

Scilab Textbook Companion for
Engineering Mechanics-Schaum Series
by McLean¹

Created by
Akshay
Engineering
Civil Engineering
university Of Pune
College Teacher
None
Cross-Checked by
Spandana

April 8, 2015

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

Book Description

Title: Engineering Mechanics-Schaum Series

Author: McLean

Publisher: McGraw-Hill, New Delhi

Edition: 5

Year: 1997

ISBN: 0-07-046193-7

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	5
1 Vectors	14
2 Operations With Forces	25
3 Resolution of Coplanar Force System	32
4 Resolution of Non Coplanar Force System	45
5 Equilibrium of Coplanar Force System	52
6 Equilibrium of Non Coplanar Force System	63
7 Trusses and Cables	73
8 Forces In Beams	83
9 Friction	87
10 First Moment and Centroid	99
11 Virtual Work	105
12 Kinematics of a Particle	108
13 Dynamics of a Particle	128
14 Kinematics of a Rigid Body in Plane Motion	140

15 Moment of Inertia	150
16 Dynamics of a Rigid Body in Plane Motion	157
17 Work and Energy	184
18 Impulse and Momentum	202
19 Mechanical Vibration	223

List of Scilab Codes

Exa 1.1	Vectors	14
Exa 1.2	Vectors	14
Exa 1.3	Vectors	15
Exa 1.4	Vectors	16
Exa 1.5	Vectors	16
Exa 1.6	Vectors	17
Exa 1.7	Vectors	18
Exa 1.8	Vectors	18
Exa 1.9	Vectors	19
Exa 1.10	Vectors	19
Exa 1.11	Vectors	20
Exa 1.13	Vectors	20
Exa 1.14	Vectors	21
Exa 1.15	Vectors	21
Exa 1.16	Vectors	22
Exa 1.17	Vectors	23
Exa 1.19	Vectors	23
Exa 2.1	OPF	25
Exa 2.2	OPF	25
Exa 2.3	OPF	26
Exa 2.4	OPF	27
Exa 2.5	OPF	27
Exa 2.6	OPF	28
Exa 2.8	OPF	29
Exa 2.11	OPF	29
Exa 2.12	OPF	30
Exa 2.13	OPF	30
Exa 2.14	OPF	31

Exa 3.1	Resolution	32
Exa 3.2	Resolution	33
Exa 3.3	Resolution	34
Exa 3.4	Resolution	34
Exa 3.5	Resolution	35
Exa 3.6	Resolution	36
Exa 3.7	Resolution	36
Exa 3.8	Resolution	37
Exa 3.9	Resolution	38
Exa 3.10	Resolution	39
Exa 3.11	Resolution	40
Exa 3.12	Resolution	41
Exa 3.13	Resolution	42
Exa 3.14	Resolution	43
Exa 4.1	Resolution Non Coplanar	45
Exa 4.2	Resolution Non Coplanar	46
Exa 4.3	Resolution Non Coplanar	47
Exa 4.4	Resolution Non Coplanar	48
Exa 4.5	Resolution Non Coplanar	50
Exa 5.1	Equilibrium of CFS	52
Exa 5.2	Equilibrium of CFS	52
Exa 5.3	Equilibrium of CFS	53
Exa 5.4	Equilibrium of CFS	53
Exa 5.5	Equilibrium of CFS	54
Exa 5.6	Equilibrium of CFS	54
Exa 5.8	Equilibrium of CFS	55
Exa 5.9	Equilibrium of CFS	56
Exa 5.10	Equilibrium of CFS	56
Exa 5.11	Equilibrium of CFS	57
Exa 5.12	Equilibrium of CFS	57
Exa 5.13	Equilibrium of CFS	58
Exa 5.15	Equilibrium of CFS	58
Exa 5.16	Equilibrium of CFS	59
Exa 5.17	Equilibrium of CFS	60
Exa 5.18	Equilibrium of CFS	60
Exa 5.19	Equilibrium of CFS	61
Exa 6.1	Equilibrium of NCFS	63
Exa 6.2	Equilibrium of NCFS	64

Exa 6.3	Equilibrium of NCFS	65
Exa 6.4	Equilibrium of NCFS	65
Exa 6.5	Equilibrium of NCFS	66
Exa 6.6	Equilibrium of NCFS	67
Exa 6.7	Equilibrium of NCFS	68
Exa 6.8	Equilibrium of NCFS	68
Exa 6.9	Equilibrium of NCFS	69
Exa 6.11	Equilibrium of NCFS	69
Exa 6.12	Equilibrium of NCFS	70
Exa 6.13	Equilibrium of NCFS	70
Exa 6.14	Equilibrium of NCFS	71
Exa 6.15	Equilibrium of NCFS	72
Exa 7.1	Truss and Cable	73
Exa 7.2	Truss and Cable	74
Exa 7.3	Truss and Cable	75
Exa 7.4	Truss and Cable	75
Exa 7.5	Truss and Cable	76
Exa 7.6	Truss and Cable	77
Exa 7.7	Truss and Cable	78
Exa 7.8	Truss and Cable	78
Exa 7.9	Truss and Cable	79
Exa 7.10	Truss and Cable	79
Exa 7.11	Truss and Cable	80
Exa 7.12	Truss and Cable	80
Exa 7.13	Truss and Cable	81
Exa 8.1	Beams	83
Exa 8.2	Beams	83
Exa 8.3	Beams	84
Exa 8.5	Beams	85
Exa 9.1	Friction	87
Exa 9.3	Friction	87
Exa 9.4	Friction	88
Exa 9.5	Friction	89
Exa 9.6	Friction	90
Exa 9.7	Friction	90
Exa 9.8	Friction	91
Exa 9.9	Friction	91
Exa 9.10	Friction	92

Exa 9.11	Friction	93
Exa 9.12	Friction	93
Exa 9.13	Friction	93
Exa 9.15	Friction	94
Exa 9.16	Friction	95
Exa 9.17	Friction	95
Exa 9.18	Friction	96
Exa 9.19	Friction	97
Exa 9.20	Friction	97
Exa 9.21	Friction	97
Exa 10.5	1st Moment and CG	99
Exa 10.6	1st Moment and CG	100
Exa 10.11	1st Moment and CG	100
Exa 10.16	1st Moment and CG	101
Exa 10.17	1st Moment and CG	102
Exa 10.26	1st Moment and CG	102
Exa 10.27	1st Moment and CG	103
Exa 10.28	1st Moment and CG	104
Exa 11.5	V W	105
Exa 11.6	V W	105
Exa 11.7	V W	106
Exa 11.10	V W	106
Exa 12.1	Kin of a Part	108
Exa 12.2	Kin of a Part	108
Exa 12.3	Kin of a Part	109
Exa 12.4	Kin of a Part	110
Exa 12.5	Kin of a Part	110
Exa 12.6	Kin of a Part	111
Exa 12.7	Kin of a Part	112
Exa 12.8	Kin of a Part	113
Exa 12.9	Kin of a Part	113
Exa 12.10	Kin of a Part	114
Exa 12.12	Kin of a Part	114
Exa 12.14	Kin of a Part	115
Exa 12.16	Kin of a Part	115
Exa 12.17	Kin of a Part	116
Exa 12.18	Kin of a Part	117
Exa 12.20	Kin of a Part	117

Exa 12.21	Kin of a Part	118
Exa 12.25	Kin of a Part	119
Exa 12.26	Kin of a Part	119
Exa 12.27	Kin of a Part	120
Exa 12.28	Kin of a Part	120
Exa 12.29	Kin of a Part	121
Exa 12.30	Kin of a Part	121
Exa 12.31	Kin of a Part	122
Exa 12.32	Kin of a Part	122
Exa 12.34	Kin of a Part	123
Exa 12.35	Kin of a Part	124
Exa 12.36	Kin of a Part	124
Exa 12.37	Kin of a Part	125
Exa 12.38	Kin of a Part	125
Exa 12.39	Kin of a Part	126
Exa 12.40	Kin of a Part	126
Exa 12.41	Kin of a Part	127
Exa 13.1	Dyna of a Part	128
Exa 13.2	Dyna of a Part	128
Exa 13.3	Dyna of a Part	129
Exa 13.4	Dyna of a Part	130
Exa 13.7	Dyna of a Part	130
Exa 13.8	Dyna of a Part	131
Exa 13.10	Dyna of a Part	131
Exa 13.11	Dyna of a Part	132
Exa 13.12	Dyna of a Part	133
Exa 13.13	Dyna of a Part	133
Exa 13.14	Dyna of a Part	134
Exa 13.15	Dyna of a Part	135
Exa 13.16	Dyna of a Part	135
Exa 13.19	Dyna of a Part	136
Exa 13.20	Dyna of a Part	137
Exa 13.28	Dyna of a Part	137
Exa 13.29	Dyna of a Part	137
Exa 13.30	Dyna of a Part	138
Exa 13.31	Dyna of a Part	139
Exa 14.2	Kin of rig body in PM	140
Exa 14.3	Kin of rig body in PM	141

Exa 14.4	Kin of rig body in PM	141
Exa 14.9	Kin of rig body in PM	142
Exa 14.10	Kin of rig body in PM	142
Exa 14.13	Kin of rig body in PM	143
Exa 14.14	Kin of rig body in PM	143
Exa 14.15	Kin of rig body in PM	144
Exa 14.18	Kin of rig body in PM	145
Exa 14.19	Kin of rig body in PM	145
Exa 14.20	Kin of rig body in PM	146
Exa 14.21	Kin of rig body in PM	147
Exa 14.22	Kin of rig body in PM	147
Exa 14.23	Kin of rig body in PM	148
Exa 14.24	Kin of rig body in PM	148
Exa 14.26	Kin of rig body in PM	149
Exa 15.11	MI	150
Exa 15.12	MI	151
Exa 15.14	MI	152
Exa 15.15	MI	152
Exa 15.20	MI	153
Exa 15.24	MI	153
Exa 15.25	MI	154
Exa 15.30	MI	154
Exa 15.36	MI	155
Exa 15.37	MI	155
Exa 16.2	Dyna of rig body in PM	157
Exa 16.3	Dyna of rig body in PM	158
Exa 16.5	Dyna of rig body in PM	158
Exa 16.6	Dyna of rig body in PM	159
Exa 16.8	Dyna of rig body in PM	160
Exa 16.9	Dyna of rig body in PM	161
Exa 16.10	Dyna of rig body in PM	161
Exa 16.11	Dyna of rig body in PM	162
Exa 16.12	Dyna of rig body in PM	163
Exa 16.14	Dyna of rig body in PM	164
Exa 16.16	Dyna of rig body in PM	164
Exa 16.18	Dyna of rig body in PM	165
Exa 16.21	Dyna of rig body in PM	165
Exa 16.24	Dyna of rig body in PM	166

Exa 16.25	Dyna of rig body in PM	167
Exa 16.26	Dyna of rig body in PM	168
Exa 16.27	Dyna of rig body in PM	168
Exa 16.28	Dyna of rig body in PM	169
Exa 16.29	Dyna of rig body in PM	169
Exa 16.30	Dyna of rig body in PM	170
Exa 16.31	Dyna of rig body in PM	171
Exa 16.32	Dyna of rig body in PM	171
Exa 16.33	Dyna of rig body in PM	172
Exa 16.34	Dyna of rig body in PM	173
Exa 16.36	Dyna of rig body in PM	173
Exa 16.43	Dyna of rig body in PM	174
Exa 16.47	Dyna of rig body in PM	175
Exa 16.48	Dyna of rig body in PM	175
Exa 16.49	Dyna of rig body in PM	176
Exa 16.50	Dyna of rig body in PM	176
Exa 16.51	Dyna of rig body in PM	177
Exa 16.52	Dyna of rig body in PM	178
Exa 16.53	Dyna of rig body in PM	178
Exa 16.54	Dyna of rig body in PM	179
Exa 16.55	Dyna of rig body in PM	179
Exa 16.56	Dyna of rig body in PM	180
Exa 16.57	Dyna of rig body in PM	180
Exa 16.58	Dyna of rig body in PM	181
Exa 16.60	Dyna of rig body in PM	181
Exa 16.61	Dyna of rig body in PM	182
Exa 16.62	Dyna of rig body in PM	182
Exa 17.4	Work	184
Exa 17.6	Work	184
Exa 17.7	Work	185
Exa 17.9	Work	186
Exa 17.10	Work	186
Exa 17.11	Work	187
Exa 17.12	Work	187
Exa 17.13	Work	188
Exa 17.15	Work	188
Exa 17.16	Work	189
Exa 17.18	Work	189

Exa 17.19	Work	190
Exa 17.21	Work	191
Exa 17.22	Work	191
Exa 17.23	Work	192
Exa 17.24	Work	192
Exa 17.25	Work	193
Exa 17.26	Work	194
Exa 17.28	Work	194
Exa 17.31	Work	195
Exa 17.32	Work	196
Exa 17.33	Work	196
Exa 17.34	Work	197
Exa 17.35	Work	198
Exa 17.36	Work	198
Exa 17.37	Work	199
Exa 17.38	Work	200
Exa 17.40	Work	200
Exa 17.41	Work	201
Exa 18.8	Imp	202
Exa 18.9	Imp	203
Exa 18.10	Imp	203
Exa 18.11	Imp	204
Exa 18.12	Imp	204
Exa 18.13	Imp	205
Exa 18.14	Imp	206
Exa 18.15	Imp	206
Exa 18.17	Imp	207
Exa 18.18	Imp	207
Exa 18.19	Imp	208
Exa 18.21	Imp	209
Exa 18.22	Imp	209
Exa 18.23	Imp	210
Exa 18.24	Imp	211
Exa 18.25	Imp	212
Exa 18.26	Imp	212
Exa 18.27	Imp	213
Exa 18.28	Imp	213
Exa 18.29	Imp	214

Exa 18.30	Imp	214
Exa 18.32	Imp	215
Exa 18.33	Imp	215
Exa 18.34	Imp	216
Exa 18.35	Imp	217
Exa 18.39	Imp	217
Exa 18.40	Imp	218
Exa 18.41	Imp	218
Exa 18.42	Imp	219
Exa 18.46	Imp	219
Exa 18.48	Imp	220
Exa 18.50	Imp	220
Exa 18.51	Imp	221
Exa 18.52	Imp	222
Exa 19.2	Vibrations	223
Exa 19.11	Vibrations	223
Exa 19.13	Vibrations	224
Exa 19.14	Vibrations	225
Exa 19.15	Vibrations	225
Exa 19.16	Vibrations	226
Exa 19.19	Vibrations	226
Exa 19.20	Vibrations	227
Exa 19.24	Vibrations	227
Exa 19.25	Vibrations	228
Exa 19.28	Vibrations	228

Chapter 1

Vectors

Scilab code Exa 1.1 Vectors

```
1 //Initialisation of Variables
2 f1=120 //lb
3 f2=100 //lb
4 theta=((60*%pi)/180) //radians
5 //Calculations
6 R=sqrt(120^2+100^2-(2*120*100*cos(theta))) //
   Applying Thr rule of Cosines
7 alpha1((((asin(120*sin(theta)/111))*180)/%pi) //
   Applying the Law of Sines
8 alpha=alpha1+270 //As the vector lies in the fourth
   Quadrant by obsrevaton
9 //Results
10 clc
11 printf('The Resultant of The force system is equal
   to:%f N\n ',R ) //lb
12 printf('The Resultant is at:%f degrees ',alpha) //
   degrees
```

Scilab code Exa 1.2 Vectors

```

1 //Initilization of variables
2 P=100 //lb
3 Q=120 //lb
4 theta=((30*%pi)/180) //radians
5 // Calculations
6 R_x=Q*cos(theta) //lb
7 R_y=Q*sin(theta)-P //lb
8 R=sqrt(R_x^2+R_y^2) //lb Triangle law
9 Theta_1=((atan(R_y/R_x))*180)/%pi //degrees
10 Theta_R=360+Theta_1 //degrees
11 //Result
12 clc
13 printf('The resultant of the force system is:%f N\n',
        ,R) //lb
14 printf('The resultant is at:%f degrees',Theta_R) //
        Degrees

```

Scilab code Exa 1.3 Vectors

```

1 //Initialization of variables
2 R=400 // N
3 F2=200 //N
4 Theta1=((120*%pi)/180) //radians
5 Theta2=((20*%pi)/180) //radians
6 Theta=Theta1-Theta2 //radians
7 //Calculation
8 F=sqrt(R^2+F2^2-(2*R*F2*cos(Theta))) //N Applying
        the Rule of Cosine
9 Theta_r=asin((400*sin(Theta))/F) //radians Applying
        the rule of sines
10 Theta_R=(Theta_r*180)/%pi
11 //Result
12 clc
13 printf('The resultant of the force system is:%f N \n
        ',F) //N

```



```
14 printf('The Angle between F and 200N force is :%f
degrees \n',Theta_R) //degrees
```

Scilab code Exa 1.4 Vectors

```
1 //Initialization of variables
2 F1=280 //N
3 F2=130 //N
4 Theta1=((320*%pi)/180) //Radians
5 Theta2=((60*%pi)/180) //Radians
6 //Calculations
7 R_x=-F1*cos(Theta1)+F2*cos(Theta2) //N
8 R_y=F1*sin(Theta1)-F2*sin(Theta2) //N
9 R=sqrt(R_x^2+R_y^2) //N Applying Triangle Law
10 ThetaR=atan(R_y/R_x) //radians
11 Theta_R=360-(ThetaR*180/%pi) //degrees
12 //Result
13 clc
14 printf('The resultant of the force system is:%f N\n',
,R) //N
15 printf('The resultant is at:%f degrees',Theta_R) //
degrees
16 //The decimal point accuracy causes discrepancy in
answers
```

