_x=Horizontal velocity after 30
   seconds
11 // MOTION IN VERTICAL DIRECTION: // using the eq'n v
   =u+a*t
12 V_y=v_y-(g*t) // m/s // -ve sign denotes downward
   motion
13 // Let the Resultant velocity be v_R. It is given as
   ,
14 v_R=sqrt((V_x)^2+(-V_y)^2) // m/s
15 theta=atand((-V_y)/V_x) // degree // direction of
```

```

        the projectile
16 // Results
17 clc
18 printf('The velocity of the projectile is %f m/s \n'
        ,v_R) // The answer of velocity is wrong in the
        text book.
19 printf('The direction of the projectile is %f degree
        \n',theta) // -ve value of theta indicates that
        the direction is in downward direction

```

---

### Scilab code Exa 20.2 Motion of Projectile

```

1 // Initialization of variables
2 v_A=10 // m/s // velocity of body A
3 alpha_A=60 // degree // direction of body A
4 alpha_B=45 // degree // direction of body B
5 // Calculations
6 // (a) The velocity (v_B) for the same range is
        given by eq'n;
7 v_B=sqrt((v_A^2*sind(2*alpha_A))/(sind(2*alpha_B)))
        // m/s
8 // (b) Now velocity v_B for the same maximum height
        is given as,
9 v_b=sqrt((v_A^2)*((sind(alpha_A))^2/(sind(alpha_B))
        ^2)) // m/s
10 // (c) Now the velocity (v) for the equal time of
        flight is;
11 v=(v_A*sind(alpha_A))/(sind(alpha_B)) // m/s
12 // Results
13 clc
14 printf('(a) The velocity of body B for horizontal
        range is %f m/s \n',v_B)
15 printf('(b) The velocity of body B for the maximum
        height is %f m/s \n',v_b)
16 printf('(c) The velocity of body B for equal time of

```

flight is %f m/s \n',v)

---

### Scilab code Exa 20.3 Motion of Projectile

```
1 // Initialization of variables
2 y=3.6 // m // height of the wall
3 x_1=4.8 // m // position of the boy w.r.t the wall
4 x_2=3.6 // m // distance from the wall where the
   ball hits the ground
5 g=9.81 // m/s^2 // acc due to gravity
6 // Calculations
7 // The range of the projectile is r, given as,
8 r=x_1+x_2 // m
9 // Let the angle of the projection be alpha, which
   is derived and given as,
10 alpha=atand((y)/(x_1-(x_1^2/r))) // degree
11 // Now substituting the value of alpha in eq'n 3 we
   get the least velocity (v_o) as;
12 v_o=sqrt((g*r)/(sind(2*alpha))) // m/s
13 // Results
14 clc
15 printf('The least velocity with which the ball can
   be thrown is %f m/s \n',v_o)
16 printf('The angle of projection for the same is %f
   degree \n',alpha)
```

---

### Scilab code Exa 20.5 Motion of Projectile

```
1 // Initialization of variables
2 v_o=400 // m/s // initial velocity of each gun
3 r=5000 // m // range of each of the guns
4 g=9.81 // m/s^2 // acc due to gravity
5 pi=180 // degree
```

```

6 // Calculations
7 // now from eq'n 1
8 theta_1=(asind((r*g)/(v_o^2)))/(2) // degree //
    angle at which the 1st gun is fired
9 // from eq'n 3
10 theta_2=(pi-(2*theta_1))/2 // degree
11 // For 1st & 2nd gun, s is
12 s=r // m
13 // For 1st gun
14 v_x=v_o*cosd(theta_1) // m/s
15 // Now the time of flight for 1st gun is t_1, which
    is given by relation,
16 t_1=s/(v_x) // seconds
17 // For 2nd gun
18 V_x=v_o*cosd(theta_2)
19 // Now the time of flight for 2nd gun is t_2
20 t_2=s/(V_x) // seconds
21 // Let the time difference between the two hits be
    delta.T. Then,
22 delta.T=t_2-t_1 // seconds
23 // Results
24 clc
25 printf('The time difference between the two hits is
    %f seconds \n',delta.T)

```

---

### Scilab code Exa 20.6 Motion of Projectile

```

1 // Initialization of variables
2 h=2000 // m/ height of the plane
3 v=540*(1000/3600) // m/s // velocity of the plane
4 g=9.81 // m/s^2 // acc due to gravity
5 // Calculations
6 // Time t required to travel down a height 2000 m is
    given by eq'n,
7 u=0 // m/s // initial velocity

```

```

8 t=sqrt((2*h)/(g)) // seconds
9 // Now let s be the horizontal distance travelled by
  the bomb in time t seconds, then
10 s=v*t // m
11 // angle is given as theta,
12 theta=atand(h/s) // degree
13 // Results
14 clc
15 printf('The pilot should release the bomb from a
  distance of %f m \n',s)
16 printf('The angle at which the target would appear
  is %f degree \n',theta)

```

---

#### Scilab code Exa 20.7 Motion of Projectile

```

1 // Initialization of variables
2 theta=30 // degree // angle at which the bullet is
  fired
3 s=-50 // position of target below hill
4 v=100 // m/s // velocity at which the bullet is
  fired
5 g=9.81 // m/s^2
6 // Calculations
7 v_x=v*cosd(theta) // m/s // Initial velocity in
  horizontal direction
8 v_y=v*sind(theta) // m/s // Initial velocity in
  vertical direction
9 // (a) Max height attained by the bullet
10 h=v_y^2/(2*g) // m
11 // (b) Let the vertical Velocity with which the
  bullet will hit the target be V_y. Then,
12 V_y=sqrt((2*-9.81*s)+(v_y)^2) // m/s // the value of
  V_y is +ve & -ve
13 // Let V be the velocity with which it hits the
  target

```



```

14 V=sqrt((v_x)^2+(V_y)^2) // m/s
15 // (c) The time required to hit the target
16 a=g // m/s^2
17 t=(v_y-(-V_y))/a // seconds
18 // Results
19 clc
20 printf('(a) The maximum height to which the bullet
    will rise above the soldier is %f m \n',h)
21 printf('(b) The velocity with which the bullet will
    hit the target is %f m/s \n',V)
22 printf('(c) The time required to hit the target is
    %f seconds \n',t)

```

---

#### Scilab code Exa 20.8 Motion of Projectile

```

1 // Initialization of variables
2 W=30 // N // Weight of the hammer
3 theta=30 // degree // ref fig.20.12
4 mu=0.18 // coefficient of friction
5 s=10 // m // distance travelled by the hammer // fig
    20.12
6 g=9.81 // m/s^2 // acc due to gravity
7 // Calculations
8 // The acceleration of the hammer is given as,
9 a=g*((sind(theta))-(mu*cosd(theta))) // m/s^2
10 // The velocity of the hammer at point B is,
11 v=sqrt(2*a*s) // m/s
12 // Let the initial velocity of the hammer in
    horizontal direction be v_x & v_y in vertical
    direction, Then,
13 v_x=v*cosd(theta) // m/s
14 v_y=v*sind(theta) // m/s
15 // MOTION IN VERTICAL DIRECTION
16 // Now, let time required to travel vertical
    distance (i.e BB'=S=5 m) is given by finding the

```