Scilab code Exa 1.5 Vectors

```
1 //Initialization of variables
2 F1=26 //lb
3 F2=39 //lb
4 F3=63 //lb
5 F4=57 //lb
6 T1=((10*%pi)/180) //Radians
```

```

7 T2=((114*%pi)/180) //Radians
8 T3=((183*%pi)/180) //radians
9 T4=((261*%pi)/180) //radians
10 //Calculations
11 R_x=F1*cos(T1)+F2*cos(T2)+F3*cos(T3)+F4*cos(T4) //lb
    Resolving vectors
12 R_y=F1*sin(T1)+F2*sin(T2)+F3*sin(T3)+F4*sin(T4) //lb
    resolving vectors
13 R=sqrt(R_x^2+R_y^2) //lb Applying Triangle Law
14 theta=atan(R_y/R_x) //radians
15 Theta=theta*180/%pi //degrees
16 Theta_R=180+Theta
17 //Results
18 clc
19 printf('The Resultant of the force system is:%f lb\n
    ',R) //lb
20 printf('The resultant is at:%f degrees ',Theta_R) //
    degrees

```

Scilab code Exa 1.6 Vectors

```

1 //Initilization of variables
2 F=10 //lb
3 theta1=((60*%pi)/180) //radians
4 theta2=((45*%pi)/180) //radians
5 theta=theta1-theta2 //radians
6 //Calculation
7 F_OH=F/cos(theta) //lb resolving vectors
8 //Result
9 clc
10 printf('The component of F in the direction of OH is
    :%f lb ',F_OH) //lb

```

Scilab code Exa 1.7 Vectors

```
1 //Initilization of variables
2 weight=80 //kg
3 theta=((20*%pi)/180) //radians
4 theta_p=((70*%pi)/180)
5 //Calculations
6 //Part (a)
7 F=weight*9.81 //N
8 R=F*cos(theta) //N
9 //part (b)
10 R_p=F*cos(theta_p) //N
11 //Result
12 clc
13 printf('The normal component is:%f N \n',R) //N
14 printf('The parallel component is:%f N',R_p) //N
```

Scilab code Exa 1.8 Vectors

```
1 //Initilization of variables
2 P=235 //N
3 theta=((60*%pi)/180) //radians
4 bet=((22*%pi)/180) //radians
5 gam=((38*%pi)/180) //radians
6 //Calculations
7 //Part (a)
8 P_h=P*cos(theta) //N
9 P_v=P*sin(theta) //N
10 //Part (b)
11 P_l=P*cos(theta-bet) //N
12 P_p=P*sin(gam) //N
13 //Result
14 clc
15 printf('The horizontal component is:%f N\n',P_h) //N
16 printf('The vertical component is:%f N\n',P_v) //N
```

```

17 printf('The component parallel to plane is:%f N\n',
    P_1) //N
18 printf('The component perpendicular to the plane is:
    %f N',P_p) //N
19 //The decimal point accuracy might cause a small
    discrepancy in the answers

```

Scilab code Exa 1.9 Vectors

```

1 //Initilization of variables
2 F1=90 //lb
3 theta1=((40*%pi)/180) //radians
4 theta2=((30*%pi)/180) //radians
5 //Calculations
6 R_x=0 //lb
7 R_y=20 //lb
8 //Taking the sum of forces in the X-Direction
9 P=((F1*cos(theta1))/cos(theta2)) //lb
10 //Taking the sum of the forces in the Y-Direction
11 F=(P*sin(theta2))+(F1*sin(theta1))-20 //lb
12 //Results
13 clc
14 printf('The value of P is:%f lb\n',P) //lb
15 printf('The value of F is:%f lb',F) //lb

```

Scilab code Exa 1.10 Vectors

```

1 //Initilization of variables
2 x=4 //m
3 y=3 //m
4 z=2 //m
5 F=50 //N
6 //Calculations

```

```

7 OP=sqrt(x^2+y^2+z^2) //m
8 thetax=(x/OP) //radians
9 thetay=(y/OP) //Radians
10 thetaz=(z/OP) //radians
11 P_x=F*(thetax) //N
12 P_y=F*(thetay) //N
13 P_z=F*(thetaz) //N
14 //Result
15 clc
16 printf('The vector P is :%fi+%fj+%fk N',P_x,P_y,P_z)
    //N

```

Scilab code Exa 1.11 Vectors

```

1 //Initilization of variables
2 x=2
3 y=-4
4 z=1
5 F=100 //N
6 //Calculation
7 thetax=x/sqrt(x^2+y^2+z^2) //radians
8 thetay=y/sqrt(x^2+y^2+z^2) //radians
9 thetaz=z/sqrt(x^2+y^2+z^2) //radians
10 P_x=F*thetax //N
11 P_y=F*thetay //N
12 P_z=F*thetaz //N
13 //Result
14 clc
15 printf('The vector P is :%fi%fj+%fk N',P_x,P_y,P_z)
    //N

```

Scilab code Exa 1.13 Vectors

```

1 //Initilization of variables
2 Fx=2.63 //N
3 Fy=4.28 //N
4 Fz=-5.92 //N
5 // Calculation
6 F=sqrt(Fx^2+Fy^2+Fz^2) //N
7 thetax=((acos(Fx/F))*180)/%pi //degrees
8 thetay=((acos(Fy/F))*180)/%pi //degrees
9 thetaz=((acos(Fz/F))*180)/%pi //degrees
10 //Result
11 clc
12 printf('The magnitude of force is:%f N \n',F) //N
13 printf('Thetax:%f degrees\n',thetax) //degrees
14 printf('Thetay:%f degrees\n',thetay) //degrees
15 printf('Thetaz:%f degrees ',thetaz) //degrees

```

Scilab code Exa 1.14 Vectors

```

1 //Initilization of variables
2 P=[4.82, -2.33, 5.47] //N
3 Q=[-2.81, -6.09, 1.12 ] //m
4 // Calculations
5 M=P*Q' //Nm
6 //Results
7 clc
8 printf('Result is:%f N.m',M) //N-m

```

Scilab code Exa 1.15 Vectors

```

1 //Initilization of variables
2 x1=2 //units
3 x2=-2 //units
4 y1=3 //units

```

```

5 y2=4 //units
6 z1=0 //units
7 z2=6 //units
8 P=[2,3,-1] //units
9 //Calculations
10 X=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2) //units
11 eLx=(x2-x1)/X //units
12 eLy=(y2-y1)/X //units
13 eLz=(z2-z1)/X //units
14 Q=[eLx,eLy,eLz] //units
15 Z=P*Q' //units
16 //Result
17 clc
18 printf('The unit vector is:%fi+%fj+%fk \n',eLx,eLy,
        eLz)
19 printf('The projection of P is:%f',Z)
20 //Note:The final answer for the projection of P is
        off by 0.1 units
21 //The answer mentioned in the textbook is -1.41

```

Scilab code Exa 1.16 Vectors

```

1 //Initialization of variables
2 x1=2 //units
3 x2=5 //units
4 y1=-5 //units
5 y2=2 //units
6 z1=3 //units
7 z2=-4 //units
8 P=[10,-8,14] //units
9 //Calculations
10 X=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2) //units
11 eLx=(x2-x1)/X //units
12 eLy=(y2-y1)/X //units
13 eLz=(z2-z1)/X //units

```

```

14 Q=[eLx , eLy , eLz] // units
15 Z=P*Q' // units
16 // Result
17 clc
18 printf('The unit vector is : %fi+%fj%fk \n', eLx, eLy,
        eLz)
19 printf('The projection of P is : %f lb ', Z) //lb

```

Scilab code Exa 1.17 Vectors

```

1 Px=2.85 // ft
2 Py=4.67 // ft
3 Pz=-8.09 // ft
4 Qx=28.3 //lb
5 Qy=44.6 //lb
6 Qz=53.3 //lb
7 // Calculations
8 X=(Py*Qz-Pz*Qy) //N.m
9 Y=(Pz*Qx-Px*Qz) //N.m
10 Z=(Px*Qy-Py*Qx) //N.m
11 // Result
12 clc
13 printf('The cross product is : %fi%fj%fk lb-ft ', X, Y, Z)
        //lb-ft

```

Scilab code Exa 1.19 Vectors

```

1 // Initilization Of Variables
2 a=1 //Lower Limit of the Integral
3 b=3 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 // Calculation
6 //Using Trapezoidal Rule for Intergration

```



```

7 function [I1]=Trap_Composite1(f,a,b,n)
8     h=(b-a)/n
9     t=linspace(a,b,n+1)
10    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
11 endfunction
12 deff(' [y]=f(t) ', 'y=t^2 ')
13 function [I2]=Trap_Composite2(f1,a,b,n)
14     h=(b-a)/n
15     t=linspace(a,b,n+1)
16    I2=(h/2)*((2*sum(f1(t)))-f1(t(1))-f1(t(n+1)))
17 endfunction
18 deff(' [y1]=f1(t) ', 'y1=t*2 ')
19 z=b-a
20 //Result
21 clc
22 printf(' %fi+%fj%fk ',Trap_Composite1(f,a,b,n),
        Trap_Composite2(f1,a,b,n),-z)

```

Chapter 2

Operations With Forces

Scilab code Exa 2.1 OPF

```
1 //Initialization of variables
2 F=20 //lb
3 L=4.33 //ft
4 //Calculation
5 M=-F*L //lb-ft
6 //Result
7 clc
8 printf('The moment of force F about O is:%f lb-ft ',M
    )
```

Scilab code Exa 2.2 OPF

```
1 //Initialization of variables
2 F=20 //lb
3 theta=((60*%pi)/180) //radians
4 L=5 //ft
5 //Calculations
6 F_x=F*cos(theta) //Resolving the vector
```

```

7 F_y=F*sin(theta) //Resolving the vector
8 M=-F_y*L //Appling Varignon's theorem
9 //Negative sign tells that moment is clockwise
10 //Result
11 clc
12 printf('The moment of the force about O is:%f lb-ft ',
        ,M)

```

Scilab code Exa 2.3 OPF

```

1 //Initilization of variables
2 F=100 //N
3 x1=2 //m
4 x2=5 //m
5 y1=0 //m
6 y2=1 //m
7 z1=4 //m
8 z2=1 //m
9 //Calculations
10 xside=(x2-x1) //m
11 yside=(y2-y1) //m
12 zside=(z2-z1) //m
13 LD=sqrt(xside^2+yside^2+zside^2)
14 Fx=(xside/LD)*F //N
15 Fy=(yside/LD)*F //N
16 Fz=(zside/LD)*F //N
17 Mx=-Fy*z1 //N-m
18 My=Fx*x1-Fz*z1 //N-m
19 Mz=Fy*x1 //N-m
20 //Result
21 clc
22 printf('Fx is:%f N\n',Fx) //N
23 printf('Fy if:%f N\n',Fy) //N
24 printf('Fz is:%f N\n',Fz) //N
25 printf('Moment about X-Axis is:%f N.m\n',Mx) //N-m

```

```
26 printf('Moment about Y-Axis is :%f N.m\n',My) //N-m
27 printf('Moment about Z-Axis is :%f N.m',Mz) //N-m
```

Scilab code Exa 2.4 OPF

```
1 //Initialization of variables
2 Fx=68.7 //N
3 Fy=22.9 //N
4 Fz=-68.7 //N
5 rx=2 //m
6 ry=0 //m
7 rz=4 //m
8 rx1=5 //m
9 ry1=1 //m
10 rz1=1 //m
11 //Calculation
12 Mx=Fz*ry-Fy*rz //N-m
13 My=-(Fz*rx-Fx*rz) //N-m
14 Mz=Fy*rx-Fx*ry //N-m
15 Mx1=Fz*ry1-Fy*rz1 //N-m
16 My1=-(Fz*rx1-Fx*rz1) //N-m
17 Mz1=Fy*rx1-Fx*ry1 //N-m
18 //Result
19 clc
20 printf('Moment with respect to origin using point
      (2,0,4): %fi+%fj+%fk N.m\n',Mx,My,Mz) //N-m
21 printf('Moment with respect to origin using point
      (5,1,1): %fi+%fj+%fk N.m \n',Mx1,My1,Mz1) //N-m
```

Scilab code Exa 2.5 OPF

```
1 //Initialization of variables
2 Fx=2 //lb
```

```

3  Fy=3 //lb
4  Fz=-1 //lb
5  rx=1 //ft
6  ry=-4 //ft
7  rz=3 //ft
8  //Coordinates of points
9  ax=3 //ft
10 ay=1 //ft
11 az=1 //ft
12 bx=3 //ft
13 by=-1 //ft
14 bz=1 //ft
15 cx=2 //ft
16 cy=5 //ft
17 cz=-2 //ft
18 //Calculations
19 Rx=ax-cx //ft
20 Ry=ay-cy //ft
21 Rz=az-cz //ft
22 Mx=(Ry*Fz)-(Rz*Fy) //lb-ft
23 My=-((Rx*Fz)-(Rz*Fx)) //lb-ft
24 Mz=(Rx*Fy)-(Ry*Fx) //lb-ft
25 E_u=sqrt((bx-cx)^2+(by-cy)^2+(bz-cz)^2) //ft
26 ex=(bx-cx)/E_u //ft
27 ey=(by-cy)/E_u //ft
28 ez=(bz-cz)/E_u //ft
29 M_lx=Mx*ex //lb-ft
30 M_ly=My*ey //lb-ft
31 M_lz=Mz*ez //lb-ft
32 M_l=M_lx+M_ly+M_lz //lb-ft
33 //Result ]
34 clc
35 printf('Hence the moment about line is %f lb-ft \n',
        M_l)

```

Scilab code Exa 2.6 OPF

```
1 //Initilization of variables
2 P_x=22 //N
3 P_y=23 //N
4 P_z=7 //N
5 p1=1 //m
6 p2=-1 //m
7 p3=-2 //m
8 //Calculations
9 Mx=(p2*P_z)-(p3*P_y) //N-m
10 My=-((p1*P_z)-(p3*P_x)) //N-m
11 Mz=(p1*P_y)-(p2*P_x) //N-m
12 //Result
13 clc
14 printf('The moment about the line from the origin is
        : %fi% fj+%fk N.m', Mx, My, Mz) //N-m
```

Scilab code Exa 2.8 OPF

```
1 //Initilization of variables
2 F=10 //N Force couple
3 a=3 //m Moment arm
4 //Calculations
5 C=-F*a //N-m
6 //Result
7 clc
8 printf('The resultant couple is: %fN.m \n', C)
```

Scilab code Exa 2.11 OPF

```
1 //Initilization of variables
2 C1=20 //N-m
```

```

3 C2=40 //N-m
4 C3=-55 //N-m
5 // Calculations
6 C=sqrt(C1^2+C2^2+C3^2) //N-m
7 thetax=C2/C
8 thetay=C3/C
9 thetaz=C1/C
10 Cx=C*thetax //N-m
11 Cy=C*thetay //N-m
12 Cz=C*thetaz //N-m
13 // Result
14 clc
15 printf('Couple in vector notation: %fi%fj+%fk N.m',Cx
        ,Cy,Cz) //N-m

```

Scilab code Exa 2.12 OPF

```

1 // Initialization of variables
2 F1=25 //lb
3 F2=25 //lb
4 L1=14 //in
5 L2=20 //in
6 // Calculations
7 C=F1*L1 //lb-in
8 M=-F2*L2 //lb-in
9 // Result
10 clc
11 printf('The twisting couple is: %f lb-in\n',C) //lb-
    in
12 printf('The bending moment is: %f lb-in',M) //lb-in

```

Scilab code Exa 2.13 OPF

```

1 //Initilization of variables
2 rx=20 //in
3 ry=0 //in
4 rz=14 //in
5 Fx=0 //lb
6 Fy=-25 //lb
7 Fz=0 //lb
8 //Calculation
9 Mx=ry*Fz-rz*Fy //lb-in
10 My=rx*Fz-rz*Fx //lb-in
11 Mz=rx*Fy-ry*Fx //lb-in
12 //Result
13 clc
14 printf('The moment of the 25-lb force is:%fi+%fj%fk
        lb.in ',Mx,My,Mz) //lb-in

```

Scilab code Exa 2.14 OPF

```

1 //Initilization of variables
2 //Co-ordinates with respect to point O
3 x=17.9 //ft
4 y=6.91 //ft
5 z=46.3 //ft
6 Fz=-4000 //lb
7 Fy=0 //lb
8 //Calculation
9 Mx=y*Fz-z*Fy //lb-ft
10 //Result
11 clc
12 printf('The scalar coefficient of the i term is the
        moment about the X-Axis Mx:%f lb-ft lb-ft ',Mx) //
        lb-ft
13 //The answer in the text book is incorrect due to
        decimal point accuracy in scilab