```

    roots of the second degree eq'n as ,
17 // From the eq'n  $4.9*t^2+4.1*t-5=0$ ,
18 a=4.9
19 b=4.1
20 c=-5
21 // The roots of the eq'n are ,
22 t=((-b)+(sqrt(b^2-(4*a*c))))/(2*a)
23 // MOTION IN HORIZONTAL DIRECTION
24 // Let the horizontal distance travelled by the
    hammer in time t be s_x.Then,
25 s_x=v_x*cosd(theta)*t // m
26 x=1+s_x // m
27 // Results
28 clc
29 printf('The distance x where the hammer hits the
    round is %f m \n',x)

```

---

### Scilab code Exa 20.9 Motion of Projectile

```

1 // Initialization of variables
2 s=1000 // m // distance OB (ref fig.20.13)
3 h=19.6 // m // height of shell from ground
4 g=9.81 // m/s^2 // acc due to gravity
5 // Calculations
6 // MOTION OF ENTIRE SHELL FROM O to A.
7 v_y=sqrt(2*(g)*h) // m/s // initial velocity of
    shell in vertical direction
8 t=v_y/g // seconds // time taken by the entire shell
    to reach point A
9 v_x=s/t // m/s // velocity of shell in vertical
    direction
10 // VELOCITIES OF THE TWO PARTS OF THE SHELL AFTER
    BURSTING AT A:
11 // Let v_x2 be the horizontal velocity of 1st & the
    2nd part after bursting which is given as ,

```

```

12 v_x2=v_x*2 // m/s
13 // Now distance BC travelled by part 2 is
14 BC=v_x2*t // m
15 // Distance from firing point OC
16 OC=s+BC // m
17 // Results
18 clc
19 printf('(a) The velocity of shell just before
    bursting is %f m/s \n',v_x)
20 printf('(b) The velocity of first part immediately
    after the shell burst is %f m/s \n',v_x2)
21 printf('(c) The velocity of second part immediately
    after the shell burst is %f m/s \n',v_x2)
22 printf('(b) The distance between the firing point &
    the point where the second part of the shell hit
    the ground is %f m \n',OC)

```

---

#### Scilab code Exa 20.10 Motion of Projectile

```

1 // Initialization of variables
2 v_o=200 // m/s // initial velocity
3 theta=60 // degree // angle of the incline
4 y=5 // rise of incline
5 x=12 // length of incline
6 g=9.81 // m/s^2 // acc due to gravity
7 // Calculations
8 // The angle of the inclined plane with respect to
    horizontal
9 beta=atand(y/x) // degree
10 // The angle of projection with respect to
    horizontal
11 alpha=90-theta // degree
12 // Range is given by eq'n (ref. fig.20.14)
13 AB=(2*v_o^2*(sind(alpha-beta))*cosd(alpha))/(g*(cosd
    (beta))^2) // m

```

```
14 // Range AC when the short is fired down the plane
15 AC=(2*v_o^2*(sind(alpha+beta))*cosd(alpha))/(g*(cosd
    (beta))^2) // m
16 BC=AB+AC // m
17 // Results
18 clc
19 printf('The range covered (i.e BC) is %f m \n',BC)
```

---

# Chapter 21

## Kinematics of rigid body

Scilab code Exa 21.1 Linear and angular velocity linear and angular acceleration i

```
1 // Initialization of variables
2 N=1800 // r.p.m // Speed of the shaft
3 t=5 // seconds // time taken to attain the rated
   speed // case (a)
4 T=90 // seconds // time taken by the unit to come to
   rest // case (b)
5 // Calculations
6 omega=(2*%pi*N)/(60)
7 // (a)
8 // we take alpha_1, theta_1 & n_1 for case (a)
9 alpha_1=omega/t // rad/s^2 //
10 theta_1=(omega^2)/(2*alpha_1) // radian
11 // Let n_1 be the number of revolutions turned,
12 n_1=theta_1*(1/(2*%pi))
13 // (b)
14 // similarly we take alpha_1, theta_1 & n_1 for case
   (b)
15 alpha_2=(omega/T) // rad/s^2 // However here alpha_2
   is -ve
16 theta_2=(omega^2)/(2*alpha_2) // radians
17 // Let n_2 be the number of revolutions turned,
```

```

18 n_2=theta_2*(1/(2*pi))
19 // Results
20 clc
21 printf('(a) The no of revolutions the unit turns to
    attain the rated speed is %f \n',n_1)
22 printf('(b) The no of revolutions the unit turns to
    come to rest is %f \n',n_2)

```

---

**Scilab code Exa 21.2** Absolute and relative velocity in plane motion

```

1 // Initialization of variables
2 r=1 // m // radius of the cylinder
3 v_c=20 // m/s // velocity of the cylinder at its
    centre
4 // Calculations
5 // The velocity of point E is given by using the
    triangle law as,
6 v_e=sqrt(2)*v_c // m/s
7 // Similarly the velocity at point F is given as,
8 v_f=2*v_c // m/s
9 // Results
10 clc
11 printf('The velocity of point E is %f m/s \n',v_e)
12 printf('The velocity of point F is %f m/s \n',v_f)

```

---

**Scilab code Exa 21.3** Absolute and relative velocity in plane motion

```

1 // Initialization of Variables
2 v_1=3 // m/s // uniform speed of the belt at top
3 v_2=2 // m/s // uniform speed of the belt at the
    bottom
4 r=0.4 // m // radius of the roller
5 // Calculations

```

```

6 // equating eq'ns 2 & 4 and solving for v_c & theta'
   (angular velocity). We use matrix to solve the
   eqn's
7 A=[1 r;1 -r]
8 B=[v_1;v_2]
9 C=inv(A)*B
10 // Results
11 clc
12 printf('The linear velocity (v_c) at point C is %f m
   /s \n',C(1))
13 printf('The angular velocity at point C is %f radian
   /seconds \n', C(2))
14 // NOTE: The answer of angular velocity is incorrect
   in the book

```

---

**Scilab code Exa 21.4 Absolute and relative velocity in plane motion**

```

1 // Initialization of Variables
2 l=1 // m // length of bar AB
3 v_a=5 // m/s // velocity of A
4 theta=30 // degree // angle made by the bar with the
   horizontal
5 // Calculations
6 // From the vector diagram linear velocity of end B
   is given as,
7 v_b=v_a/tand(theta) // m/s
8 // Now let the relative velocity be v_ba which is
   given as,
9 v_ba=v_a/sind(theta) // m/s
10 // Now let the angular velocity of the bar be
   theta_a which is given as,
11 theta_a=(v_ba)/l // radian/second
12 // Velocity of point A
13 v_a=(1/2)*theta_a // m/s
14 // Magnitude of velocity at point C is,

```

```

15 v_c=v_a // m/s // from the vector diagram
16 // Results
17 clc
18 printf('(a) The angular velocity of the bar is %f
        radian/second \n',theta_a)
19 printf('(b) The velocity of end B is %f m/s \n',v_b)
20 printf('(c) The velocity of mid point C is %f m/s \n
        ',v_c)

```

---

### Scilab code Exa 21.5 Absolute and relative velocity in plane motion