```

Chapter 3

Resolution of Coplanar Force System

Scilab code Exa 3.1 Resolution

```
1 //Initialization of variables
2 F1=150 //lb
3 F2=200 //lb
4 F3=80 //lb
5 F4=180 //lb
6 theta1=((30*%pi)/180) //radians
7 theta2=((150*%pi)/180) //radians
8 theta3=((240*%pi)/180) //radians
9 theta4=((315*%pi)/180) //radians
10 //Calculations
11 F1x=F1*cos(theta1) //lb
12 F1y=F1*sin(theta1) //lb
13 F2x=F2*cos(theta2) //lb
14 F2y=F2*sin(theta2) //lb
15 F3x=F3*cos(theta3) //lb
16 F3y=F3*sin(theta3) //lb
17 F4x=F4*cos(theta4) //lb
18 F4y=F4*sin(theta4) //lb
19 Fx=F1x+F2x+F3x+F4x //lb
```

```

20 Fy=F1y+F2y+F3y+F4y //lb
21 R=sqrt(Fx^2+Fy^2) //lb
22 theta=((atan(Fy/Fx))*180)/%pi //degrees
23 theta_R=360+theta //degrees
24 //Result
25 clc
26 printf('The resultant of the force system is:%f lb\n
',R) //lb
27 printf('The resultant is at :%f degrees',theta_R) //
degrees

```

Scilab code Exa 3.2 Resolution

```

1 //Initilization of variables
2 F1=50 //N
3 F2=100 //N
4 F3=30 //N
5 //Calculation
6 //The book has a misprint for squareroot of 1^2
7 F1x=F1/sqrt(2) //N
8 F1y=F1/sqrt(2) //N
9 F2x=-(F2*3)/sqrt(10) //N
10 F2y=(-F2)/sqrt(10) //N
11 F3x=F3/sqrt(5) //N
12 F3y=(-F3*2)/sqrt(5) //N
13 Fx=F1x+F2x+F3x //N
14 Fy=F1y+F2y+F3y //N
15 R=sqrt(Fx^2+Fy^2) //N
16 theta=atan(Fy/Fx) //radians
17 theta_x=180+(theta*180)/%pi //degrees
18 //Result
19 clc
20 printf('The resultant is:%f N\n',R) //N
21 printf('The resultant makes an angle of:%f degrees',
theta_x) //degrees

```

Scilab code Exa 3.3 Resolution

```
1 //Initialization of variables
2 F1=70 //lb
3 F2=100 //lb
4 F3=125 //lb
5 theta1=0 //radians
6 theta2=((10*%pi)/180) //radians
7 theta3=((30*%pi)/180) //radians
8 //Calculations
9 Fx=F1-(F2*cos(theta3))-(125*sin(theta2)) //lb
10 Fy=125*cos(theta2)-(100*sin(theta3)) //lb
11 R=sqrt(Fx^2+Fy^2) //lb
12 theta=atan(Fy/Fx) //radians
13 theta_x=180+(theta*180)/%pi //degrees
14 //Result
15 clc
16 printf('The resultant of the force system is:%f lb\n',R) //lb
17 printf('The resultant is at %f with respect to the X
    -Axis degrees ',theta_x) //degrees
```

Scilab code Exa 3.4 Resolution

```
1 //Initialization of variables
2 F1=-20 //N
3 F2=30 //N
4 F3=5 //N
5 F4=-40 //N
6 //Distances with respect to point O
7 x1=6 //m
```

```

8 x2=0 //m
9 x3=8 //m
10 x4=13 //m
11 // Calculations
12 R=F1+F2+F3+F4 //N
13 //Applying moment about point O equal to zero
14 M_0=-(F1*x1)+(F2*x2)+(F3*x3)+(F4*x4) //N-m
15 //Applying moment about point O equal to R*x
16 x=M_0/R //m
17 //Result
18 clc
19 printf('The resultant of force system is:%f N\n',R)
    //N
20 printf('The moment about point O is:%f N-m\n',M_0)
    //N-m
21 printf('The resultant of moment acts at %f meters
    from point O',x) //m

```

Scilab code Exa 3.5 Resolution

```

1 // Initialization of variables
2 F1=-100 //lb
3 F2=200 //lb
4 F3=-200 //lb
5 F4=400 //lb
6 F5=-300 //lb
7 //Distance with respect to point O
8 x1=0 //ft
9 x2=2 //ft
10 x3=5 //ft
11 x4=9 //ft
12 x5=11 //ft
13 // Calculation
14 R=F1+F2+F3+F4+F5 //lb
15 M_0=(F1*x1)+(F2*x2)+(F3*x3)+(F4*x4)+(F5*x5) //N-m

```

```

16 //Result
17 clc
18 printf('The resultant of the force system is:%i lb \
      n',R) //lb
19 printf('The moment about point O is:%i lb-ft ',M_0)
      //lb-ft

```

Scilab code Exa 3.6 Resolution

```

1 //Initilization of variables
2 F1=20 //lb
3 F2=20 //lb
4 F3=-40 //lb
5 //Distance from point O
6 x1=3 //ft
7 x2=3 //ft
8 //Calculations
9 R=F1+F2+F3 //lb
10 M_0=-(F1*x1)+(F2*x2) //lb-ft
11 //Results
12 clc
13 printf('The resultant of the force system is:%i lb\n
      ',R) //lb
14 printf('The Moment about point O is:%i lb-ft ',M_0)
      //lb-ft

```

Scilab code Exa 3.7 Resolution

```

1 //Initilization of variables
2 F1=500 //N
3 F2=-400 //N
4 F3=-200 //N
5 C=1500 //N-m

```

```

6 //Distance from point O
7 x1=2 //m
8 x2=4 //m
9 x3=6 //m
10 // Calculations
11 R=F1+F2+F3 //N
12 M_0=(F1*x1)+(F2*x2)+(F3*x3)+C //N-m
13 //Applying Varignons theorem
14 x=M_0/R //m
15 //Result
16 clc
17 printf('The resultant of the force system is:%i N\n',
        ,R) //N
18 printf('The moment about point O is:%i N-m\n',M_0)
        //N-m
19 printf('The resultant acts at %i meters from point O
        m',x) //m

```

Scilab code Exa 3.8 Resolution

```

1 //Initilization of variables
2 F1=50 //lb
3 F2=100 //lb
4 theta1=((45*%pi)/180) //radians
5 //Distance from point O
6 x1=5 //ft
7 x2=4 //ft
8 // Calculation
9 F_x=F1-(F2*cos(theta1)) //lb
10 F_y=F1-(F2*sin(theta1)) //lb
11 R=sqrt(F_x^2+F_y^2) //lb
12 M_0=F1*x1-(x2*F1) //lb-ft
13 //Applying Varignons Theorem
14 x=M_0/R //ft
15 //Result

```

```

16 clc
17 printf('The resultant of the force system is:%f lb\n
    ',R) //lb
18 printf('The Moment about point O is:%f lb-ft\n',M_0)
    //lb-ft
19 printf('The Resultant acts at %f feet from point O
    ft ',x) //ft

```

Scilab code Exa 3.9 Resolution

```

1 //Initialization of variables
2 A=80 //N
3 B=120 //N
4 C=100 //N
5 D=50 //N
6 thetaA=((90*%pi)/180) //radians
7 thetaB=((150*%pi)/180) //radians
8 thetaC=((45*%pi)/180) //radians
9 thetaD=((340*%pi)/180) //radians
10 //Calculations
11 Ax=A*cos(thetaA) //N
12 Ay=A*sin(thetaA) //N
13 Bx=B*cos(thetaB) //N
14 By=B*sin(thetaB) //N
15 Cx=C*cos(thetaC) //N
16 Cy=C*sin(thetaC) //N
17 Dx=D*cos(thetaD) //N
18 Dy=D*sin(thetaD) //N
19 M_Ax=0 //N-m
20 M_Ay=0 //N-m
21 M_Bx=-Bx*5 //N-m
22 M_By=By*8 //N-m
23 M_Cx=-Cx*1 //N-m
24 M_Cy=Cy*1 //N-m
25 M_Dx=-Dx*-1 //N-m

```

```

26 M_Dy=Dy*8 //N-m
27 Fx=Ax+Bx+Cx+Dx //N
28 Fy=Ay+By+Cy+Dy //N
29 R=sqrt(Fx^2+Fy^2) //N
30 M_0=M_Dx+M_Dy+M_Cx+M_Cy+M_Bx+M_By+M_Ax+M_Ay //N-m
31 theta=atan(Fy/Fx) //radians
32 theta_x=(theta*180)/%pi //degrees
33 //Applying Varignons theorem
34 x=M_0/R //m
35 //Result
36 clc
37 printf('The resultant of the force system is:%f N\n',
        ,R) //N
38 printf('The moment about point O is:%f N\n',M_0) //N
39 printf('The resultant acts at and angle of %f
        degrees with respect to X-Axis degrees\n',theta_x
        ) //degrees
40 printf('The resultant of the force system acts at %f
        meters from point O',x) //m

```

Scilab code Exa 3.10 Resolution

```

1 //Initialization of variables
2 F1=100 //lb
3 F2=80 //lb
4 F3=120 //lb
5 F4=150 //lb
6 theta1=((60*%pi)/180) //radians
7 theta2=((45*%pi)/180) //radians
8 theta3=((90*%pi)/180) //radians
9 theta4=((75*%pi)/180) //radians
10 //Distance from point O
11 x1=-5 //ft
12 y1=20 //ft
13 x2=10 //ft

```



```

14 y2=10 //ft
15 x3=25 //ft
16 y3=25 //ft
17 x4=35 //ft
18 y4=15 //ft
19 //Calculations
20 Fx=F1*cos(theta1)+F2*cos(theta2)+F4*cos(theta4) //lb
21 Fy=-F1*sin(theta1)+F2*sin(theta2)-F3-F4*sin(theta4)
    //lb
22 R=sqrt(Fx^2+Fy^2) //lb
23 theta=atan(Fy/Fx) //radians
24 theta_x=(theta*180)/%pi //degrees
25 M_0=-(F1*cos(theta1)*y1)+(-x1)*(F1*sin(theta1))-
    (x2*(F2*cos(theta2))+(y2)*(F2*sin(theta2))-(x3*F3)-
    (y4*F4*cos(theta4))-(x4*F4*sin(theta4))) //lb-ft
26 //Applying varignons theorem
27 x=M_0/Fy //ft
28 y=-M_0/Fx //ft
29 //Results
30 clc
31 printf('The resultant of the force system is:%f lb\n
    ',R) //lb
32 printf('The resultant acts at %f degrees with
    respect to X-Axis \n',theta_x) //degrees
33 printf('The moment about point O is:%f lb-ft \n',M_0
    ) //lb-ft
34 printf('The x intercept of resultant is:%f ft\n',x)
    //ft
35 printf('The y intercept of resultant is:%f ft\n',y)
    //ft
36 //Answer for angle should be negative which has not
    been mentioned in the tectbook but a schematic
    shows the angle in fourth quadrant to clarify the
    doubt

```

Scilab code Exa 3.11 Resolution

```
1 //Initialization of variables
2 F1=150 //lb
3 F2=80 //lb
4 F3=100 //lb
5 F4=50 //lb
6 theta1=((45*%pi)/180) //radians
7 r=3 //units
8 //Calculations
9 Fh=F1-F3*cos(theta1) //lb
10 Fv=F4-F2-F3*sin(theta1) //lb
11 R=sqrt(Fh^2+Fv^2) //lb
12 //Applying the Varignons Theorem
13 a=(F4*r-F1*r+F2*r-F3*r)/R //units
14 //Result
15 clc
16 printf('The resultant of the force system is:%f lb \
n',R) //lb
17 printf('The resultant acts at %f units from the
point O',a) //units
18 //Negative sign indicates a negative moment caused
by the resultant
```

Scilab code Exa 3.12 Resolution

```
1 //Initialization of variables
2 F1=150 //lb
3 F2=200 //lb
4 F3=200 //lb
5 F4=225 //lb
6 M=900 //lb-ft
7 Theta1=(45*%pi)/180 //radians
8 Theta2=(30*%pi)/180 //radians
9 x1=3 //ft
```

```

10 x2=15 //ft
11 x3=12 //ft
12 x4=6 //ft
13 //Calculations
14 Fx=F1*cos(Theta1)+F2-F4*cos(Theta2) //Applying sum
    of all forces equal to zero in X direction
15 Fy=F1*sin(Theta1)-F4*sin(Theta2)+F2 //Applying sum
    of all forces equal to zero in Y direction
16 R=sqrt(Fx^2+Fy^2) //lb
17 theta=atand(Fy/Fx) //degrees
18 M_o=x1*F2-x2*F1*cos(Theta1)+x3*F1*sin(Theta1)-x4*F2+
    M+x4*F4*cos(Theta2)-x1*F4*sin(Theta2) //Moment
    about point O
19 x=M_o/Fy //Varignons Theorem
20 //Result
21 clc
22 printf('The x intercept of resultant position is %f\
    n',x)
23 printf('The Resultant is %f lb and acts at an angle
    of %f degrees',R,theta)

```

Scilab code Exa 3.13 Resolution

```

1 //Initilization Of Variables
2 a=0 //Lower Limit of the Integral
3 b=6 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 l=20 //lb/ft
6 //Calculation
7 //Using Trapezoidal Rule for Intergration
8 R=(b-a)*l
9 function [I2]=Trap_Composite2(f1,a,b,n)
10     h=(b-a)/n
11     t=linspace(a,b,n+1)
12     I2=(h/2)*((2*sum(f1(t)))-f1(t(1))-f1(t(n+1)))

```

```

13 endfunction
14 deff(' [y1]=f1 (t) ', 'y1=20*t ')
15 d=Trap_Composite2(f1,a,b,n)/R //ft
16 //Result
17 clc
18 printf('The value of R is:%i lb\n',R)
19 printf('The value of d is:%i ft ',d)

```

Scilab code Exa 3.14 Resolution

```

1 //Initilization Of Variables
2 a=0 //Lower Limit of the Integral
3 b=9 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 //Calculation
6 //Using Trapezoidal Rule for Intergration
7 function [I1]=Trap_Composite1(f,a,b,n)
8     h=(b-a)/n
9     t=linspace(a,b,n+1)
10    I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
11 endfunction
12 deff(' [y]=f(t) ', 'y=(t/9)*30 ') //y defined as a
    function of t and not x
13 function [I2]=Trap_Composite2(f1,a,b,n)
14     h=(b-a)/n
15     t=linspace(a,b,n+1)
16     I2=(h/2)*((2*sum(f1(t)))-f1(t(1))-f1(t(n+1)))
17 endfunction
18 deff(' [y1]=f1 (t) ', 'y1=(t^2/9)*30 ')
19 d=Trap_Composite2(f1,a,b,n)/Trap_Composite1(f,a,b,n)
    //m
20 //Result
21 clc
22 printf('The value of d is:%f m\n',d) //m
23 printf('The value of R is %f N',Trap_Composite1(f,a,

```

b,n)) //N

Chapter 4

Resolution of Non Coplanar Force System

Scilab code Exa 4.1 Resolution Non Coplanar

```
1 //Initialization Of Variables
2 F1=20 //lb
3 F2=15 //lb
4 F3=30 //lb
5 F4=50 //lb
6 //Co-ordinates of Forces
7 C1=[2;1;6]
8 C2=[4;-2;5]
9 C3=[-3;-2;1]
10 C4=[5;1;-2]
11 //Calculations
12 A=sqrt((C1(1,1))^2+(C1(2,1))^2+(C1(3,1)^2))
13 B=sqrt((C2(1,1))^2+(C2(2,1))^2+(C2(3,1)^2))
14 C=sqrt((C3(1,1))^2+(C3(2,1))^2+(C3(3,1)^2))
15 D=sqrt((C4(1,1))^2+(C4(2,1))^2+(C4(3,1)^2))
16 //Calculations for cos(thetax),cos(thetay) and cos(
    thetaz)
17 theta1=[C1(1,1)/A;C1(2,1)/A;C1(3,1)/A]
18 theta2=[C2(1,1)/B;C2(2,1)/B;C2(3,1)/B]
```

```

19 theta3=[C3(1,1)/C;C3(2,1)/C;C3(3,1)/C]
20 theta4=[C4(1,1)/D;C4(2,1)/D;C4(3,1)/D]
21 //Calculations for forces (in form of force vectors)
22 Fa=F1*theta1 //lb
23 Fb=F2*theta2 //lb
24 Fc=F3*theta3 //lb
25 Fd=F4*theta4 //lb
26 Fx=Fa(1,1)+Fb(1,1)+Fc(1,1)+Fd(1,1) //lb
27 Fy=Fa(2,1)+Fb(2,1)+Fc(2,1)+Fd(2,1) //lb
28 Fz=Fa(3,1)+Fb(3,1)+Fc(3,1)+Fd(3,1) //lb
29 R=sqrt(Fx^2+Fy^2+Fz^2) //lb
30 thetax=acosd(Fx/R) //degrees
31 thetay=180-acosd(Fy/R) //degrees
32 thetaz=acosd(Fz/R) //degrees
33 //Result
34 clc
35 printf('The resultant of the force system is %f lb \
n',R)
36 printf('The angle of the resultant with respect to x
y and z axes are %f, %f and %f degrees
respectively ',thetax,thetay,thetaz)

```

Scilab code Exa 4.2 Resolution Non Coplanar

```

1 //Initilization of variables
2 F=[20;-10;30] //N
3 //co-ordinates in meters
4 a=2 //m
5 b=4 //m
6 c=7 //m
7 d=3 //m
8 e=2 //m
9 f=4 //m
10 //Calculations
11 R=F(1,1)+F(2,1)+F(3,1) //N

```

```

12 M_o=F(1,1)*a+F(2,1)*b+F(3,1)*c //N-m
13 x=M_o/R //m
14 M_x=-F(3,1)*e-F(1,1)*d+F(2,1)*f //N-m
15 z=-M_x/R //m
16 //Result
17 clc
18 printf('The resultant is %f N \n',R)
19 printf('The moment about point O is %f N-m \n',M_o)
20 printf('The position of R is at %f from origin m \n',
,x)
21 printf('The moment is %f N-m\n',M_x)
22 printf('The z co-ordinate is %f m',z)

```

Scilab code Exa 4.3 Resolution Non Coplanar

```

1 //Initilization of variables
2 F=[100;50;-150] //Force vector N
3 a=2 //m
4 b=2 //m
5 c=3 //m
6 d=2 //m
7 e=4 //m
8 f=8 //m
9 //Calculations
10 R=F(1,1)+F(2,1)+F(3,1) //N
11 M_x=-F(1,1)*a+F(2,1)*b-F(3,1)*c //N-m
12 M_z=F(1,1)*d+F(2,1)*e+F(3,1)*f //N-m
13 C=sqrt(M_x^2+M_z^2) //N-m
14 thetax=atand(-M_x/M_z) //degrees
15 //result
16 clc
17 printf('The resultant is %f N \n',R)
18 printf('The moment about x axis is %f N.m \n',M_x)
19 printf('The moment about z axis is %f N.m\n',M_z)
20 printf('The couple acting is %f N.m\n',C)

```



```
21 printf('The trace makes an angle with x axis of %f
degrees ',thetax)
```

Scilab code Exa 4.4 Resolution Non Coplanar

```
1 //Initialization of variables
2 x1=-2
3 y1=2
4 z1=-2
5 x2=3
6 y2=0
7 z2=-4
8 x3=3
9 y3=2
10 z3=2
11 F1=40 //lb
12 F2=30 //lb
13 F3=20 //lb
14 Mxm=[-92.4,-48,-19.4]
15 Mym=[-46.2,72,9.8]
16 Mzm=[46.2,-36,19.4]
17 //Calculations
18 mag1=sqrt(x1^2+y1^2+z1^2)
19 mag2=sqrt(x2^2+y2^2+z2^2)
20 mag3=sqrt(x3^2+y3^2+z3^2)
21 thetax1=acosd(x1/mag1) //degrees
22 thetay1=acosd(y1/mag1) //degrees
23 thetaz1=acosd(z1/mag1) //degrees
24 thetax2=acosd(x2/mag2) //degrees
25 thetay2=acosd(y2/mag2) //degrees
26 thetaz2=acosd(z2/mag2) //degrees
27 thetax3=acosd(x3/mag3) //degrees
28 thetay3=acosd(y3/mag3) //degrees
29 thetaz3=acosd(z3/mag3) //degrees
30 //Now we will define all the components in terms of
```

```

    matrices for simplicity of computation
31 F=[F1 ,F2 ,F3] //lb
32 COSthetax=[cosd(thetax1);cosd(thetax2);cosd(thetax3)
    ]
33 COSthetay=[cosd(thetay1);cosd(thetay2);cosd(thetay3)
    ]
34 COSthetaz=[cosd(thetaz1);cosd(thetaz2);cosd(thetaz3)
    ]
35 Fx=F*COSthetax //lb
36 Fy=F*COSthetay //lb
37 Fz=F*COSthetaz //lb
38 R=sqrt(Fx^2+Fy^2+Fz^2) //lb
39 thetax=acosd(Fx/R) //degrees
40 thetay=acosd(Fy/R) //degrees
41 thetaz=acosd(Fz/R) //degrees
42 //Moment calculations
43 Mx=Mxm(1)+Mxm(2)+Mxm(3) //lb-ft
44 My=Mym(1)+Mym(2)+Mym(3) //lb-ft
45 Mz=Mzm(1)+Mzm(2)+Mzm(3) //lb-ft
46 C=sqrt(Mx^2+My^2+Mz^2) //lb-ft
47 //Direction cosines
48 PHIx=acosd(Mx/C) //degrees
49 PHIy=acosd(My/C) //degrees
50 PHIz=acosd(Mz/C) //degrees
51 //Result
52 clc
53 printf('The result of the force is %f lb\n',R)
54 printf('The angles with respect to X-Axis,Y-Axis and
    Z-axis are %f ,%f and %f degrees respectively\n'
    ,thetax,thetay,thetaz)
55 printf('The magnitude of resultant couple is %f lb-
    ft\n',C)
56 printf('The angles are as follows Cosphix=%f degrees
    ,Cosphiy=%f degrees and Cosphiz=%f degrees ',PHIx
    ,PHIy,PHIz)

```

Scilab code Exa 4.5 Resolution Non Coplanar

```
1 //Initialization of variables
2 F=[150;90;160] //lb force vector kind of decleration
3 //Co-ordinates defined as [x;y;z] all the co-
   ordinates are in feet
4 C_1=[2;0;0]
5 C_2=[0;0;1]
6 C_3=[0;-2;-1]
7 C_4=[-1;0;-1]
8 //Calculations
9 A=C_2-C_1
10 B=C_4-C_3
11 F1=(F(1,1)*A)/sqrt(A(1,1)^2+A(2,1)^2+A(3,1))
12 F2=(F(2,1)*B)/sqrt(B(1,1)^2+B(2,1)^2+B(3,1))
13 R=F1+F2
14 //The calculations for this is done differently
15 C1x=det([1 0 0;C_1(1,1) C_1(2,1) C_1(3,1);F1(1,1) F1
   (2,1) F1(3,1)])
16 C1y=det([0 1 0;C_1(1,1) C_1(2,1) C_1(3,1);F1(1,1) F1
   (2,1) F1(3,1)])
17 C1z=det([0 0 1;C_1(1,1) C_1(2,1) C_1(3,1);F1(1,1) F1
   (2,1) F1(3,1)])
18 C2x=det([1 0 0;C_3(1,1) C_3(2,1) C_3(3,1);F2(1,1) F2
   (2,1) F2(3,1)])
19 C2y=det([0 1 0;C_3(1,1) C_3(2,1) C_3(3,1);F2(1,1) F2
   (2,1) F2(3,1)])
20 C2z=det([0 0 1;C_3(1,1) C_3(2,1) C_3(3,1);F2(1,1) F2
   (2,1) F2(3,1)])
21 C3z=[0;0;F(3,1)]
22 sC1=[C1x;C1y;C1z]
23 sC2=[C2x;C2y;C2z]
24 C=sC1+sC2+C3z
25 //Result
```

```
26 clc
27 printf('The resultant force couple is %fi%fj+%fk lb-  
ft ',C(1,1), C(2,1), C(3,1))
```

Chapter 5

Equilibrium of Coplanar Force System

Scilab code Exa 5.1 Equilibrium of CFS

```
1 //Initialization of variables
2 D=[6/sqrt(40) -4/sqrt(20);2/sqrt(40) 2/sqrt(20)]
3 B=[0;25] //lb
4 //Calculations
5 X=inv(D)*B
6 //Result
7 clc
8 printf('The tension in cable AB is %flb and the
   tension in cable AC is %f lb',X(2),X(1))
```

Scilab code Exa 5.2 Equilibrium of CFS

```
1 //Initialization of variables
2 F1=100 //lb
3 R=16 //in
4 //Calculations
```

```

5 theta=asind(14/16) //degrees
6 N=100/sind(theta) //lb
7 P=N*cosd(theta) //lb
8 //Result
9 clc
10 printf('The value of normal reaction offered is %flb
        and the push required is %f lb',N,P)

```

Scilab code Exa 5.3 Equilibrium of CFS

```

1 //Initilization of variables
2 L=20 //m
3 M=1200 //kg
4 g=9.81 //m/s^2
5 H=10 //m
6 //Calculations
7 AB=sqrt(L^2-H^2) //Applying Pythagoras Theorem
8 costheta=17.3/20
9 F1=M*g*H/AB //N
10 F2=M*g/costheta //N
11 //Result
12 clc
13 printf('Force F1 is %f N and Force F2 is %f N',F1,F2
        )
14 //Decimal accuracy causes discrepancy in answers
        compared to the textbook answers

```

Scilab code Exa 5.4 Equilibrium of CFS

```

1 //Initilization of variables
2 Fx=1000 //lb
3 Fy=1000 //lb
4 costheta=9/15

```

```

5 cosbeta=12/15
6 sintheta=4/5
7 sinbeta=3/5
8 // Calculations
9 // Matrix solution
10 A=[costheta -cosbeta;sintheta sinbeta]
11 B=[-1000;1000]
12 X=inv(A)*B
13 // Result
14 clc
15 printf('Thus force in AB is %i lb compression and BC
        has %i lb compression ',X(1),X(2))

```

Scilab code Exa 5.5 Equilibrium of CFS

```

1 // Initialization of variables
2 w=10 //lb/ft
3 L=12 //ft
4 theta=30 //degrees
5 // Calculation
6 // Matrix Calculations
7 A=[cosd(30) -cosd(30);sind(30) sind(30)]
8 B=[0;120]
9 X=inv(A)*B
10 // Result
11 clc
12 printf('The tension in the cable is %i lb and the
        reaction at B is %i lb ', X(1) ,X(2))

```

Scilab code Exa 5.6 Equilibrium of CFS

```

1 // Initialization of variables
2 W1=40 //lb

```

```

3 W2=30 //lb
4 theta1=30 //degrees
5 // Calculations
6 //Summing the forces parallel to 30 degree plane
7 T=40*sind(theta1)
8 theta=asind(T/W2)
9 //Result
10 clc
11 printf('The tension in the cable is %flb and the
        angle theta is %f degrees ',T,theta)

```

Scilab code Exa 5.8 Equilibrium of CFS

```

1 //Initilization of variables
2 F1=125 //N
3 F2=200 //N
4 F3=340 //N
5 F4=180 //N
6 x1=4 //m
7 x2=3 //m
8 x3=10 //m
9 x4=15 //m
10 x5=17 //m
11 // Calculations
12 Rb=(-F1*x1+F2*x2+F3*x3+F4*x4)/x5 //moment about point
    A
13 Ra=(F1*(x1+x5)+F3*(x5-x3)+F2*(x5-x2)+F4*(x5-x4))/x5
    //moment about point B
14 //Result
15 clc
16 printf('The reaction at A is %fN and reaction at B
        is %fN ',Ra,Rb)

```

Scilab code Exa 5.9 Equilibrium of CFS

```
1 //Initialization of variables
2 F1=1000 //lb
3 F2=1200 //lb
4 F3=2000 //lb
5 x1=1 //ft
6 x2=7 //ft
7 x4=2 //ft
8 x3=6 //ft
9 //Calculation
10 //Equilibrium equations
11 Rn=(F3*(x1+x2+x3)+F2*(x1+x2)+F1*x1)/(x1+x3+x2+x4) //
    Moment about point M
12 Rm=(F1*(x2+x3+x4)+F2*(x3+x4)+F3*x4)/(x1+x2+x3+x4) //
    Moment about point N
13 //Result
14 clc
15 printf('The reaction at M is %flb and reaction at N
    is %flb ',Rm,Rn)
16 //Decimal Accuracy causes discrepancy in answers
    between computation and textbook
```

Scilab code Exa 5.10 Equilibrium of CFS

```
1 //Initialization of variables
2 P=10 //kg
3 g=9.81 //m/s^2
4 //Calculations
5 //equilibrium at fig b
6 T1=P*g/2 //N
7 //equilibrium at fig c
8 T2=T1/2 //N
9 //equilibrium at fig d
10 P=T2
```

```
11 //Result
12 clc
13 printf('The force P is %fN',P)
```

Scilab code Exa 5.11 Equilibrium of CFS

```
1 //Initilization of variables
2 k=20 //lb/in
3 w=20 //lb/ft
4 x1=4 //ft
5 x2=10 //ft
6 x3=8 //ft
7 x4=6 //ft
8 x5=9 //ft
9 F1=1920 //lb.rad
10 F2=3360 //lb.rad
11 //calculations
12 theta=(w*x2*x5)/(F1*x3+F2*(x3+x4)) //radians
13 FB=F1*theta
14 FC=F2*theta
15 A=(w*x2)-FB-FC
16 //Result
17 clc
18 printf('The force in spring B is %flb and spring C
        is %f lb and the reaction at A is %f lb',FB,FC,A)
```

Scilab code Exa 5.12 Equilibrium of CFS

```
1 //Initilization of variables
2 L=3.8 //m
3 w=10 //kg/m
4 P=1000 //N
5 t=0.8 //m
```

```

6 g=9.81 //m/s^2
7 //Calculations
8 Gf=L*w*g //N
9 A=(P*L+Gf*L*0.5)/t //N Taking moment about point B
10 B=(P*(L-t)+Gf*(0.5*L-t))/t //N Taking moment about
    point A
11 //Result
12 clc
13 printf('The reaction at point A and B are %f N and
    %f N respectively ',A,B)
14 //Decimal accuracy causes discrepancy in answers
    compared to the textbook

```

Scilab code Exa 5.13 Equilibrium of CFS

```

1 //Initilization of variables
2 Wa=400 //lb
3 Wb=200 //lb
4 theta=30 //degrees
5 //Calculations
6 Ta=Wa*sind(theta) //lb
7 Tb=Wb*sind(theta) //lb
8 //Taking moment about point O
9 P=(Tb*12+Ta*6)/24 //lb
10 //Result
11 clc
12 printf('The value of Ta is %f lb and that of Tb is
    %f lb ,also P is %f lb ',Ta,Tb,P)

```

Scilab code Exa 5.15 Equilibrium of CFS

```

1 //Initilization of variables

```

```

2 F=[5;2;3;1.5] //kN Forces are defined as a column
   matrix
3 theta=[90;60;45;80] //degrees angles are also
   defined as a column matrix
4 d=[2;6;13;17] //distances from point C of each force
5 c=[17;15;11;4] //distance form point D of each force
6 //Calculations
7 //Summing horizontal forces
8 Ch=-F(3,1)*cosd(theta(3,1))+F(2,1)*cosd(theta(2,1))+
   F(4,1)*cosd(theta(4,1)) //kN "which indidcates
   that Ch acts to the left instead of the assumed"
9 //Taking moment about point C
10 D=(F(1,1)*d(1,1)+F(2,1)*sind(theta(2,1))*d(2,1)+F
   (3,1)*sind(theta(3,1))*d(3,1)+F(4,1)*sind(theta
   (4,1))*d(4,1))/d(4,1) //kN
11 //Taking moment about point D
12 Cv=(F(1,1)*c(2,1)+F(2,1)*sind(theta(2,1))*c(3,1)+F
   (3,1)*sind(theta(3,1))*c(4,1))/c(1,1) //kN
13 //Result
14 clc
15 printf('The values of Ch,D and Cv are %f kN,%f kN
   and %f kN respectively ',Ch,D,Cv)

```

Scilab code Exa 5.16 Equilibrium of CFS

```

1 //Initilization of variables
2 w=100 //N/m
3 F1=200 //N
4 M=500 //N.m
5 Lw=2 //m
6 //Distance from point A
7 d=[1;2;3;4;5] //m
8 //Distance from point B
9 b=[5;4;3;2;1] //m
10 //Calculations

```

```

11 //Taking moment about point A
12 Ra=(w*Lw*d(1,1)+F1*d(3,1)-M)/d(4,1) //N
13 //Taking moment about point B
14 Rb=(w*Lw*b(3,1)+F1*b(5,1)+M)/b(2,1) //N
15 //Result
16 clc
17 printf('The value of reaction at A and B are %f N
        and %f N respectively ',Ra,Rb)

```

Scilab code Exa 5.17 Equilibrium of CFS

```

1 //Initilization of variables
2 L=14 //feet
3 W=18 //lb
4 theta1=60 //degrees
5 theta2=30 //degrees
6 L1=10 //ft
7 //Calculations
8 //Taking moment about point B
9 Rd=(W*(L/2)*cosd(theta1)*cosd(theta2))/L1 //lb
10 //Summing all the forces in the horizontal direction
11 T=Rd*cosd(theta2) //lb
12 //Result
13 clc
14 printf('The value of Rd and T is %f lb and %f lb
        respectively ',Rd,T)

```

Scilab code Exa 5.18 Equilibrium of CFS

```

1 //Initilization of variables
2 theta=[60;60;45] //degrees theta has been defined in
        the form of a matrix
3 d=[4.46;3.54;2] //feet defined as a matrix

```

```

4 F=400 //lb
5 //Calculations
6 //Taking moment about point A
7 Re=(F*(8-d(2,1)))/8 //lb
8 Ra=400-Re //lb here i have used the summation of
    forces in the vertical direction
9 //Taking moment about point B
10 Dv=(-F*3.644)/5.77 //lb
11 //Taking moment about point D
12 Bv=(F*2.126)/5.77 //lb
13 //Taking summation of forces in the vertical
    direction
14 Cv=-223-Dv //lb
15 //Taking moment about point D
16 Ch=(223*d(3,1)*cosd(theta(3,1))-Cv*5.173*cosd(theta
    (3,1)))/(5.173*sind(theta(3,1))) //lb
17 //Taking summation of forces in the horizontal
    direction
18 Dh=-Ch //lb
19 //Taking sum of forces in horizontal direction
20 Bh=-Dh //lb
21 //Result
22 clc
23 printf('The Floor reactions are \n')
24 printf('Ra=%f lb and Re=%f lb\n',Ra,Re)
25 printf('Pin reaction at C on CE are\n')
26 printf('Ch=%f lb and Cv=%f lb\n',Ch,Cv)
27 printf('The pin reactions at B on AC are\n')
28 printf('Bh=%f lb and Bv=%f lb ',Bh,Bv)

```

Scilab code Exa 5.19 Equilibrium of CFS

```

1 //Initialization of variables
2 r=0.5 //m
3 m=10 //kg