```

1 // Initialization of Variables
2 r=0.12 // m // length of the crank
3 l=0.6 // m // length of the connecting rod
4 N=300 // r.p.m // angular velocity of the crank
5 theta=30 // degree // angle made by the crank with
        the horizontal
6 // Calculations
7 // Now let the angle between the connecting rod and
        the horizontal rod be phi
8 phi=asind((r*sind(theta))/(l)) // degree
9 // Now let the angular velocity of crank OA be
        omega_oa, which is given by eq'n
10 omega_oa=(2*%pi*N)/(60) // radian/second
11 // Linear velocity at A is given as,
12 v_a=r*omega_oa // m/s
13 // Now using the sine rule linear velocity at B can
        be given as,
14 v_b=(v_a*sind(35.7))/(sind(84.3)) // m/s
15 // Similarly the relative velocity (assume v_ba) is
        given as,
16 v_ba=(v_a*sind(60))/(sind(84.3))
17 // Angular velocity (omega_ab) is given as,
18 omega_ab=v_ba/l // radian/second
19 // Results

```



```

20 clc
21 printf('(a) The angular velocity of the connecting
    rod is %f radian/second \n',omega_ab)
22 printf('(b) The velocity of the piston when the
    crank makes an angle of 30 degree is %f m/s \n',
    v_b)

```

---

**Scilab code Exa 21.6** Instantaneous Centre of rotation in plane motion

```

1 // Initiization of variables
2 r=1 // m // radius of the cylinder
3 v_c=20 // m/s // velocity at the centre
4 // Calculations
5 // Angular velocity is given as ,
6 omega=v_c/r // radian/second
7 // Velocity at point D is
8 v_d=omega*sqrt(2)*r // m/s // from eq'n 1
9 // Now, the velocity at point E is ,
10 v_e=omega*2*r // m/s
11 // Results
12 clc
13 printf('The velocity at point D is %f m/s \n',v_d)
14 printf('The velocity at point E is %f m/s \n',v_e)

```

---

**Scilab code Exa 21.7** Instantaneous Centre of rotation in plane motion

```

1 // Initalization of Variables
2 r=5 // cm // radius of the roller
3 AB=0.1 // m
4 v_a=3 // m/s // velocity at A
5 v_b=2 // m/s // velocity at B
6 // Calculations

```

```

7 // Solving eqn's 1 & 2 using matrix for IA & IB we
  get ,
8 A=[-2 3;1 1]
9 B=[0;AB]
10 C=inv(A)*B
11 d1=C(2)*10^2 // cm // assume d1 for case 1
12 // Similary solving eqn's 3 & 4 again for IA & IB we
  get ,
13 P=[-v_b v_a;1 -1]
14 Q=[0;AB]
15 R=inv(P)*Q
16 d2=R(2)*10^2 // cm // assume d2 for case 2
17 // Results
18 clc
19 printf('The distance d when the bars move in the
  opposite directions are %f cm \n',d1)
20 printf('The distance d when the bars move in the
  same directions are %f cm \n',d2)

```

---

**Scilab code Exa 21.8** Instantaneous Centre of rotation in plane motion

```

1 // Initalization of Variables
2 v_c=1 // m/s // velocity t the centre
3 r1=0.1 // m
4 r2=0.20 // m
5 EB=0.1 // m
6 EA=0.3 // m
7 ED=sqrt(r1^2+r2^2) // m
8 // Calculations
9 // angular velocity is given as ,
10 omega=v_c/r1 // radian/seconds
11 // Velocit at point B
12 v_b=omega*EB // m/s
13 // Velocity at point A
14 v_a=omega*EA // m/s

```

```

15 // Velocity at point D
16 v_d=omega*ED // m/s
17 // Results
18 clc
19 printf('The velocity at point A is %f m/s \n',v_a)
20 printf('The velocity at point B is %f m/s \n',v_b)
21 printf('The velocity at point D is %f m/s \n',v_d)

```

---

**Scilab code Exa 21.9** Instantaneous Centre of rotation in plane motion

```

1 // Initalization of variables
2 l=1 // m // length of bar AB
3 v_a=5 // m/s // velocity at A
4 theta=30 // degree // angle made by the bar with the
   horizontal
5 // Calculations
6 IA=l*sind(theta) // m
7 IB=l*cosd(theta) // m
8 IC=0.5 // m // from triangle IAC
9 // Angular veocity is given as,
10 omega=v_a/(IA) // radian/second
11 v_b=omega*IB // m/s
12 v_c=omega*IC // m/s
13 // Results
14 clc
15 printf('The velocity at point B is %f m/s \n',v_b)
16 printf('The velocity at point C is %f m/s \n',v_c)

```

---

**Scilab code Exa 21.11** Instantaneous Centre of rotation in plane motion

```

1 // Initalization of variables
2 v_a=2 // m/s // velocity at end A
3 r=0.05 // m // radius of the disc

```

```

4 alpha=30 // degree // angle made by the bar with the
    horizontal
5 // Calculations
6 // Solving eqn's 1 & 2 and substituting eqn 1 in it we
    get eq'n for omega as,
7 omega=(v_a*(sind(alpha))^2)/(r*cosd(alpha)) //
    radian/second
8 // Results
9 clc
10 printf('The angular velocity of the bar is %f radian/
    second \n',omega)

```

---

**Scilab code Exa 21.12** Instantaneous Centre of rotation in plane motion

```

1 // Initialization of variables
2 l=0.6 // m
3 r=0.12 // m
4 theta=30 // degree // angle made by OA with the
    horizontal
5 phi=5.7 // degree // from EX 21.5
6 N=300
7 // Calculations
8 // Let the angular velocity of the connecting rod be
    (omega_ab) which is given from eqn's 1 & 4 as,
9 omega_oa=(2*pi*N)/(60) // radian/ second
10 // Now, in triangle IBO.
11 IB=(l*cosd(phi)*tand(theta))+(r*sind(theta)) // m
12 IA=(l*cosd(phi))/(cosd(theta)) // m
13 // from eq'n 5
14 v_b=(r*omega_oa*IB)/(IA) // m/s
15 // From eq'n 6
16 omega_ab=(r*omega_oa)/(IA) // radian/second
17 // Results
18 clc
19 printf('The velocity at B is %f m/s \n',v_b)

```

```
20 printf('The angular velocity of the connecting rod
    is %f radian/second \n',omega_ab)
```

---

### Scilab code Exa 21.13 Instantaneous Centre of rotation in plane motion

```
1 // Initilization of variables
2 omega_ab=5 // rad/s // angular veocity of the bar
3 AB=0.20 // m
4 BC=0.15 // m
5 CD=0.3 // m
6 theta=30 // degree // where theta= angle made by AB
    with the horizontal
7 alpha=60 // degree // where alpha=angle made by CD
    with the horizontal
8 // Calculations
9 // Consider triangle BIC
10 IB=sind(alpha)*BC*1 // m
11 IC=sind(theta)*BC*1 // m
12 v_b=omega_ab*AB // m/s
13 // let the angular velocity of the bar BC be
    omega_bc
14 omega_bc=v_b/IB // radian/second
15 v_c=omega_bc*IC // m/s
16 // let the angular velocity of bar DC be omega_dc
17 omega_dc=v_c/CD // radian/second
18 // Results
19 clc
20 printf('The angular velocity of bar BC is %f rad/s \
    n',omega_bc)
21 printf('The angular velocity of bar CD is %f rad/s \
    n',omega_dc)
```

---

## Chapter 22

# Kinetics of Rigid Body Force and Acceleration

Scilab code Exa 22.1 Relation between the translatory motion and rotary motion of

```
1 // Initialization of variables
2 N=1500 // r.p.m
3 r=0.5 // m // radius of the disc
4 m=300 // N // weight of the disc
5 t=120 // seconds // time in which the disc comes to
   rest
6 omega=0
7 g=9.81 // m/s^2
8 // Calculations
9 omega_0=(2*%pi*N)/60 // rad/s
10 // angular deceleration is given as,
11 alpha=-(omega_0/t) // radian/second^2
12 theta=(omega_0^2)/(2*(-alpha)) // radian
13 // Let n be the no of revolutions taken by the disc
   before it comes to rest, then
14 n=theta/(2*%pi)
15 // Now,
16 I_G=((1/2)*m*r^2)/g
17 // The frictional torque is given as,
```

```

18 M=I_G*alpha // N-m
19 // Results
20 clc
21 printf('(a) The no of revolutions executed by the
    disc before coming to rest is %f \n',n)
22 printf('(b) The frictional torque is %f N-m \n',M)

```

---

**Scilab code Exa 22.2** Relation between the translatory motion and rotary motion of

```

1 // Initialization of variables
2 s=1 // m
3 mu=0.192 // coefficient of static friction
4 g=9.81 // m/s^2
5 // Calculations
6 // The maximum angle of the inclined plane is given
    as,
7 theta=atand(3*mu) // degree
8 a=(2/3)*g*sind(theta) // m/s^2 // by solving eq'n 4
9 v=sqrt(2*a*s) // m/s
10 // Let the acceleration at the centre be A which is
    given as,
11 A=g*sind(theta) // m/s^2 // from eq'n 1
12 // Results
13 clc
14 printf('(a) The acceleration at the centre is %f m/s
    ^2 \n',A)
15 printf('(b) The maximum angle of the inclined plane
    is %f degree \n',theta)

```

---

**Scilab code Exa 22.5** Relation between the translatory motion and rotary motion of

```

1 // Initialization of variables
2 W_a=25 // N

```

```

3 W_b=25 // N
4 W=200 // N // weight of the pulley
5 i_g=0.2 // m // radius of gyration
6 g=9.81 // m/s^2
7 // Calculations
8 // Solving eqn's 1 & 2 for acceleration of weight A
  (assume a)
9 a=(0.15*W_a*g)/(((W*i_g^2)/(0.45))+0.45*W_a)+((0.6*
  W_b)/(3))) // m/s^2
10 // Results
11 clc
12 printf('The acceleration of weight A is %f m/s^2 \n'
  ,a)

```

---

**Scilab code Exa 22.8** Relation between the translatory motion and rotary motion of

```

1 // Initialization of variables
2 r_1=0.075 // m
3 r_2=0.15 // m
4 P=50 // N
5 W=100 // N
6 i_g=0.05 // m
7 theta=30 // degree
8 g=9.81 // m/s^2
9 // Calculations
10 // The eq'n for acceleration of the pool is given by
  solving eqn's 1,2 &3 as,
11 a=(50*g*(r_2*cosd(theta)-r_1))/(100*((i_g^2/r_2)+r_2
  )) // m/s^2
12 // Results
13 clc
14 printf('The acceleration of the pool is %f m/s^2 \n'
  ,a)

```

---



Scilab code Exa 22.10 Relation between the translatory motion and rotary motion of

```
1 // Initialization of variables
2 L=1 // m // length of rod AB
3 m=10 // kg // mass of the rod
4 g=9.81
5 theta=30 // degree
6 // Calculations
7 // solving eq'n 4 for omega we get ,
8 omega=sqrt(2*16.82*sind(theta)) // rad/s
9 // Now solving eq'ns 1 &3 for alpha we get ,
10 alpha=(12/7)*g*cosd(theta) // rad/s
11 // Components of reaction are given as ,
12 R_t=((m*g*cosd(theta))-((m*alpha*L)/4)) // N
13 R_n=((m*omega^2*L)/(4))+((m*g*sind(theta))) // N
14 R=sqrt(R_t^2+R_n^2) // N
15 // Results
16 clc
17 printf('(a) The angular velocity of the rod is %f
        rad/sec \n',omega)
18 printf('(b) The reaction at the hinge is %f N \n',R)
```

---

## Chapter 23

# Kinetics of Rigid Body Work and Energy

Scilab code Exa 23.2 Principle of work and energy for a rigid body

```
1 // Initialization of variables
2 m=600 // kg // mass of the roller
3 r=0.25 // m // radius of the roller
4 P=850 // N // Force
5 v=3 // m/s // velocity to be acquired
6 theta=30 // degree // angle made by v with the force
   P
7 // Calculations
8 // The distance required to be rolled is given by
   equating the Work done between positions 1 & 2 as
   ,
9 x=((3/4)*m*v^2)/(P*cosd(theta)) // m
10 // Results
11 clc
12 printf('The distance required to be rolled is %f m \
   n',x)
```

---

# Chapter 24

## Mechanical Vibrations

Scilab code Exa 24.1 Simple Harmonic Motion

```
1 // Initialization of variables
2 f=1/6 // oscillations/second
3 x=8 // cm // distance from the mean position
4 // Calculations
5 omega=2*pi*f
6 // Amplitude is given by eq'n
7 r=sqrt((25*x^2)/16) // cm
8 // Maximum acceleration is given as,
9 a_max=(pi/3)^2*10 // cm/s^2
10 // Velocity when it is at a dist of 5 cm (assume s=5
    cm) is given by
11 s=5 // cm
12 v=omega*sqrt(r^2-s^2) // cm/s
13 // Results
14 clc
15 printf('(a) The amplitude of oscillation is %f cm \n
    ',r)
16 printf('(b) The maximum acceleration is %f cm/s^2 \n
    ',a_max)
17 printf('(c) The velocity of the particle at 5 cm
    from mean position is %f cm/s \n',v)
```

---

Scilab code Exa 24.2 Simple Harmonic Motion

```
1 // Initialization of variables
2 x_1=0.1 // m // assume the distance of the particle
   from mean position as (x_1 & x_2)
3 x_2=0.2 // m
4 // assume velocities as v_1 & v_2
5 v_1=1.2 // m/s
6 v_2=0.8 // m/s
7 // Calculations
8 // The amplitude of oscillations is given by
   dividing eq'n 1 by 2 as,
9 r=sqrt(0.32/5) // m
10 omega=v_1/(sqrt(r^2-x_1^2)) // radians/second
11 t=(2*pi)/omega // seconds
12 v_max=r*omega // m/s
13 // let the max acceleration be a which is given as,
14 a=r*omega^2 // m/s^2
15 // Results
16 clc
17 printf('(a) The amplitude of oscillations is %f m \n
   ',r)
18 printf('(b) The time period of oscillations is %f
   seconds \n',t)
19 printf('(c) The maximum velocity is %f m/s \n',v_max
   )
20 printf('(d) The maximum acceleration is %f m/s^2 \n'
   ,a) // the value of max acc is incorrect in the
   textbook
21 // NOTE: the value of t is incorrect in the text
   book
22 // The values may differ slightly due to decimal
   point accuracy
```

---

### Scilab code Exa 24.5 Equivalent spring constant