```

```
4 g=9.81 //m/s^2
5 theta=60 //degrees
6 //Calculations
7 //Due to symmetry the reaction will be shared by the
  structure
8 A=m*g*0.5 //N
9 B=A //N
10 //Vertical forces summed
11 N1=m*g/(2*sind(theta/2)) //N
12 //Taking moment about point C
13 T=(N1*0.866+B*sind(theta*0.5))/(1.5*cosd(theta*0.5))
14 //Result
15 clc
16 printf('The value of N1 and T are %f N and %f N
  respectively ',N1,T)
```

Chapter 6

Equilibrium of Non Coplanar Force System

Scilab code Exa 6.1 Equilibrium of NCFS

```
1 //Initialization of variables
2 H=30 //ft
3 F=150 //lb
4 theta1=10 //degrees
5 theta2=30 //degrees
6 theta3=60 //degrees
7 //Calculations
8 //Matrix solution of simultaneous equations
9 X=[cosd(theta3)*sind(theta2) -cosd(theta3)*sind(
    theta2);cosd(theta3)*cosd(theta2) cosd(theta3)*
    cosd(theta2)]
10 Y=[0;F*cosd(theta1)]
11 R=inv(X)*Y
12 //To find P,sum the forces vertically along the y-
    axis
13 P=F*sind(theta1)+2*R(1,1)*sind(theta3) //lb
    Copression
14 //Result
15 clc
```



```
16 printf('The value of A and B is %f lb and that of P
    is %f lb ',R(1,1),P)
```

Scilab code Exa 6.2 Equilibrium of NCFS

```
1 //Initialization of variables
2 F=150 //lb
3 theta1=10 //degrees
4 theta2=30 //degrees
5 theta3=60 //degrees
6 //Calculations
7 A=[-cosd(theta3)*cosd(theta2);-sind(theta3);cosd(
    theta3)*sind(theta2)]
8 B=[-cosd(theta3)*cosd(theta2);-sind(theta3);cosd(
    theta3)*sind(theta2)]
9 //150lb force is actually a vector
10 F_v=[F*cosd(theta1);F*sind(theta1);0] //lb
11 //Position vector relative to C
12 r=[0;30;0]
13 //Moment about point C is zero
14 //solution by matrix
15 X=[7.5 -7.5;13 13]
16 Y=[0;4470]
17 R=inv(X)*Y
18 A=R(1,1) //lb
19 B=R(2,1) //lb
20 //Summing forces in y direction
21 Cy=0.866*A+0.866*B+25.9 //lb
22 //Result
23 clc
24 printf('The value of A and B is %f lb and that of Cy
    is %f lb ',A,Cy)
```

Scilab code Exa 6.3 Equilibrium of NCFS

```
1 // Initilization of variables
2 m=6.12 //kg
3 g=9.81 //m/s^2
4 x1=3 //m
5 x2=4 //m
6 y=6 //m
7 z1=2 //m
8 z2=4 //m
9 AB=5
10 // Calculations
11 AD=sqrt(x1^2+y^2+z1^2)
12 AC=sqrt(x2^2+z1^2)
13 //Sum of forces in the y direction
14 T1=(m*g*AD)/6 //N
15 //sum of forces in the x and z direction
16 //Matrix solution of the following simultaneous
    equations
17 X=[x2/AC, -x1/AB;-z1/AC, z2/AB]
18 Y=[T1*(x1/AD);T1*(z1/AD)]
19 R=inv(X)*Y
20 T2=R(1) //N
21 T3=R(2) //N
22 //Result
23 clc
24 printf('The values of T1,T2 and T3 are %f N,%f N and
    %f N respectively ',T1,T2,T3)
```

Scilab code Exa 6.4 Equilibrium of NCFS

```
1 //Intilization of variables
2 F=[0;60;0] //Force defined as a matrix
3 t1=[-3/7;6/7;-2/7] //Tension defined as a matrix
4 t2=[4/4.47;0;-2/4.47] //tension defined as a mtrix
```

```

5 t3=[-3/5;0;4/5] //Tension defined as a matrix
6 //Calculations
7 //Summation of forces in the y-direction
8 T1=F(2,1)/t1(2,1) //N
9 //Summation of forces in the x-direction and z
  direction
10 M1=[t2(1,1) t3(1,1);t2(3,1) t3(3,1)]
11 M2=-1*[t1(1,1)*T1;t1(3,1)*T1]
12 R=inv(M1)*M2
13 //Result
14 clc
15 printf('The tension in the strings are T1=%f,T2=%f
  and T3=%f',T1,R(1),R(2))

```

Scilab code Exa 6.5 Equilibrium of NCFS

```

1 //Initilization of variables
2 m=80 //kg
3 g=9.81 //m/s^2
4 //Co-ordinates of points in Meters
5 A=[1,3,0]
6 B=[3,3,-4]
7 C=[4,3,0]
8 D=[2,0,-1]
9 //Calculations
10 //Tension in DC will be
11 a=[C(1)-D(1),C(2)-D(2),C(3)-D(3)]
12 h=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2+(C(3)-D(3))^2)
13 c=a/h
14 //Unit vector calculations
15 e=[B(1)-A(1),B(2)-A(2),B(3)-A(3)]
16 v=sqrt((B(1)-A(1))^2+(B(2)-A(2))^2+(B(3)-A(3))^2)
17 e_ab=e/v
18 //Position vector AD
19 r_ad=[D(1)-A(1),D(2)-A(2),D(3)-A(3)]

```

```

20 //Moment Calculations
21 O=[1,0,0;1,-3,-1;0,-m*g,0]
22 P=[0,1,0;1,-3,-1;0,-m*g,0]
23 Q=[0,0,1;1,-3,-1;0,-m*g,0]
24 C1=[1,0,0;1,-3,-1;2,3,1]
25 C2=[0,1,0;1,-3,-1;2,3,1]
26 C3=[0,0,1;1,-3,-1;2,3,1]
27 rxF1=[det(O),det(P),det(Q)]
28 rxF2=[(det(C1)/h),(det(C2)/h),(det(C3)/h)]
29 //Final Moment calculations
30 rxF=rxF1+rxF2
31 //Taking dot product
32 dot1=e_ab.*rxF
33 dot2=e_ab.*rxF2
34 //equating dot product to zero to obtain C
35 C=-(dot1(1)+dot1(3))/dot2(3)
36 //Result
37 clc
38 printf('The tension in CD is %f N',C)

```

Scilab code Exa 6.6 Equilibrium of NCFS

```

1 //Initialization of variables
2 w=200 //lb
3 Dh=4 //ft
4 //Calculation
5 theta=atand(2/Dh) //degrees
6 T=w/(3*cosd(theta)) //lb
7 //result
8 clc
9 printf('The Tension in each rope is %f lb\n Theta=%f
degrees ',T,theta)

```

Scilab code Exa 6.7 Equilibrium of NCFS

```
1 //Initialization of variables
2 F=[100,0,0] //N
3 CE=5 //m
4 BC=sqrt(34) //m
5 AC=sqrt(41) //m
6 //Calculations
7 //solving as a matrix for system of linear equations
8 A=[3/BC,-4/AC,0;0,0,(6*4)/CE;-3/BC,-3/AC,-3/CE]
9 B=[0;F(1)*4;-F(1)]
10 C=inv(A)*B
11 //Result
12 clc
13 printf('The forces F1 F2 and F3 are as %f N %fN and
        %fN respectively\n',C(1),C(2),C(3))
14 printf('Here F3 is compression assumed and rest are
        Tension ')
```

Scilab code Exa 6.8 Equilibrium of NCFS

```
1 //Initialization of variables
2 F=[100,0,0] //N
3 CE=5 //m
4 BC=sqrt(34) //m
5 AC=sqrt(41) //m
6 //Calculations
7 //solving as a matrix for system of linear equations
8 A=[3/BC,-4/AC,0;0,0,(6*4)/CE;-3/BC,-3/AC,-3/CE]
9 B=[0;F(1)*4;-F(1)]
10 C=inv(A)*B
11 //Result
12 clc
13 printf('The forces F1 F2 and F3 are as %f N %fN and
        %fN respectively\n',C(1),C(2),C(3))
```

```
14 printf('Here F3 is compression assumed and rest are  
Tension')
```

Scilab code Exa 6.9 Equilibrium of NCFS

```
1 //Initialization of variables  
2 //here forces will be defines as matrices along with  
   their co-ordinates  
3 //Force in N and co-ordinates in mm  
4 F1=[30 200 300]  
5 F2=[10 400 200]  
6 F3=[20 200 500]  
7 F4=[50 400 500]  
8 //Calculations  
9 //solving as system of linear equations  
10 A=[1 1 1;-600 -600 0;0 600 600]  
11 B=[F1(1)+F2(1)+F3(1)+F4(1);-(F3(1)*F3(3)+F1(1)*F1(3)  
   +F4(1)*F4(3)+F2(1)*F2(3));-(-F3(1)*F3(2)-F1(1)*F1  
   (2)-F4(1)*F4(2)-F2(1)*F2(2))]  
12 C=inv(A)*B  
13 //Result  
14 clc  
15 printf('The reactions are as R1=%fN, R2=%fN and R3=  
   %fN',C(1),C(2),C(3))
```

Scilab code Exa 6.11 Equilibrium of NCFS

```
1 //Initialization of variables  
2 w=50 //lb wind load  
3 W=60 //lb weight of door  
4 //Calculations  
5 //Calculation as system of linear equations  
6 A=[0 0 33;1 1 -1;28 10 -28]
```

```

7 B=[50*18;-50;-50*24]
8 C=inv(A)*B
9 P=C(3)/cosd(20)
10 D=[-28 -10;1 1]
11 E=[1080-(28*(P*sind(20)));P*sind(20)]
12 F=inv(D)*E
13 By=60
14 //Result
15 clc
16 printf('The forces are as follows: \n')
17 printf('Az=%flb Bz=%flb Pz=%flb Ax=%flb Bx=%flb P=
    %flb By=%flb ',C(1),C(2),C(3),F(1),F(2),P,By)

```

Scilab code Exa 6.12 Equilibrium of NCFS

```

1 //Initilization of variables
2 m=1 //kg
3 g=9.81 //m/s^2
4 t1=45 //degrees
5 t2=30 //degrees
6 //Calculations
7 //Solving as system of linear equations
8 A=[1 0 -cosd(t1) 0;0 1 0 3/5;-5 g*m*cosd(t1)*cosd(t2
    ) 0 0;-1 0 0 4/5]
9 B=[0;g*m;g*m*5*cosd(t1)*cosd(t2);0]
10 C=inv(A)*B
11 //Result
12 clc
13 printf('The forces are Nb=%fN Nc=%fN Tc=%fN Tb=%fN ',
    C(1),C(2),C(3),C(4))

```

Scilab code Exa 6.13 Equilibrium of NCFS

```

1 //Initilization of variables
2 l=200 //lb
3 //Calculations
4 P=1*5/12 //lb
5 //Solving as system of linear equations
6 A=[0 -36 0 0;0 0 0 36;0 0 1 1;1 1 0 0]
7 B=[-P*cosd(25)*48;l*20+P*sind(25)*48;P*sind(25)+200;
    P*cosd(25)]
8 C=inv(A)*B
9 //Result
10 clc
11 printf('The forces are Az=%flb Bz=%flb Ay=%flb By=
    %flb ',C(1),C(2),C(3),C(4))

```

Scilab code Exa 6.14 Equilibrium of NCFS

```

1 //Initilization of variables
2 A=80 //lb
3 B=40 //lb
4 C=60 //lb
5 l1=2 //in
6 l2=4 //in
7 l3=6 //in
8 l4=9 //in
9 l5=3 //in
10 l6=7 //in
11 //Calculations
12 P=-(-A*l1+B*l2-C*l2)/l1
13 By=-(-A*l3+P*l3)/l4
14 Ay=(-A*l5-P*l5)/l4
15 Bz=-(-C*l1-B*l1)/l4
16 Az=(C*l6+B*l6)/l4
17 //Result
18 clc
19 printf('The forces are Ay=%flb By=%flb Az=%flb Bz=

```


`%flb',Ay,By,Az,Bz)`

Scilab code Exa 6.15 Equilibrium of NCFS

```
1 //Initialization of variables
2 W=138 //lb
3 w=80 //lb
4 //Calculations
5 u=sqrt(3*3+4*4+6*6)
6 a=[-3/u 4/u -6/u]
7 v=sqrt(3*3+3*3+3*3)
8 c=[3/v 3/v -3/v]
9 P=[1 0 0;0 0 8;0 -W 0]
10 Q=[0 0 1;0 0 8;0 -W 0]
11 R=[1 0 0;0 0 4;0 -w 0]
12 S=[0 0 1;0 0 4;0 -w 0]
13 T=[1 0 0;0 0 6;a(1) a(2) a(3)]
14 U=[0 1 0;0 0 6;a(1) a(2) a(3)]
15 V=[1 0 0;0 0 3;c(1) c(2) c(3)]
16 Y=[0 1 0;0 0 3;c(1) c(2) c(3)]
17 //Solving for A and C
18 MAT1=[det(T) det(V);det(U) det(Y)]
19 MAT2=[det(P)+det(R);0]
20 res=-inv(MAT1)*MAT2
21 A=[a(1)*res(1) a(2)*res(1) a(3)*res(1)]
22 C=[c(1)*res(2) c(2)*res(2) c(3)*res(2)]
23 E=[-(A(1)+C(1)) -(-w-W+A(2)+C(2)) -(A(3)+C(3))]
24 //Result
25 clc
26 printf('The force vectors are as follows\n')
27 printf('A=%fi+%fj%fk lb and C=%fi+%fj%fk lb also Ex=
%f Ey=%flb Ez=%flb ',A(1),A(2),A(3),C(1),C(2),C(3)
,E(1),E(2),E(3))
28 //Decimal accuracy causes discrepancy in the answers
```

Chapter 7

Trusses and Cables

Scilab code Exa 7.1 Truss and Cable

```
1 //Initialization of variables
2 F1=2000 //lb
3 F2=4000 //lb
4 l1=10 //ft
5 l2=30 //ft
6 l3=20 //ft
7 l4=40 //ft
8 t=60 //degrees
9 //Calculations
10 //Taking moment about point B and A
11 Ra=(F1*l2+F2*l1)/l4
12 Rb=(F2*l2+F1*l1)/l4
13 //Consider fig 7-4(c)
14 A=[1 -cosd(t);0 -sind(t)]
15 B=[0;-2500]
16 C=inv(A)*B
17 //Consider figure 7-4(d)
18 A1=[1 cosd(t);0 -sind(t)]
19 B1=[-C(2)*cosd(t);-C(2)*sind(t)+F1]
20 C1=inv(A1)*B1
21 //Consider figure 7-4(e)
```

```

22 CD=577
23 CE=C(1)+C1(2)*cosd(t)+CD*cosd(t)
24 //Consider figure 7-4(f)
25 DE=Rb/sind(t)
26 //Result
27 clc
28 printf('The reactions are Ra=%flb and Rb=%flb\n',Ra,
        Rb)
29 printf('Force in member AB=%flb and AC=%flb\n',C(2),
        C(1))
30 printf('Force in member BC=%flb and BD=%flb\n',C1(2)
        ,C1(1))
31 printf('Force in member CD=%flb and CE=%flb\n',CD,CE
        )
32 printf('Force in member DE=%flb',DE)
33 //Decimal Accuracy causes discrepancy in answers

```

Scilab code Exa 7.2 Truss and Cable

```

1 //Initilization of variables
2 s=4 //m length of sides
3 l=2 //kN load acting on each node
4 r=7 //kN by inspection reaction at A
5 //Calculation
6 //Taking Moment about point G
7 FH=(-r*12+2*10+2*6+2*2)/(2*tand(60)) //kN
    Compressive
8 //Taking moment about point H
9 GI=(r*14-2*12-2*8-2*4)/(2*tand(30)) //kN Tension
10 //Summing forces in the vertical direction
11 HG=-(r-(1*3))/sind(60) //kN Compression
12 //Taking moment about point J yields
13 IK=(-2*4-2*8+r*10)/(2*tand(60)) //kN
14 //Result
15 clc

```

```

16 printf('The value of the forces in the components
    are as follows\n')
17 printf('FH=%fkN, GI=%fkN, HG=%fkN and IK=%fkN\n', FH, GI
    , HG, IK)
18 printf('The answer in the text book for GI is wrong'
    )

```

Scilab code Exa 7.3 Truss and Cable

```

1 //Initialization of variables
2 theta=30 //degrees
3 EF=40000 //lb
4 l=36 //feet
5 //Calculation
6 //Taking moment about point D and setting EF=40000
    lbs
7 P=- (EF*sind(theta)*l)/l //lb
8 //Result
9 clc
10 printf('The maximum value of P is %flb\n', P)
11 printf('The negative sign indicates the downward
    direction ')

```

Scilab code Exa 7.4 Truss and Cable

```

1 //Initialization of variables
2 l=12 //m
3 theta1=30 //degrees
4 F1=1000 //N
5 F2=2000 //N
6 //Calculation
7 FG=l*cosd(theta1) //m
8 DG=(l+(l/2))/cosd(theta1) //m