```
1 // Initialization of variables
2 W=50 // N // weight
3 x_0=0.075 // m // amplitude
4 f=1 // oscillation/sec // frequency
5 g=9.81
6 // Calculations
7 omega=2*%pi*f
8 K=((2*%pi)^2*W)/g*(10^-2) // N/cm
9 // let the total extension of the string be delta
   which is given as,
10 delta=(W/K)+(x_0*10^2) // cm
11 T=K*delta // N // Max Tension
12 v=omega*x_0 //m/s // max velocity
13 // Results
14 clc
15 printf('(a) The stiffness of the spring is %f N/cm \
   n',K)
16 printf('(b) The maximum Tension in the spring is %f
   N \n',T)
17 printf('(c) The maximum velocity is %f m/s \n',v)
```

---

### Scilab code Exa 24.10 Pendulum Motion

```
1 // Initialization of variables
2 l=1 // m // length of the simple pendulum
3 g=9.81 // m/s^2
4 // Calculations
5 // Let t_s be the time period when the elevator is
   stationary
6 t_s=2*%pi*sqrt(l/g) /// seconds
```

```

7 // Let t_u be the time period when the elevator
  moves upwards. Then from eqn 1
8 t_u=2*%pi*sqrt((1)/(g+(g/10))) // seconds
9 // Let t_d be the time period when the elevator
  moves downwards.
10 t_d=2*%pi*sqrt(1/(g-(g/10))) // seconds
11 // Results
12 clc
13 printf('The time period of oscillation of the
  pendulum for upward acc of the elevator is %f
  seconds \n',t_u)
14 printf('The time period of oscillation of the
  pendulum for downward acc of the elevator is %f
  seconds \n',t_d)

```

---

#### Scilab code Exa 24.11 Pendulum Motion

```

1 // Initialization of variables
2 t=1 // second // time period of the simple pendulum
3 g=9.81 // m/s^2
4 // Calculations
5 // Length of pendulum is given as,
6 l=(t/(2*%pi)^2)*g // m
7 // Let t_u be the time period when the elevator
  moves upwards. Then the time period is given as,
8 t_u=2*%pi*sqrt((1)/(g+(g/10))) // seconds
9 // Let t_d be the time period when the elevator
  moves downwards.
10 t_d=2*%pi*sqrt(1/(g-(g/10))) // seconds
11 // Results
12 clc
13 printf('The time period of oscillation of the
  pendulum for upward acc of the elevator is %f
  seconds \n',t_u)
14 printf('The time period of oscillation of the

```

```
pendulum for downward acc of the elevator is %f
seconds \n',t_d)
```

---

### Scilab code Exa 24.12 Pendulum Motion

```
1 // Initalization of variables
2 m=15 // kg // mass of the disc
3 D=0.3 // m // diameter of the disc
4 R=0.15 // m // radius
5 l=1 // m // length of the shaft
6 d=0.01 // m // diameter of the shaft
7 G=30*10^9 // N-m^2 // modulus of rigidity
8 // Calculations
9 // M.I of the disc about the axis of rotation is
   given as ,
10 I=(m*R^2)/2 // kg-m^2
11 // Stiffness of the shaft
12 k_t=(%pi*d^4*G)/(32*l) // N-m/radian
13 t=2*%pi*sqrt(I/k_t) // seconds
14 // Results
15 clc
16 printf('The time period of oscillations of the disc
   is %f seconds \n',t)
```

---

# Chapter 25

## Shear Force and Bending Moment

Scilab code Exa 25.5 Shear Force and Bending Moment

```
1 // Initialization of variables
2 L_AB=3 // m // length of the beam
3 L_AC=1 // m
4 L_BC=2 // m
5 M_C=12 // kNm // clockwise moment at C
6 // Calculations
7 // REACTIONS
8 R_B=M_C/L_AB // kN // moment at A
9 R_A=-M_C/L_AB // kN // moment at B
10 // S.F
11 F_A=R_A // kN
12 F_B=R_A // kN
13 // B.M
14 M_A=0 // kNm
15 M_C1=R_A*L_AC // kNm // M_C1 is the BM just before C
16 M_C2=(R_A*L_AC)+M_C // kNm // M_C2 is the BM just
    after C
17 M_B=0 // kNm
18 // Plotting SFD & BMD
```



```

19 x=[0;0.99;1;3]
20 y=[-4;-4;-4;-4]
21 a=[0;0.99;1;3]
22 b=[0;-4;8;0]
23 subplot(221)
24 xlabel("Span (m)")
25 ylabel("Shear Force (kN)")
26 plot(x,y)
27 subplot(222)
28 plot(a,b)
29 xlabel("Span (m)")
30 ylabel("Bending Moment (kNm)")
31 // Results
32 clc
33 printf('The graphs are the solutions')

```

---

#### Scilab code Exa 25.7 Shear Force and Bending Moment

```

1 // Initialization of variables
2 L_AD=8 // m // length of the beam
3 L_AB=2 // m
4 L_BC=4 // m
5 L_CD=2 // m
6 UDL=1 // kN/m
7 P=2 // kN // point load at A
8 // Calculations
9 // REACTIONS
10 // solving eqn's 1&2 using matrix to get R_B & R_C
    as,
11 A=[1 1;1 3]
12 B=[8;30]
13 C=inv(A)*B
14 // SHEAR FORCE
15 // the term F with suffixes 1 & 2 indicates SF just
    to left and right

```

```

16 F_A=-P // kN
17 F_B1=-P // kN
18 F_B2=-P+C(1) // kN
19 F_C1=-P+C(1)-(UDL*L_BC) // kN
20 F_C2=-P+C(1)-(UDL*L_BC)+C(2) // kN
21 F_D=0
22 // BENDING MOMENT
23 // the term F with suffixes 1 & 2 indicates BM just
    to left and right
24 M_A=0 // kNm
25 M_B=(-P*L_CD) // kNm
26 M_C=(-P*(L_AB+L_BC))+(C(1)*L_BC)-(UDL*L_BC*(L_BC/2))
    // kNm
27 M_D=0 // kNm
28 // LOCATION OF MAXIMUM BM
29 // Max BM occurs at E at a distance of 2.5 m from B
    i.e x=L_AE=4.5 m from free end A. Thus max BM is
    given by taking moment at B
30 L_AE=4.5 // m // given
31 M_E=(-2*L_AE)+(4.5*(L_AE-2))-((1/2)*(L_AE-2)^2) //
    kNm
32 // PLOTTING SFD & BMD
33 x=[0;1.99;2;4.5;5.99;6;8]
34 y=[-2;-2;2.5;0;-1.5;2;0]
35 a=[0;2;4.5;6;8]
36 b=[0;-4;-0.875;-2;0]
37 subplot(221)
38 xlabel("Span (m)")
39 ylabel("Shear Force (kN)")
40 plot(x,y)
41 subplot(222)
42 plot(a,b)
43 xlabel("Span (m)")
44 ylabel("Bending Moment (kNm)")
45 // Results
46 clc
47 printf('The graphs are the solutions')

```

---

# Chapter 26

## Appendix

Scilab code Exa 26.1 Appendix

```
1 // 1 APPENDIX. Ex no 1. Page no 638
2 // Initialization of variables
3 P=[-5,2,14] //Point co-ordinates
4 // Calculations
5 r=sqrt(P(1)^2+P(2)^2+P(3)^2) //Magnitude of the
   position vector
6 //Direction cosines
7 l=P(1)/r
8 m=P(2)/r
9 n=P(3)/r
10 //Unit Vector calculations
11 r_unit=P/r
12 //Results
13 clc
14 printf("The Position vector is %fi+%fj+%fk \n",P(1),
   P(2),P(3))
15 printf('The value of r is %f*l i + %f*m j + %f*n k \
   n',r,r,r)
16 printf("The unit vector in the direction of r is %fi
   +%fj+%fk",r_unit(1),r_unit(2),r_unit(3))
```

---

### Scilab code Exa 26.2 Appendix

```
1 // 1 APPENDIX. Ex no 2. Page no 639
2 // Initalizatin of variable
3 F=10 //N
4 P_1=[2,4,3]
5 P_2=[1,-5,2]
6
7 // Calculations
8 d_x=P_2(1)-P_1(1)
9 d_y=P_2(2)-P_1(2)
10 d_z=P_2(3)-P_1(3)
11 d=sqrt(d_x^2+d_y^2+d_z^2)
12 Fx=(F/d)*d_x //N
13 Fy=(F/d)*d_y //N
14 Fz=(F/d)*d_z //N
15 // Direction cosines
16 l=Fx/F
17 m=Fy/F
18 n=Fz/F
19 // Angles
20 theta_x=acosd(l) //degrees
21 theta_y=acosd(m) //degrees
22 theta_z=acosd(n) //degrees
23
24 // Result
25 clc
26 printf("The force in vector notation is %fi%fj%fk\n"
    ,Fx,Fy,Fz)
27 printf("Thetax=%f degrees ,Thetay=%f degrees ,Thetaz=
    %f degrees",theta_x,theta_y,theta_z)
```

---

### Scilab code Exa 26.3 Appendix

```
1 // 1 APPENDIX. Ex no 3. Page no 640
2 //initiation of variables
3 T=2500 //N
4 //Co-ordinates
5 Q=[40,0,-30]
6 P=[0,80,0]
7
8 // Calculations
9 mag_QP=sqrt((P(1)-Q(1))^2+(P(2)-Q(2))^2+(P(3)-Q(3))^2) //Magnitude
10 QP=[(P(1)-Q(1)),(P(2)-Q(2)),(P(3)-Q(3))]
11 F=(T/mag_QP)*QP //N
12 thetax=acosd(F(1)/T) //degrees
13 thetay=acosd(F(2)/T) //degrees
14 thetaz=acosd(F(3)/T) //degrees
15
16 //Result
17 clc
18 printf("The force vector is %fi+%fj+%fk N\n",F(1),F(2),F(3))
19 //Answer in the textbook is printed as 1600 which is incorrect
20 printf("The angles are thetax=%f,thetay=%f and thetaz=%f degrees",thetax,thetay,thetaz)
```

---

### Scilab code Exa 26.4 Appendix

```
1 // 1 APPENDIX. Ex no 4. Page no 642
2 //initilization of variables
3 A=[2,-1,1]
4 B=[1,1,2]
5 C=[3,-2,4]
6 // Calculations
```

```

7 R=[A(1)+B(1)+C(1),A(2)+B(2)+C(2),A(3)+B(3)+C(3)] //
   Resultant
8 mag=sqrt(R(1)^2+R(2)^2+R(3)^2)
9 //Unit vector
10 U=R/mag
11 //Result
12 clc
13 printf("The unit vector is %fi%fj+%fk",U(1),U(2),U
   (3))
14 //Answer for k part is incorrect in the textbook

```

---

#### Scilab code Exa 26.5 Appendix

```

1 // 1 APPENDIX. Ex no 5. Page no 642
2 //initilization of variables
3 A=[2,-6,-3]
4 B=[4,3,-1]
5 //Calculations
6 AdotB=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
7 magA=sqrt(A(1)^2+A(2)^2+A(3)^2)
8 magB=sqrt(B(1)^2+B(2)^2+B(3)^2)
9 theta=acosd(AdotB/(magA*magB)) //degrees
10
11 //Result
12 clc
13 printf("The product of both the vectors is %f\n",
   AdotB)
14 printf("The angle between them is %f degrees",theta)

```

---

#### Scilab code Exa 26.6 Appendix

```

1 // 1 APPENDIX. Ex no 6. Page no 643
2 //initilization of variables

```

```

3 A=[4,-3,1]
4 P=[2,3,-1]
5 Q=[-2,-4,3]
6 // Calculations
7 B=[Q(1)-P(1),Q(2)-P(2),Q(3)-P(3)]
8 AdotB=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
9 magB=sqrt(B(1)^2+B(2)^2+B(3)^2)
10 Acostheta=AdotB/magB
11 // Result
12 clc
13 printf("The value of A.costheta is %f",Acostheta)

```

---

#### Scilab code Exa 26.7 Appendix

```

1 // 1 APPENDIX. Ex no 7. Page no 643
2 // Initialization of variables
3 F=[5,10,-15]
4 a=[1,0,3]
5 b=[3,-1,-6]
6 // Calculations
7 d=b-a
8 work=F.*d
9 Work=work(1)+work(2)+work(3)
10 // Result
11 clc
12 printf("The work done is %f units",Work)

```

---

#### Scilab code Exa 26.8 Appendix

```

1 // 1 APPENDIX. Ex no 8. Page no 644
2 // initialization of variables
3 A=[2,-6,-3]
4 B=[4,3,-1]

```

```

5 // Calculations
6 AcrossB=[(A(2)*B(3)-B(2)*A(3)),A(3)*B(1)-A(1)*B(3),A
           (1)*B(2)-A(2)*B(1)]
7 mag=sqrt(AcrossB(1)^2+AcrossB(2)^2+AcrossB(3)^2)
8 n=AcrossB/mag
9 magA=sqrt(A(1)^2+A(2)^2+A(3)^2)
10 magB=sqrt(B(1)^2+B(2)^2+B(3)^2)
11 theta=asind(mag/(magA*magB))
12 // Result
13 clc
14 printf("The cross prduct of the two vectors is %fi
           %fj+%fk\n",AcrossB(1),AcrossB(2),AcrossB(3)) //
           the answer for j part is wrong in textbook
15 printf("The angle between the two is %f degrees",
           theta)
16 // Only 1 value for theta has been solved. Ref
           textbook for the other value

```

---

### Scilab code Exa 26.9 Appendix