```

```

 9 //Taking moment about point G
10 A=(F1*1+F2*FG+F1*DG)/(1*3) //N
11 //Summing forces in horizontal direction
12 G_x=(2*F1+F2)*sind(theta1) //N
13 //Summing forces in the vertical direction
14 G_y=(2*F1+F2)*cosd(theta1)+F1-A //N
15 //Taking moment about point C
16 BD=-(A*1)/(1/2) //N
17 //Taking moment about point D
18 CE=(A*(1+(1/2)))/FG //N
19 theta=atand((1/2)/FG) //degrees
20 //Summing forces in the vertical direction
21 CD=(A+(BD*cosd(60)))/cosd(theta) //N
22 //Result
23 clc
24 printf('The values of the forces are as follows\n')
25 printf('A=%fN , G_x=%fN , G_y=%fN , BD=%fN , CE=%fN and CD=
    %fN ', A , G_x , G_y , BD , CE , CD)
26 //Decimal Accuracy causes discrepancy in answers

```

Scilab code Exa 7.5 Truss and Cable

```

1 //Initilization of variables
2 A=2000 //lb
3 E=2000 //lb
4 theta=60 //degrees
5 theta1=30 //degrees
6 //Sign convention positive means Tension and
    negative means Compression
7 //Taking sum of forces along x and y direction in
    fig7-13
8 AB=-A/sind(theta) //lb
9 AG=-AB*cosd(theta) //lb
10 //Taking sum of forces along x and y direction in
    fig7-14

```

```

11 BG=((-AB*cosd(theta1))-1000)/(cosd(theta1)) //lb
12 BC=((AB*sind(theta1))-(BG*sind(theta1))) //lb
13 //Taking sum of forces along x and y direction in
    fig7-15
14 GC=-(BG*sind(theta))/sind(theta) //lb
15 GF=AG+BG*cosd(theta)-GC*(cosd(theta)) //lb
16 //By symmetry of structure
17 DE=AB //lb
18 FE=AG //lb
19 DF=BG //lb
20 CD=BC //lb
21 //Result
22 clc
23 printf('The forces in the truss are\n')
24 printf('AB=DE=%flb ,AG=FE=%flb ,BG=DF=%flb ,BC=CD=%flb
    and CG=CF=%flb ',AB ,AG ,BG ,BC ,GC)

```

Scilab code Exa 7.6 Truss and Cable

```

1 //Initilization of variables
2 F=500 //N
3 A=1000 //N
4 theta=60 //degrees
5 l=20 //m
6 //Calculations
7 //Taking moment about point G
8 R_c=(20*3*A+50*F+30*F+10*F)/40 //N
9 //Returning to fig7-17
10 //Taking moment about point C
11 BD=(1*A+(1/2)*F)/(1*sind(theta)) //N
12 //Taking sum of forces in vertical direction
13 CD=(A+F-R_c)/sind(theta) //N
14 //Result
15 clc
16 printf('The forces in the members are as follows\n')

```

```

17 printf('BD=%fN and CD=%fN, Also the reaction at C is
    %fN ', BD, CD, R_c)
18 //Decimal accuracy causes discrepancey in answers

```

Scilab code Exa 7.7 Truss and Cable

```

1 //Initilization of variables
2 w=800 //lb/ft
3 a=600 //ft
4 d=40 //ft
5 //Calculations
6 T=0.5*w*a*(sqrt(1+(a^2/(16*d^2)))) //lb
7 H=(w*a^2)/(8*d) //lb
8 //Taking the first two terms of the series
9 l=a(1+(8/3)*(d/a)^2-(32/5)*0.00002) //ft
10 //Result
11 clc
12 printf('The value of T=%flb and that of H=%flb , Also
    l=%fft ', T, H, l)
13 //Deciaml accuracy causes discrepancey in answers

```

Scilab code Exa 7.8 Truss and Cable

```

1 //Initilization of variables
2 l=800*300 //lb
3 //Calculations
4 //Summing forces in horizontal and vertical
    direction
5 theta=atand(40/150) //degrees
6 H=l/tand(theta) //lb
7 T_max=sqrt(l^2+H^2) //lb
8 //Result
9 clc

```

```

10 printf('The maximum tension is %flb and H=%flb ',
        T_max,H)
11 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 7.9 Truss and Cable

```

1 //Initialization of variables
2 //The variable declaration is taken to simplify the
  solution
3 x1=10 //m
4 x2=20 //m
5 m=3 //kg/m
6 g=9.8 //m/s^2
7 //For simplicity a1 and a2 values are being
  considered as constant free of H
8 a_1=sqrt(x1/(m*g*0.5))
9 a_2=sqrt((x1+x2)/(m*g*0.5))
10 y=10 //m
11 //Calculations
12 H=(300/(a_1+a_2))^2 //N
13 //Now reconsidering a1 and a2 actual values
14 a1=a_1*sqrt(H) //m
15 a2=a_2*sqrt(H) //m
16 //Theta calculations
17 x=a1
18 theta=atand(2*y/x)
19 //T calculations
20 T=sqrt((864*a2^2)+H^2) //N
21 //result
22 clc
23 printf('The tension in the cable is %fN',T)

```

Scilab code Exa 7.10 Truss and Cable


```

1 //Initilization of variables
2 T=140000 //N
3 w=2000 //N/m
4 a=20 //m
5 // Calculations
6 // Calculation step by step
7 lhs=(140000*2)/(2000*20)
8 d=sqrt(1/((((lhs^2)-1)*16)/(20^2))) //m
9 l=a(1+(8/3)*(d/a)^2) //m
10 //Result
11 clc
12 printf('The sag in the cable is %fm and the required
        length is %fm',d,l)

```

Scilab code Exa 7.11 Truss and Cable

```

1 //Initilization of variables
2 w=10/16 //lb/ft
3 a=80 //ft
4 P=500 //lb
5 // Calculations
6 lhs=(P*2)/(w*a)
7 d=sqrt(1/((((lhs^2)-1)*16)/(80^2))) //ft
8 //Result
9 clc
10 printf('The sag in the cable is %fft',d)

```

Scilab code Exa 7.12 Truss and Cable

```

1 //Initilization of variables
2 w=0.518 //lb/ft
3 d=50 //ft
4 l=500 //ft

```

```

5 //Plot coding
6 A=linspace(0,800,9) //defined x axis
7 B=A+50
8 C=[50000,(500/(2*100)),(500/(2*200)),(500/(2*300))
    ,(500/(2*400)),(500/(2*500)),(500/(2*600))
    ,(500/(2*700)),(500/(2*800))]
9 D=cosh(C)
10 E=[D(1)*A(1),D(2)*A(2),D(3)*A(3),D(4)*A(4),D(5)*A(5)
    ,D(6)*A(6),D(7)*A(7),D(8)*A(8),D(9)*A(9)]
11 plot(A,B,A,E) //plotting two lines on the same plot
12 //Calculations
13 //By close observation of plot taking c around 650
14 //consider c=635
15 c=635
16 T_max=w*(c+d) //lb
17 a=c+d
18 l=sqrt(4*(a*a-c*c))
19 //Result
20 clc
21 printf('The maximum tension is %flb and length
    required is %fft',T_max,l)

```

Scilab code Exa 7.13 Truss and Cable

```

1 //Initilization of variables
2 m=0.6 //kg/m
3 l=240 //m
4 d=24 //m
5 //Calculations
6 c=(((1/4)*(l^2))-(24*24))/(2*d)
7 T_max=9.8*m*(c+d) //N
8 a=asinh((l)/(2*c))*576
9 //Result
10 clc
11 printf('The maximun tension is %fN and a=%fm',T_max,

```

a)

Chapter 8

Forces In Beams

Scilab code Exa 8.1 Beams

```
1 //Initilization of variables
2 R_A=100 //N
3 R_B=200 //N
4 //Calculations
5 //Shear force at 2m
6 V=100 //N
7 //Moment at 2m
8 M=R_A*2 //N.m
9 //Result
10 clc
11 printf('The shear force at 2m is %fN and the moment
    at 2m is %fN-m',V,M)
```

Scilab code Exa 8.2 Beams

```
1 //Initilization of variables
2 //length matrix
3 L1=[0,4,6] //m
```

```

4 //Bending moment matrix
5 B=[0,400,0] //N.m
6 //Shear force plotting
7 //Here the left side and right side lengths are
   considered as close as 4 to keep up with right
   and left distinctions
8 L=[0,3.999,4,5.99998,6]
9 S=[100,100,-200,-200,0]
10 //Calculations cum Result
11 subplot(221)
12 plot(L1,B)
13 xtitle("Bending Moment Diagram","Span","Bending
   Moment")
14 subplot(222)
15 plot(L,S,L,0)
16 xtitle("Shear Force Diagram","Span","Shear Force")

```

Scilab code Exa 8.3 Beams

```

1 //Initilization of variables
2 w=196 //N/m
3 M_app=4000 //N.m
4 L=6 //m
5 //Calculations
6 //Taking Moment about Point L and equating it to 0
7 R_r=(M_app+w*L*L*0.5)/(3*L) //N
8 //Taking Moment about Point R and equating it to 0
9 R_l= (((2*L)+(L/2))*(w*L))-(M_app))/(3*L) //N
10 //finding point of zero shear
11 a=R_l/w
12 //defining x
13 x0=[0,18]
14 x=[0,0.5,1,1.5,2,2.5,3,3.5,a,4,4.5,5,5.5,6] //for 0<
   x<6
15 x1=[6,12] //for6<x<12

```

```

16 x2=[12,18] //for 12<x<18
17 xv=[6,12,18] //specially for shear force
18 xo=[12.001,12.002] //Straight line plot
19 //Shear Force Calculations
20 //Summing forces in vertical direction and equating
    to 0
21 V1=R_l-w*x //N for 0<x<6
22 V2=R_l-w*L //N for 6<x<18
23 //Bending Moment Calculations
24 M1=R_l*x-(w*x^2*0.5) //N.m for 0<x<6
25 M2=R_l*x1-((w*L)*(x1-3)) //N.m for 6<x<12
26 M3=R_l*x2-((w*L)*(x2-3))+M_app //N.m for 12<x<18
27 Mo=[-1464.8652,2509.3333]
28 //Maximum bending moment
29 M_max=R_l*a*0.5 //N.m
30 //Plotting
31 subplot(221)
32 plot(x,V1,xv,V2,x0,0)
33 xtitle('Shear Force Diagram','Span','Shear Force')
34 subplot(222)
35 plot(x,M1,x1,M2,x2,M3,x0,0,xo,Mo)
36 xtitle('Bending Moment Diagram','Span','Bending
    Moment')
37 //Result
38 clc
39 printf('The value of reactions are R_l=%fN and R_r=
    %fN\n',R_l,R_r)
40 printf('The point of maximum bending moment is %f
    meters from left support nad maximum bending
    moment is %fN-m\n',a,M_max)
41 printf('The bending moment and shear force diagrams
    have been plotted')

```

Scilab code Exa 8.5 Beams

```

1 //Initialization of variables
2 F1=2000 //lb
3 w=100 //lb/ft
4 //Calculations
5 R_r=(-F1*5+w*14*13)/20 //lb
6 R_l=(F1*25+w*14*7)/20 //lb
7 //Shear Force matrix
8 V=[-2000,-2000,990,990,-410,0] //lb
9 //Bending Moment matrix
10 B=[0,-10000,-4060,840,0]
11 //Length matrix for shear force
12 X_v=[0,5,5.0001,11,20.89999,20.9]
13 //Length matrix for bending moment
14 X_b=[0,5,11,19.9,20.9]
15 //Plotting
16 subplot(221)
17 plot(X_v,V,X_v,0)
18 xlabel("Shear Force Diagram","Span","Shear Force")
19 subplot(222)
20 plot(X_b,B,X_b,0)
21 xlabel("Bending Moment Diagram","Span","Bending
    Moment")
22 //Result
23 clc
24 printf('The bending Moment and Shear Force diagrams
    have been plotted\n')
25 //Note
26 //The textbook does not specify the span and hence
    there seems to be a disagreement between the
    textbook and scilab solution.here the values have
    just been plotted

```

Chapter 9

Friction

Scilab code Exa 9.1 Friction

```
1 // Calculations
2 // Simplifying equation (3) after substituting value
   of Nb in it we get
3 //  $m \cdot \mu^2 + m \cdot \mu \cdot 2 \cdot \tan(50) - 1 = 0$ 
4 // Solution of the equation
5 a=1
6 b=2*tand(50)
7 c=-1
8 g=sqrt(b^2-(4*a*c))
9 // solution
10 x1=(-b+g)/(2*a)
11 x2=(-b-g)/(2*a)
12 // As x2 does not make any physical sense x1 is the
   answer
13 // Result
14 clc
15 printf('The value of mu is %f',x1)
```

Scilab code Exa 9.3 Friction


```

1 //Initilization of variables
2 m=70 //kg
3 g=9.81 //m/s^2
4 theta=20 //degrees
5 //Calculations
6 //Solving by martix method
7 //Taking sum along vertical and horizontal direction
   and equating them to zero
8 A=[sind(theta) 1 0;-cosd(theta) 0 1;0 -1/4 1]
9 //RHS matrix
10 R=[m*g;0;0]
11 ans1=inv(A)*R //force vector N
12 //Calculation part 2
13 //Similar solution by matrix method
14 //Taking moment about point O and summing forces in
   horizontal and vertical direction and equating
   all to zero
15 B=[4*cosd(theta) 0 0;-cosd(theta) 1 0;sind(theta) 0
   1]
16 //RHS matrix
17 J=[m*g*1.5;0;m*g]
18 ans2=inv(B)*J //force Vector N
19 //Result
20 clc
21 printf('The value of P in first case is %iN and that
   in second case is %iN',ans1(1),ans2(1))

```

Scilab code Exa 9.4 Friction

```

1 //Initilization of variables
2 W=200 //lb
3 Fapp=300 //lb
4 mu=0.3 //coefficient of friction
5 theta=30 //degrees
6 //Calculations

```

```

7 //Summing forces in the plane parallel to the slope
8 F=-(W*sind(theta)-Fapp*cosd(theta)) //lb
9 N1=(W*cosd(theta)+Fapp*sind(theta)) //lb
10 //Max value obtained
11 Fprime= mu*N1
12 //Result
13 clc
14 printf('The value of F and N1 are %flb and %flb
        respectively and the maximum value obtained is
        %flb ',F,N1,Fprime)

```

Scilab code Exa 9.5 Friction

```

1 //Initilization of variables
2 mu1=0.2 //coefficient of friction between wedges and
        A
3 mu2=1/4 //coefficient of friction between wedges
4 F=20 //tonnes
5 //Calculations
6 //Using the matrix method to solve
7 //Summing forces in vertical and horizontal
        direction
8 A=[1,-(mu1*10+1)/(sqrt(101));0,(10-mu1*1)/sqrt(101)
        ] //force matrix
9 B=[mu2*F*1000;F*1000] //lb
10 //Solving both matrices
11 R=inv(A)*B //lb
12 //Result
13 clc
14 printf('The forces N2 and P are %ilb and %ilb
        respectively ',R(2),R(1))
15 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 9.6 Friction

```
1 //Initialization of variables
2 theta=45 //degrees
3 mu1=1/4 //coefficient of friction between A and B
4 mu2=1/3 //coefficient of friction between A and
  Floor
5 ma=14 //kg
6 mb=9 //kg
7 g=9.81 //m/s^2
8 //Calculations
9 //Summing forces in vertical direction
10 Nb=mb*g //N
11 //Also
12 Fprimeb=mu1*Nb //N
13 //Summing forces in direction
14 T=Fprimeb //N
15 //Considering the fig(c)
16 //Summing forces in the horizontal direction and
  vertical direction and solving by matrix method
17 A=[-cosd(theta) mu2;sind(theta) 1] //N
18 B=[-Fprimeb;(mb*g+ma*g)] //N
19 R=inv(A)*B //N
20 //Result
21 clc
22 printf('The value of P and Na are %fN and %fN
  respectively ',R(1),R(2))
```

Scilab code Exa 9.7 Friction

```
1 //Initialization of variables
2 m1=40 //kg
3 m2=13.5 //kg
4 mu=1/3 //coefficient of friction
5 g=9.81 //m/s^2
```

```

6 // Calculations
7 // Solving by substitution
8 // After simplification we get
9 x=mu*m2*g
10 y=mu*(m1*g+m2*g)
11 theta=atand((x+y)/(m1*g)) //degrees
12 // Result
13 clc
14 printf('The value of the angle is %f degrees',theta)

```

Scilab code Exa 9.8 Friction

```

1 // Initialization of variables
2 W=350 //lb
3 theta=30 //degrees
4 phi=15 //degrees
5 // Calculations
6 // Solving by the matrix method
7 A=[cosd(theta) sind(phi);-sind(theta) cosd(phi)]
8 B=[W*sind(theta);W*cosd(theta)]
9 an=inv(A)*B //lb
10 // Result
11 clc
12 printf('The value of P and R are %flb and %flb
    respectively ',an(1),an(2))

```

Scilab code Exa 9.9 Friction

```

1 // Initialization of variables
2 theta=45 //degrees
3 m1=45 //kg
4 m2=135 //kg
5 g=9.81 //m/s^2

```

```

6 mu=0.25 //coefficient of riction
7 //Calculations
8 N2=m2*g //N
9 T=mu*N2 //N
10 N1=m1*g*cosd(theta) //N
11 Fprime1=N1*mu //N
12 P=T+Fprime1-(m1*g*sind(theta)) //N
13 //Result
14 clc
15 printf('The values are N2=%fN,T=%fN,N1=%fN,Fprime1=
%fN and P=%fN',N2,T,N1,Fprime1,P)