```

1 // 1 APPENDIX. Ex no 9. Page no 645
2 // Initalization of Variable
3 // NOTE:Some Notation has been change to avoid
           conflict
4 // Points As martices
5 A=[0,1,2]
6 B=[1,3,-2]
7 P=[3,6,4]
8 a_s=2 // Angular speed in rad/s
9
10 // Calculations
11 C=[B(1)-A(1),B(2)-A(2),B(3)-A(3)]
12 magC=(C(1)^2+C(2)^2+C(3)^2)^0.5 // Magnitude of the
           Vector C
13 // Unit vector

```



```

14 C_unit=[C(1)/magC,C(2)/magC,C(3)/magC] // Unit
    vector
15 // Position Vector
16 r=[P(1)-A(1),P(2)-A(2),P(3)-A(3)]
17 // Velocity Vector
18 // Calculating the cross product as ,
19 V=[(C(2)*r(3)-C(3)*r(2)),(C(3)*r(1)-C(1)*r(3)),(C(1)
    *r(2)-C(2)*r(1))]
20 // Vector notation
21 V_n=(a_s/magC)*[V(1),V(2),V(3)]
22 // Velocity Magnitude
23 magV=sqrt(V(1)^2+V(2)^2+V(3)^2)
24 v=(a_s/magC)*magV
25 // Result
26 clc
27 printf("The vector notation of velocity is %fi %fj
    %fk \n",V_n(1),V_n(2),V_n(3))
28 printf("The magnitude of the Velocity Vector is %f \
    n",v)

```

---

### Scilab code Exa 26.10 Appendix

```

1 // 1 APPENDIX. Ex no 10. Page no 647
2 // Initialization of variables
3 // Points as matrices
4 O=[-2,3,5]
5 P=[1,-2,4]
6 Q=[5,2,3]
7 F=[4,4,-1] // Force vector
8 // Calculations
9 // Position vector ( we will solve only by using r_1
    as r_2 also gives the same answer)
10 r_1=[P(1)-O(1),P(2)-O(2),P(3)-O(3)]
11 // Moment
12 // Calculating the cross product

```

```

13 M=[(r_1(2)*F(3)-r_1(3)*F(2)),(r_1(3)*F(1)-r_1(1)*F
      (3)),(r_1(1)*F(2)-r_1(2)*F(1))]
14 // Results
15 clc
16 printf('The Moment about point O is %fi %fj+%fk \n',
      M(1),M(2),M(3))

```

---

### Scilab code Exa 26.11 Appendix

```

1 // 1 APPENDIX. Ex no 11. Page no 647
2 // Initalization of variables
3 // Points as matrices
4 P=[1,-1,2] // Point where force is applied
5 O=[2,-1,3] // point where moment is to be found
6 F=[3,2,-4] // Force vector
7 // Calculations
8 // Position vector of point P wrt O
9 r=[P(1)-O(1),P(2)-O(2),P(3)-O(3)]
10 // Moment
11 M=[(r(2)*F(3)-r(3)*F(2)),(r(3)*F(1)-r(1)*F(3)),(r(1)
      *F(2)-r(2)*F(1))]
12 // Results
13 clc
14 printf('The moment of the force is %fi %fj %fk \n',M
      (1),M(2),M(3))

```

---

### Scilab code Exa 26.12 Appendix

```

1 // 1 APPENDIX. Ex no 12. Page no 648
2 // Initalization of variables
3 // Different notations have been used at some places
4 f=22 // N
5 // Points as matrices

```

```

6 A=[4,-1,7]
7 O=[1,-3,2]
8 V=[9,6,-2] // Given vector
9 // Calculations
10 // Unit vector in the direction of the vector
11 v=[V(1),V(2),V(3)]/sqrt(V(1)^2+V(2)^2+V(3)^2)
12 // Force
13 F=f*[v(1),v(2),v(3)]
14 // Position vector of point A wrt O
15 r=[A(1)-O(1),A(2)-O(2),A(3)-O(3)]
16 // Moment
17 M=[(r(2)*F(3)-r(3)*F(2)),(r(3)*F(1)-r(1)*F(3)),(r(1)
    *F(2)-r(2)*F(1))]
18 // Results
19 clc
20 printf('The moment is %fi + %fj + %fk \n',M(1),M(2)
    ,M(3))

```

---

### Scilab code Exa 26.13 Appendix

```

1 // 1 APPENDIX. Ex no 13. Page no 648
2 // Initialization of variables
3 // Notations have been changed
4 // Force Vector
5 F=[50,-80,30]
6 // from fig.13
7 theta1=30 // angles by which the axis is rotated (
    all in degrees)
8 theta2=60
9 theta3=90
10 theta4=120
11 theta5=0
12 // Calculations
13 // Unit vector in u-direction
14 u_unit=[1*cosd(theta1),1*cosd(theta2),1*cosd(theta3)]

```

```

    ]
15 // Unit vector in v-direction
16 v_unit=[1*cosd(theta4),1*cosd(theta1),1*cosd(theta3)
    ]
17 // Unit vector in w-direction
18 w_unit=[1*cosd(theta3),1*cosd(theta3),1*cosd(theta5)
    ]
19 // Components of force
20 // finding the dot product as
21 u=F(1)*u_unit(1)+F(2)*u_unit(2)+F(3)*u_unit(3) // N
22 v=F(1)*v_unit(1)+F(2)*v_unit(2)+F(3)*v_unit(3) // N
23 w=F(1)*w_unit(1)+F(2)*w_unit(2)+F(3)*w_unit(3) // N
24 // Results
25 clc
26 printf('The components of the force is ,along u=%f N,
    along v=%f N, along w=%f N \n',u,v,w)

```

---

#### Scilab code Exa 26.14 Appendix

```

1 // 1 APPENDIX. Ex no 14. Page no 649
2 // Initialization of variables
3 // Some notations have been assumed
4 f=100 // N // magnitude of force
5 // Co-ordinates of corners of the box as matrices
6 A=[0,0,0]
7 B=[0.5,0,0]
8 C=[0.5,0,1]
9 D=[0,0,1]
10 E=[0,0.5,0]
11 F=[0.5,0.5,0]
12 G=[0.5,0.5,1]
13 H=[0,0.5,1]
14 // Calculations
15 // Force vector
16 Fmag=f/sqrt((F(1)-C(1))^2+(F(2)-C(2))^2+(F(3)-C(3))

```