```

Scilab code Exa 9.10 Friction

```

1 //Initilization of variables
2 mu=0.2 //coefficient of friction
3 F1=150 //lb
4 F2=100 //lb
5 theta=60 //degrees
6 //Calculations
7 N1=F1*cosd(theta) //lb
8 T=(mu*N1+(F1*cosd(theta/2))) //lb considering
    positive
9 //Equilibrium for 100lb
10 //Eliminating N2 from both equations
11 //Taking derivative we get
12 theta2=atand(mu) //degrees
13 //Hence P becomes
14 P=(F2*mu+T)/(cosd(theta2)+(mu*sind(theta2))) //lb
15 //Result
16 clc
17 printf('The minimum value of P is %flb',P)

```

Scilab code Exa 9.11 Friction

```
1 //Initialization of variables
2 F=180 //N
3 m=100 //kg
4 g=9.81 //m/s^2
5 mu=0.25 //coefficient of friction
6 //Calculations
7 //Assuming F2 is maximum
8 N2=F*2/(1+mu) //N
9 F2=mu*N2 //N
10 N1=m*g-F2 //N
11 F1=F-F2 //N
12 //Result
13 clc
14 printf('The vaules are N2=%fN,F2=%fN,N1=%fN and F1=
    %fN ',N2 ,F2 ,N1 ,F1)
```

Scilab code Exa 9.12 Friction

```
1 //Initialization of variables
2 N=4/3 //Normal reaction after solving without the mg
    term in it
3 u_N=1/2 //Frictional force without the mg term in it
4 //Calculations
5 u=u_N/N //Coefficient of friction
6 //Result
7 clc
8 printf("The frictional co-efficient is %f",u)
9 //The answer in the textbook and the code differs
    due to multiplication of fractions
```

Scilab code Exa 9.13 Friction

```

1 //Initilization of variables
2 mu_ca=0.3 //ceofficient of friction between copper
    block A and aluminium block B
3 mu_af=0.2 //coefficient of friction between
    aluminium block B and Floor
4 ma=3 //kg
5 mb=2 //kg
6 g=9.81 //m/s^2
7 //Calculations
8 //For A
9 //Taking sum of forces along X and Y direction
10 Na=ma*g //N
11 P=mu_ca*Na //N
12 //For B
13 //Taking sum of forces along X and Y direction
14 Nb=Na+mb*g //N
15 Fb=mu_ca*Na //N
16 //Now largest value of friction before slip is
17 Fprimeb=mu_af*Nb //N
18 //Now as Fb<F'b hence initial assumption is
    incorrect and P=Fb
19 P=Fb //N
20 //Result
21 clc
22 printf('The value of force that will cause motion is
    %fN',P)

```

Scilab code Exa 9.15 Friction

```

1 //Initilization of variables
2 d_m=2 //in    mean diameter of the screw
3 p=1/4 //in
4 mu=0.15 //coefficient of friction
5 l=2 //ft
6 L=4000 //lb

```

```

7 //Calculations
8 phi=atand(mu) //degrees
9 beta=atand(p/(%pi*1)) //degrees
10 //Force to raise the load
11 P=(L*tand(phi+beta))/(d_m*12) //lb
12 //Force to lower the load
13 P2=(L*tand(phi-beta))/(d_m*12) //lb
14 //Result
15 clc
16 printf('The force to raise the load is %flb and to
        lower is %flb',P,P2 )

```

Scilab code Exa 9.16 Friction

```

1 //Initilization of variables
2 r_m=2.338 //in
3 d_m=3.25 //in
4 mu=0.06 //coefficient of friction
5 P=1500 //lb
6 p=1/4 //pitch
7 //Calculation
8 phi=atand(mu) //degrees
9 beta=atand(p/(2*%pi*r_m)) //degrees
10 M=P*r_m*tand(phi+beta)+mu*P*(d_m/2) //lb.in
11 //Result
12 clc
13 printf('The moment required is %flb-in',M)
14 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 9.17 Friction

```

1 //Initilization of variables
2 d=750 //mm diameter

```



```

3 alpha=%pi //wrap angle    radians
4 mu=0.25 //coefficient of friction
5 T_t=200 //N tension on the tight side
6 //Calculation
7 T2=T_t/(exp(mu*alpha)) //N
8 //Result
9 clc
10 printf('The tension of the slack side is %fN',T2)

```

Scilab code Exa 9.18 Friction

```

1 //Initilization of variables
2 d=635 //mm diameter of the drum
3 P=178 //N
4 mu=1/3 //coefficient of friction
5 l1=100 //mm
6 l2=660 //mm
7 theta1=60 //degrees
8 GD=d/2 //mm
9 //Calculations
10 //Taking moment about point C
11 Tb=(P*(l1+l2))/(l1*sind(theta1)) //N
12 CD=((d/2)-(l1*cosd(theta1/2)))/sind(theta1/2) //mm
13 //from fig 9-22(b)
14 theta=asind(GD/CD) //degrees
15 //from fig9-22(c)
16 w_d=180+30+theta //degrees
17 w=(w_d)*(%pi/180) //radians
18 //As Tc is greater than Tb
19 Tc=Tb*(exp(mu*w)) //N
20 M=(Tc-Tb)*GD //N.mm
21 an=M/1000 //N.m
22 //Result
23 clc
24 printf('The braking moment required is %fN-m',an)

```

25 //Note the unit of the final answer carefully

Scilab code Exa 9.19 Friction

```
1 //Initialization of variables
2 L=1000 //lb
3 P=10 //lb
4 //Calculations
5 mu=log(L/P)/(4*2*%pi)
6 //Result
7 clc
8 printf('The coefficient of friction is %f',mu)
```

Scilab code Exa 9.20 Friction

```
1 //Initialization of variables
2 m=900 //kg
3 mu=0.2 //coefficient of friction
4 g=9.8 //m/s^2
5 //Calculations
6 T2=m*g/(exp(2*2*%pi*mu)) //N
7 //Result
8 clc
9 printf('The force needed to hold the mass is %fN',T2
    )
```

Scilab code Exa 9.21 Friction

```
1 //Initialization of variables
2 d=760 //mm
```

```
3 W=500 //N
4 a=0.305 //mm coefficient of rolling resisatnce
5 r=d/2 //mm
6 // Calculations
7 P=(W*a)/r //N
8 //Result
9 clc
10 printf('The force necessary is P=%fN',P)
```

Chapter 10

First Moment and Centroid

Scilab code Exa 10.5 1st Moment and CG

```
1 //Initialization of variables
2 r=50 //mm
3 L1=75 //mm
4 L2=%pi*r //mm
5 L3=61.2 //mm
6 theta1=45 //degrees
7 theta2=60 //degrees
8 //Calculations
9 x_bar=[(L1/2)*cosd(theta1),L1*cosd(theta1)+r,L1*cosd
        (theta1)+100+(L3/2)*cosd(theta2)] //mm
10 y_bar=[(L1/2)*sind(theta1),L1*sind(theta1)+(2*r)/%pi
        ,(L3/2)*sind(theta2)] //mm
11 //Centroid Calculations
12 x=(L1*x_bar(1)+L2*x_bar(2)+L3*x_bar(3))/(L1+L2+L3)
    //mm
13 y=(L1*y_bar(1)+L2*y_bar(2)+L3*y_bar(3))/(L1+L2+L3)
    //mm
14 //Result
15 clc
16 printf('The centroid is as follows x=%f mm and y=
        %fmm',x,y)
```

Scilab code Exa 10.6 1st Moment and CG

```
1 //Initialization of variables
2 theta=75 //degrees
3 alpha=(150*%pi)/180 //rad
4 r=1
5 theta1=30 //degrees
6 lhor=14 //in
7 //calculations
8 a=((2*r)/alpha)*sind(theta) //in
9 y=-a*sind(90-theta) //in
10 //Length of arc
11 l=r*alpha //in
12 //Slope length calculations
13 DF=7 //in
14 AB=DF //in
15 BC=1 //in
16 BF=BC*cosd(theta1) //in
17 FC=BC*sind(theta1) //in
18 DC=DF+FC //in
19 EC=DC/cosd(theta1) //in
20 //Centroid of EC is at G
21 yslope=0.5*EC*sind(theta1)+BF //in
22 //Y of composite figure
23 Y=((2*l*y)+14*-1+(2*EC*yslope))/(2*l+lhor+2*EC) //in
24 //Result
25 clc
26 printf('The centroid is at Y=%f in ',Y)
```

Scilab code Exa 10.11 1st Moment and CG

```

1 //Initialization of variables
2 a=100 //mm
3 b=150 //mm
4 A1=2*10^4 //mm^2
5 A2=5*10^3 //mm^2
6 A3=(%pi*(a/2)^2)/2 //mm^2
7 //Calculations
8 x=(A1*a+A2*(133.3)-A3*b)/(A1+A2-A3) //mm
9 y=(A1*a*0.5+A2*(116.66)-A3*((4*a*0.5)/(3*%pi)))/(A1+
    A2-A3) //mm
10 //Result
11 clc
12 printf('The centroidal distances are x=%f mm and y=
    %f mm',x,y)

```

Scilab code Exa 10.16 1st Moment and CG

```

1 //Initialization of variables
2 V=[1728*10^3,432*10^3,7.54*10^3]
3 x_bar=[60,140,60] //mm
4 y_bar=[30,20,30] //mm
5 //Calculations
6 x=(V(1)*x_bar(1)+V(2)*x_bar(2)+V(3)*x_bar(3))/(V(1)+
    V(2)+V(3)) //mm
7 y=(V(1)*y_bar(1)+V(2)*y_bar(2)+V(3)*y_bar(3))/(V(1)+
    V(2)+V(3)) //mm
8 z=120 //mm from symmetry
9 //Result
10 clc
11 printf('The centroid is at x=%f mm y=%f mm and z=
    %fmm',x,y,z)
12 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 10.17 1st Moment and CG

```
1 //Initilization of variables
2 tx=30 //degrees
3 ty=45 //degrees
4 tz=60 //degrees
5 //Calculations
6 V=[10,15,25] //in^3
7 x_bar=[4,12,24] //in
8 y_bar=[4*cosd(tx),-6*cosd(ty),-4*cosd(tz)]
9 z_bar=[-4*sind(tx),6*sind(ty),-4*sind(tz)]
10 //Centroid calculations
11 x=(V(1)*x_bar(1)+V(2)*x_bar(2)+V(3)*x_bar(3))/(V(1)+
    V(2)+V(3)) //in
12 y=(V(1)*y_bar(1)+V(2)*y_bar(2)+V(3)*y_bar(3))/(V(1)+
    V(2)+V(3)) //in
13 z=(V(1)*z_bar(1)+V(2)*z_bar(2)+V(3)*z_bar(3))/(V(1)+
    V(2)+V(3)) //in
14 //Result
15 clc
16 printf('The centroid is at x=%f in,y=%f in and z=%f
    in',x,y,z)
```

Scilab code Exa 10.26 1st Moment and CG

```
1 //Initilization Of Variables
2 a=0 //Lower Limit of the Integral
3 b=8 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 //Calculation
6 //Using Trapezoidal Rule for Intergration
7 function [I1]=Trap_Composite1(f,a,b,n)
8     h=(b-a)/n
9     t=linspace(a,b,n+1)
10     I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
```

```

11 endfunction
12 deff(' [y]=f(t) ', 'y=75*t^2 ')
13 Rr=Trap_Composite1(f,a,b,n)/(2*8) //lb
14 //Moment calculations
15 M=Trap_Composite1(f,a,b,n) //ft-lb
16 //Result
17 clc
18 printf('The reaction is %f lb and Moment is %f lb-ft
        ',Rr,M)
19 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 10.27 1st Moment and CG

```

1 //Initilization Of Variables
2 a=0 //Lower Limit of the Integral
3 b=0.3 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 g=9.8 //m/s^2
6 rho=1000 //kg/m^3
7 //Calculation
8 //Using Trapezoidal Rule for Intergration
9 function [I1]=Trap_Composite1(f,a,b,n)
10     h=(b-a)/n
11     t=linspace(a,b,n+1)
12     I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
13 endfunction
14 deff(' [y]=f(t) ', 'y=(g*0.6*rho*1.2*t) - (0.6*g*rho*t^2)
        ')
15 B=Trap_Composite1(f,a,b,n)/(2*b) //N
16 //Result
17 clc
18 printf('The value of B is %f N',B)

```

Scilab code Exa 10.28 1st Moment and CG

```
1 //Initialization of variables
2 l=62.4 //lb/ft^3
3 h=12 //ft
4 f=105 //lb/ft^3
5 //Calculations
6 p1=l*h //lb/ft^2
7 //Total force on left side
8 //Simplifying the equation we get a three degree
   equation in d
9 //solving for d
10 p=[1/3 0 -144 467]
11 r=roots(p)
12 d=r(3) //ft
13 //Result
14 clc
15 printf('The value of d is %f ft ',d)
```

Chapter 11

Virtual Work

Scilab code Exa 11.5 V W

```
1 //Initialization of variables
2 //Simplification constants
3 a=90
4 b=30
5 c=60
6 //Calculations
7 //Allowing for only the cos and sin terms to be zero
   after simplification
8 theta1=atand(a/(b+2*c)) //degrees
9 theta2=atand(a/(b+c)) //degrees
10 theta3=atand(a/b) //degrees
11 //Result
12 clc
13 printf('The values of theta1 ,theta2 and theta3 are
   %f,%f and %f respectively in Degrees ',theta1,
   theta2 ,theta3)
```

Scilab code Exa 11.6 V W

```

1 //Initilization of variables
2 N=100 //lb
3 mu=0.3 //coefficient of friction
4 l=5 //in compressed to length
5 //Calculations
6 //Simplfying the calculations we obtain
7 M=8*(N+N*mu) //lb-in
8 //Result
9 clc
10 printf('The Moment is %i lb-in',M)

```

Scilab code Exa 11.7 V W

```

1 //Initilization of variables
2 m=10 //kg
3 g=9.8 //m/s^2
4 F=200 //N
5 l=3 //m
6 //Calculations
7 //Applying Virtual work principle
8 By=m*g*0.5 //N
9 Bx=F*(2/3) //N
10 //By equations of equilibrium
11 Ax=-Bx-F //N negative sign indictaes the LEFT
    orientation
12 Ay=m*g-By //N
13 //Result
14 clc
15 printf('The values are Ax=%fN,Ay=%fN,Bx=%fN and By=
    %fN',Ax,Ay,Bx,By)

```

Scilab code Exa 11.10 V W

```
1 //Initialization of variables
2 l=2 //ft
3 W=20 //lb
4 k=144 //lb/ft
5 r=0.5 //ft
6 theta=44.1 //degrees
7 //Calculations
8 //Simplifying the solution to obtain
9 sinetheta=(k*r^2)/(W*l) //in terms of theta
10 //By trial and error theta=44.1 degrees
11 //Check for stable equilibrium
12 Check=-W*cosd(theta)*l+k*r^2
13 //Result
14 clc
15 printf('the Check Value is %f which indicates Stable
        Equilibrium ',Check)
```

Chapter 12

Kinematics of a Particle

Scilab code Exa 12.1 Kin of a Part

```
1 //Initilization of variables
2 t=4 //seconds
3 //Calculations
4 //Displacement
5 x=3*t^3+t+2 //ft
6 //Velocity
7 v=9*t^2+1 //ft/s
8 //Acceleration
9 a=18*t //ft/s^2
10 //Result
11 clc
12 printf('The dipalacemnt is %f ft and the velocity is
        %f ft/s and Acceleration is %f ft/s^2',x,v,a)
```

Scilab code Exa 12.2 Kin of a Part

```
1 //Initilization of variables
2 t1=4 //s
```

```

3 t2=5 //s
4 // Calculation
5 v1=9*t1^2+1 //ft/s
6 v2=9*t2^2+1 //ft/s
7 a=(v2-v1)/(t2-t1) //m/s^2
8 //Result
9 clc
10 printf('The acceleration during fifth second is %f
        ft/s^2',a)

```

Scilab code Exa 12.3 Kin of a Part

```

1 //Defining Matrices
2 t=[0 1 2 3 4 5 10] //s
3 //Displacement matrix
4 s=[8*t^2+2*t] //m
5 //Velocity Matrix
6 v=[16*t+2] //m/s
7 //Acceleration Matrix
8 a=16 //m/s^2
9 //Plotting the curves
10 //S-T curve
11 subplot(221)
12 plot(t,s)
13 xlabel('t(s)')
14 ylabel('s(m)')
15 subplot(222)
16 plot(t,v)
17 xlabel('t(s)')
18 ylabel('v(m/s)')
19 subplot(223)
20 plot(t,a)
21 xlabel('t(s)')
22 ylabel('a(m/s^2)')
23 //Result

```

```
24 clc
25 printf('The graphs are the solutions')
```

Scilab code Exa 12.4 Kin of a Part