```

^2)
17 F=Fmag*[F(1)-C(1),F(2)-C(2),F(3)-C(3)]
18 // Position vector
19 r_EC=[C(1)-E(1),C(2)-E(2),C(3)-E(3)]
20 // Moment about point E
21 // Calculating the cross product
22 M_E=[(r_EC(2)*F(3)-r_EC(3)*F(2)),(r_EC(3)*F(1)-r_EC
      (1)*F(3)),(r_EC(1)*F(2)-r_EC(2)*F(1))] // N.m //
      The value taken for F is incorrect in textbook.
23 // Unit vector
24 n_AE=[E(1)-A(1),E(2)-A(2),E(3)-A(3)]/sqrt((E(1)-A(1)
      )^2+(E(2)-A(2))^2+(E(3)-A(3))^2)
25 // Moment of force about axis AE
26 // finding the dot product
27 M_AE=M_E(1)*n_AE(1)+M_E(2)*n_AE(2)+M_E(3)*n_AE(3) //
      N.m
28 // Results
29 clc
30 printf('The moment of the force about point E is %fi
      - %fj + %fk N.m \n',M_E)
31 printf('The moment of force about axis AE is -%f N.m
      \n',M_AE)
32 // The value of MAE & ME is incorrect in the
      textbook.Incorrect value of force vector is taken
      in calculation of ME

```

---

### Scilab code Exa 26.15 Appendix

```

1 // 1 APPENDIX. Ex no 15. Page no 652
2 // Initialization of variables
3 // Consider fig. 16. The co-ordinates of various
      points are
4 // Co-ordinates as matrices
5 // Some of the notations have been changed
6 f1=5 // kN

```

```

7 f2=7.5 // kN
8 f3=10 // kN
9 A=[0,0,0]
10 E=[0,1,0]
11 D=[0,0,2]
12 F=[3,1,0]
13 G=[3,1,2]
14 H=[0,1,2] // co-ordinates of H not given in book.
    ref fig.16 for the same
15 // Calculations
16 // Force vectors
17 F1=(f1/sqrt((F(1)-E(1))^2+(F(2)-E(2))^2+(F(3)-E(3))^2))*[F(1)-E(1),F(2)-E(2),F(3)-E(3)]
18 F2=(f2/sqrt((D(1)-H(1))^2+(D(2)-H(2))^2+(D(3)-H(3))^2))*[D(1)-H(1),D(2)-H(2),D(3)-H(3)]
19 F3=(f3/sqrt((G(1)-E(1))^2+(G(2)-E(2))^2+(G(3)-E(3))^2))*[G(1)-E(1),G(2)-E(2),G(3)-E(3)]
20 // Resultant force
21 R=F1+F2+F3 // kN
22 // Position vectors
23 r_AE=[E(1)-A(1),E(2)-A(2),E(3)-A(3)]
24 r_AD=[D(1)-A(1),D(2)-A(2),D(3)-A(3)]
25 // Moment of forces about A
26 // Calculating the cross product
27 M1=[(r_AE(2)*F1(3)-r_AE(3)*F1(2)),(r_AE(3)*F1(1)-r_AE(1)*F1(3)),(r_AE(1)*F1(2)-r_AE(2)*F1(1))]
28 M2=[(r_AD(2)*F2(3)-r_AD(3)*F2(2)),(r_AD(3)*F2(1)-r_AD(1)*F2(3)),(r_AD(1)*F2(2)-r_AD(2)*F2(1))]
29 M3=[(r_AE(2)*F3(3)-r_AE(3)*F3(2)),(r_AE(3)*F3(1)-r_AE(1)*F3(3)),(r_AE(1)*F3(2)-r_AE(2)*F3(1))]
30 // Resultant moment
31 M_R=M1+M2+M3 // KN.m
32 // Results
33 clc
34 printf('The resultant force is %fi %fj + %fk kN \n',
    R)
35 printf('The resultant moment of all the forces about
    point A is %fi + %fj %fk kN.m \n',M_R)

```

---

### Scilab code Exa 26.16 Appendix

```
1 // 1 APPENDIX. Ex no 16. Page no 654
2 // Initiization of variables
3 F=20 // kN // Force acting at O
4 M_x=76 // kNm
5 M_y=82 // kNm
6 // Calculations
7 x=M_x/F // m
8 z=M_y/F // m
9 // Results
10 clc
11 printf('The point of application should be shifted
        to: x=%f m and z=%f m \n',x,z)
```

---

### Scilab code Exa 26.17 Appendix

```
1 // 1 APPENDIX. Ex no 17. Page no 655
2 // Initalization of variables
3 W=5000 // N
4 // Co-ordinates of various points
5 A=[0,4.5,0]
6 B=[2.8,0,0]
7 C=[0,0,-2.4]
8 D=[-2.6,0,1.8]
9 // Calculations
10 // Ref textbook for the values of tenion in the
    cable AB, AC & AD. The values consist of
    variables which cannot be defined here
11 // We re-arrange and define the equations of
    equilibrium as matrices and solve them as,
```

```

12 P=[0.528,0.0,-0.472;0.0,0.47,-0.327;0.85,0.88,0.818]
13 Q=[0;0;5000]
14 X=inv(P)*Q
15 // Results
16 clc
17 printf('Tension in cable AD is %f N \n',X(3))
18 printf('Tension in cable AB is %f N \n',X(1))
19 printf('Tension in cable AC is %f N \n',X(2))
20 // The answer may vary slightly due to decimal point
    error. Ans for TAB is incorrect in textbook.

```

---

#### Scilab code Exa 26.18 Appendix

```

1 // 1 APPENDIX. Ex no 18. Page no 656
2 // Initilization of variables
3 P=5 // kN
4 Q=3 // kN
5 C=5 // kNm // couple
6 // ref fig.20 // Notations have been assumed
7 z1=1.5 // m
8 z2=0.625 // m
9 z3=0.5 // m
10 x1=3.5 // m
11 x2=0.625 // m
12 // Calculations
13 // sum Mx=0
14 RA=(P*z2)+(Q*z3)+C)/z1 // kN
15 // Mz=0
16 RC=(Q*x1)+(P*x2))/x1 // kN
17 // sum Fy=0
18 RB=P+Q-RA-RC // kN
19 // Results
20 clc
21 printf('The reactions are: RA=%f kN ,RC=%f kN and
    RB=%f kN \n',RA,RC,RB)

```



---

Scilab code Exa 26.19 Appendix

```
1 // 1 APPENDIX. Ex no 19. Page no 657
2 // Initialization of variables
3 F=2 // kN
4 W=1 // kN
5 // Co-ordinates as matrices
6 A=[0,0,0]
7 C=[0,0,1.2]
8 B=[0,0,2.5]
9 D=[-1,1,0]
10 E=[1,1,0]
11 F=[0,0,1]
12 G=[0,0,2]
13 // Force vector
14 f=[0,-2,0]
15 // Weight vector
16 w=[0,-1,0]
17 // Calculations
18 // we have 5 unknowns: A_x,A_y,A_z,T_FE & T_GD
19 // we define and solve eqn's 1,2,3,4&5 using matrix
    as,
20 P
    =[1,0,0,0.58,-0.41;0,1,0,0.58,0.41;0,0,1,-0.58,-0.82;0,0,0,0.58,0
21 Q=[0;3;0;6.25;0]
22
23 X=inv(P)*Q
24
25 // Results
26 clc
27 printf('The components of reaction at A are: A_x=%f
    kN , A_y=%f kN and A_z=%f kN \n',X(1),X(2),X(3))
28 printf('The tensions in the cable are: T_FE=%f kN
```

```
and T_GD=%f kN \n',X(4),X(5))
29 // The solution in the textbook is incorrect and
    yeilds singularity in matrix calculation.
```

---