```
1 //Initialization of variables
2 v_o=0 //ft/s
3 v_f=88//ft/s
4 t=28 //s
5 //Calculations
6 k=(v_f-v_o)/t //ft/s^2
7 s=((v_f-v_o)/2)*t //ft
8 //Result
9 clc
10 printf('The constant k is %f and displacement is %f
    ft ',k,s)
11 //Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 12.5 Kin of a Part

```
1 //Initialization of variables
2 v_o=0 //ft/s
3 v_f1=30 //ft/s
4 v_f2=0 //ft/s
5 t1=3 //s
6 t2=2 //s
7 //Calculations
8 //Plotting the v-t curve
9 //Velocity matrix
10 v=[v_o,v_f1,v_f2]
11 //Time matrix
12 t=[0,3,5]
13 plot(t,v)
```

```

14 xlabel('t')
15 ylabel('v')
16 //Part "b"
17 //Acceleration at 3s
18 a1=(v_f1-v_o)/t1 //ft/s^2
19 //Acceleration at 5s
20 a2=(v_f2-v_f1)/t2 //ft/s^2
21 //Part "c"
22 s=(v_f1*t1*0.5)+(v_f1*t2*0.5) //ft
23 //Part "d"
24 //Simplfying the equation we get
25 //7.5 t^2-30t+5=0
26 a=7.5
27 b=-30
28 c=5
29 q=sqrt(b^2-4*a*c)
30 x1=(-b+q)/(2*a)
31 x2=(-b-q)/(2*a)
32 //As x1 is greater than 2 it does not hold as a
   solution
33 t=x2 //s
34 //Hence total time is
35 T=t1+t //s
36 clc
37 //Result
38 printf('The acceleration at 3s and 5s are %f ft/s^2
   and %f ft/s^2 respectively\n',a1,a2)
39 printf('The displacement is %f ft\n',s)
40 printf('The total time is %f s',T)

```

Scilab code Exa 12.6 Kin of a Part

```

1 //Initialization of variables
2 v_o=2 //m/s
3 y_o=120 //m

```



```

4 g=9.8 //m/s^2
5 //Calculations
6 //Solve using ground as datum
7 y=0
8 //Simplfying the equation
9 a=4.9
10 b=-2
11 c=-120
12 q=sqrt(b^2-4*a*c)
13 x1=(-b+q)/(2*a) //s
14 x2=(-b-q)/(2*a) //s
15 //Result
16 clc
17 printf('The time required is %f s',x1)
18 //As x2 is negative and negative time does not make
    any physical sense

```

Scilab code Exa 12.7 Kin of a Part

```

1 //Initilization of variables
2 Vo1=80 //ft/s
3 Vo2=60 //ft/s
4 g=32.2 //ft/s^2
5 //Calculations
6 //Simplfying by equating the two times
7 t=(-(Vo2*2)-(g*0.5*4))/(Vo1-Vo2-(g*0.5*4)) //s
8 //Substituting this t in s we get
9 s=(Vo1*t)-(0.5*g*t*t) //ft
10 //Result
11 clc
12 printf('The time obtained is %f s and the balls meet
    at %f ft',t,s)

```

Scilab code Exa 12.8 Kin of a Part

```
1 //Initialization of variables
2 theta=40 //degrees
3 x=100 //ft
4 ay=32.2 //ft/s^2
5 //Calculations
6 //Simplifying the equation
7 t=sqrt((tand(theta)*x)/(ay/2)) //s
8 //Velocity calculations
9 Vo=100/(cosd(theta)*t) //ft/s
10 //Result
11 clc
12 printf('The initial speed should be %f ft/s',Vo)
```

Scilab code Exa 12.9 Kin of a Part

```
1 //Initialization of variables
2 t=[0,1,2,3,4,5,6] //s
3 //Solving the Differential Equations we obtain
4 s=(t+1)^3 //ft
5 v=3*(t+1)^2 //ft/s
6 a=6*(t+1) //ft/s^2
7 //Plotting
8 subplot(221)
9 plot(t,s)
10 xlabel('t(s)')
11 ylabel('s(ft)')
12 subplot(222)
13 plot(t,v)
14 xlabel('t(s)')
15 ylabel('v(ft/s)')
16 subplot(223)
17 plot(t,a)
18 xlabel('t(s)')
```

```

19 ylabel('a(ft/s^2)')
20 //Result
21 clc
22 printf('The result are the gplots that have been
        generated')

```

Scilab code Exa 12.10 Kin of a Part

```

1 //Initilization of variables
2 t=3 //s
3 //Calculations
4 //After solving the differential equation
5 s=(1/3)*(t+2)^3 //ft
6 v=(t+2)^2 //ft/s
7 a=2*(t+2) //ft/s^2
8 //Result
9 clc
10 printf('The displacement, velocity and acceleration
        at t=3s are %f ft,%f ft/s and %f ft/s^2
        respectively ',s,v,a)

```

Scilab code Exa 12.12 Kin of a Part

```

1 //Initilization of variables
2 //Calling upward direction positive
3 xdot1=6 //ft/s
4 xdot3=3 //ft/s
5 xdoubledot=2 //ft/s^2
6 xdoubledot3=-4 //ft/s^2
7 //Calculations
8 xdot=-xdot1 //ft/s
9 xdot2=2*xdot-xdot3 //ft/s
10 xdoubledot2=2*xdoubledot-xdoubledot3 //ft/s^2

```

```
11 //Result
12 clc
13 printf('The value of velocity is %f ft/s and
         acceleration is %f ft/s^2',xdot2,xdoubledot2)
```

Scilab code Exa 12.14 Kin of a Part

```
1 //Initilization of variables
2 x=2 //m/s
3 //Differentiation constant
4 dc=3.6*2*2*2
5 //Calculations
6 //y=3.6*x^2
7 //Taking the derivative twice of both x and y
  quantities we get
8 a=dc //m/s^2
9 //The rest of the solution is theoretical hence not
  coded
10 //Result
11 clc
12 printf('The acceleration is:%f m/s^2',a)
```

Scilab code Exa 12.16 Kin of a Part

```
1 //Initilization of variables
2 t=4 //s
3 //Calculations
4 //Part (a)
5 x=t^3 //in
6 y=-2*t^2 //in
7 z=2*t //in
8 //Part (b)
9 //Theory question
```

```

10 //Part(c)
11 //Unit vector calculation
12 m=sqrt(4^2+1^1+(-3)^2)
13 e_1=[4/m,1/m,-3/m]
14 v=[3*t^2,-4*t,2] //in/s
15 //Projection of v on n at t=4s
16 dot=v.*e_1 //in/s
17 a=dot(1)+dot(2)+dot(3) //in/s
18 //Result
19 clc
20 printf('The co-ordinates of position are x=%fin,y=
    %fin,z=%fin\n',x,y,z)
21 printf('The projection of v on n at t=4s is %fin/s',
    a)

```

Scilab code Exa 12.17 Kin of a Part

```

1 //Initialization of variables
2 theta=%pi/3 //rad
3 //Calculations
4 //Method (a)
5 t=sqrt(theta) //s
6 r=2*theta
7 rdot=4*t
8 thetadot=2*t
9 //Velocity calculations
10 x=r*thetadot
11 v=sqrt((rdot)^2+x^2) //ft/s
12 //Theta calculations
13 thetax=30+atand(rdot/x) //degrees
14 //Method (b)
15 x=2*theta*cos(theta) //ft
16 y=2*theta*sin(theta) //ft
17 xdot=4*t*((cos(t^2)))+2*t^2*(-sin(t^2))*(2*t) //ft/s
18 ydot=4*t^2*sin(t^2)+2*t^2*cos(t^2)*2*t //ft/s

```

```

19 v=sqrt(xdot^2+ydots^2) //ft/s
20 thetax=atand(ydot/-xdot) //degrees
21 //Result
22 clc
23 printf('By both the methods we obtain v=%f ft/s and
        thetax as %f degrees',v,thetax)

```

Scilab code Exa 12.18 Kin of a Part

```

1 //Initilization of variables
2 theta=%pi/3 //rad
3 //Calculations
4 t=sqrt(theta) //s
5 thetadot=2*t
6 thetadoubledot=2
7 r=2*t^2
8 rdot=4*t
9 rdoubledot=4
10 ax=rdoubledot-(r*thetadoubledot*thetadoubledot) //ft
    /s^2
11 ay=2*rdot*thetadot+r*thetadoubledot //ft/s^2
12 a=sqrt(ax^2+ay^2)
13 thetax=30+atand(ax/ay) //degrees
14 //Solving by cartesian co-ordinate system yields
    same solution
15 //Result
16 clc
17 printf('The acceleration is %f ft/s^2 and thetax=%f
        degrees',a,thetax)
18 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 12.20 Kin of a Part

```

1 //Initilization of variables
2 theta=45 //degrees
3 l=0.5 //m
4 w=10 //rad/s
5 //Calculations
6 //PART a
7 //Here the theta derivative with respect to time is
   angular speed w
8 Vp1=l*(secd(theta)^2)*w //m/s
9 //Part b
10 //Radial Component
11 r=l*secd(theta)*tand(theta)*w //m/s
12 //Transverse Component
13 t=l*secd(theta)*w //m/s
14 //Total
15 Vp2=sqrt(r^2+t^2) //m/s
16 //Result
17 clc
18 printf('The velocity is:%fm/s\n',Vp1)
19 printf('The velocity in part b is %fm/s ',Vp2)

```

Scilab code Exa 12.21 Kin of a Part

```

1 //Initilization of variables
2 Va=5 //ft/s
3 theta=70 //degrees
4 l=6.24 //ft
5 //Calculations
6 Vb=-cotd(theta)*Va //ft/s
7 //Result
8 clc
9 printf('The value of Vb is %fft/s ',Vb)

```

Scilab code Exa 12.25 Kin of a Part

```
1 //Initialization of variables
2 theta=linspace(0,360,13)
3 //Calculations
4 //Defining everything in terms of matrices
5 t=(theta*%pi)/(180*6) //s converting degrees to
   radians
6 costheta=cosd(theta)
7 sintheta=sind(theta)
8 x=2*costheta //ft
9 v=-12*sintheta //ft/s
10 a=-72*costheta //ft/s^2
11 //Plotting
12 subplot(221)
13 plot(t,x)
14 xlabel('t(s)')
15 ylabel('Displacement x(ft)')
16 subplot(222)
17 plot(t,v,t,0)
18 xlabel('t(s)')
19 ylabel('Velocity v(ft/s)')
20 subplot(223)
21 plot(t,a)
22 xlabel('t(s)')
23 ylabel('Acceleration a(ft/s^2)')
24 //Result
25 clc
26 printf('The results are the plots')
```

Scilab code Exa 12.26 Kin of a Part

```
1 //Initialization of variables
2 d=1.2 //m
3 w0=0 //rpm
```



```

4 w=2000 //rpm
5 t=20 //s
6 // Calculations
7 alpha=(w-w0)/t
8 alpha_rad=(alpha*2*%pi)/60 //converting to radians/s
   ^2
9 //Result
10 clc
11 printf('The angular acceleration is %frad/s^2',
        alpha_rad)

```

Scilab code Exa 12.27 Kin of a Part

```

1 //Initilization of variables
2 w0=0 //rad/s
3 w=209 //rad/s
4 t=20 //s
5 // Calculations
6 theta=0.5*(w+w0)*t //rad
7 theta_rev=round(theta/(2*%pi)) //revolutions
   rounding off
8 //Result
9 clc
10 printf('The flywheel makes %i revolutions',theta_rev
        )

```

Scilab code Exa 12.28 Kin of a Part

```

1 //Initilization of variables
2 w0=0 //rad/s
3 alpha=10.5 //rad/s^2
4 t=0.6 //s
5 r=0.6 //m

```

```

6 // Calculations
7 w=w0+alpha*t //rad/s
8 v=r*w //m/s
9 a_t=r*alpha //m/s^2
10 a_n=r*w*w //m/s^2
11 a=sqrt(a_t^2+a_n^2) //m/s^2
12 phi=atand(a_t/a_n) //degrees
13 //result
14 clc
15 printf('The tangential velocity is %fm/s\n',v)
16 printf('the acceleration is %fm/s^2 and angle is %f
degrees ',a,phi)

```

Scilab code Exa 12.29 Kin of a Part

```

1 //Initialization of variables
2 l=4 //ft
3 wb=40 //rpm
4 we=60 //rpm
5 //Calculations
6 r=l/2 //ft
7 vb=r*((wb*2*%pi)/60) //ft/s
8 ve=r*((we*2*%pi)/60)
9 //Result
10 clc
11 printf('The linear speeds are %f ft/s and %f ft/s at
b and e respectively ',vb,ve)

```

Scilab code Exa 12.30 Kin of a Part

```

1 //Initialization of variables
2 wb=40 //rpm
3 we=60 //rpm

```

```

4 t1=5 //s using different symbol to avoid conflict in
      decleration
5 t=2 //s
6 r=2 //ft
7 //Calculations
8 alpha=((we*2*pi)/60)-((wb*2*pi)/60)/t1 //rad/s^2
9 w=((wb*2*pi)/60)+alpha*t //rad/s
10 //Components of acceleration are
11 a_t=r*alpha //ft/s^2
12 a_n=r*w^2 //ft/s
13 //result
14 clc
15 printf('The tangential and normal acceleration\n are
        %f rad/s^2 and %frad/s^2 respectively ',a_t,a_n)

```

Scilab code Exa 12.31 Kin of a Part

```

1 //Initalization of variables
2 d=200 //mm
3 w0=(800*2*pi)/60 //rpm
4 w=0 //rpm
5 t=600 //s
6 //Calculations
7 alpha=(w-w0)/t //rad/s^2 (deceleration)
8 //result
9 clc
10 printf('The angular acceleration is %frad/s^2\n The
        negative sign indicates that the wheel
        decelerates ',alpha)

```

Scilab code Exa 12.32 Kin of a Part

```

1 //Initalization of variables

```

```

2 //The symbols used here differ from the textbook
  solution to avoid conflict
3 t1=0 //s
4 t2=0.5 //s
5 t3=2.5 //s
6 t4=1/3 //s
7 w=200 //rpm
8 w0=0 //rpm
9 //Calculations
10 theta1=0.5*(w0+w/60)*t2 //rev
11 theta2=(w/60)*(t3-t2) //rev
12 theta3=0.5*(w/60+w0)*t4 //rev here the values of w
  and w0 are interchanged but essentially the value
  comes out to be the same hence the decleration
  has not been changed
13 theta=theta1+theta2+theta3 //rev
14 //Result
15 clc
16 printf('The wheel undergoes %f of rotations ',theta)

```

Scilab code Exa 12.34 Kin of a Part

```

1 //Initilization of variables
2 t=1 //s
3 r=4 //m
4 //Calculations
5 s=t^3+3 //m
6 theta=s/r //rad
7 dtheta_dt=0.75*t^2 //rad/s
8 Vx=-4*sin(theta)*dtheta_dt //m/s
9 Vy=4*cos(theta)*dtheta_dt //m/s
10 V=sqrt(Vx^2+Vy^2) //m/s
11 //Result
12 clc
13 printf('The components of velocity are Vx=%fm/s ,Vy

```

=%fm/s and $V=\%fm/s'$, V_x, V_y, V)

Scilab code Exa 12.35 Kin of a Part

```
1 //Initialization of variables
2 t=1 //s
3 theta=1 //rad
4 //Calculations
5 dtheta_dt=0.75*t^2 //rad/s
6 acc=1.5*t //rad/s^2
7 ax=-4*cos(theta)*dtheta_dt^2-(4*sin(theta)*acc) //m/
  s^2 (to left)
8 ay=-4*sin(theta)*dtheta_dt^2+(4*cos(theta)*acc) //m/
  s^2 (up)
9 a=sqrt(ax^2+ay^2) //m/s^2
10 //result
11 clc
12 printf('The acceleration is %fm/s^2',a)
```

Scilab code Exa 12.36 Kin of a Part

```
1 //Initialization of variables
2 t=2 //s
3 //Calculations
4 //Velocity
5 vx=8*t-3 //ft/s
6 vy=3*t^2 //ft/s
7 v=sqrt(vx^2+vy^2) //ft/s
8 theta_x=atand(vy/vx) //degrees
9 //Acceleration
10 ax=8 //ft/s^2
11 ay=6*t //ft/s^2
12 a=sqrt(ax^2+ay^2) //ft/s^2
```

```

13 phi_x=atand(ay/ax) //degrees
14 //Result
15 clc
16 printf('The velocity is %fft/s and angle is
        %fdegrees\n The acceleration is %fft/s^2 and
        angle it makes is %fdegrees',v,theta_x,a,phi_x)

```

Scilab code Exa 12.37 Kin of a Part

```

1 //Initilization of variables
2 V_ao=29.3 //ft/s
3 OA=50 //ft
4 theta=45 //degrees
5 OB=50*sqrt(2) //ft
6 //Calculations
7 w_ao=V_ao/OA //rad/s
8 V_bo=V_ao*cosd(theta) //ft/s
9 w_bo=V_bo/OB //rad/s
10 //Result
11 clc
12 printf('The angular velocity with respect to the
        observer is %frad/s\n The angular velocity after
        moving 50ft is %frad/s',w_ao,w_bo)

```

Scilab code Exa 12.38 Kin of a Part

```

1 //Initiliztaion of variables
2 theta=30 //degrees
3 r=[100*tand(theta),100] //ft
4 v=17.6 //ft/s
5 //Calculations
6 v_1=100*secd(theta)*secd(theta)
7 w=v/v_1 //rad/s (clockwise)

```

```

8 //result
9 clc
10 printf('The angular velocity is %frad/s',w)

```

Scilab code Exa 12.39 Kin of a Part

```

1 //Initilization of variables
2 t=2 //s
3 //Calculations
4 Vx=20*t+5 //m/s
5 Vy=t^2-20 //m/s
6 //As indefinite integral is not possible
7 x=10*t^2+5*t+5 //m
8 y=0.5*t^2-20*t-15 //m
9 ax=20 //m/s^2
10 ay=2*t //m/s^2
11 //Result
12 clc
13 printf('The displacement components are x=%fm,y=%fm\n
n The velocity components are Vx=%fm/s,Vy=%fm/s\n
The acceleration components are ax=%fm/s^2 and
ay=%fm/s^2',x,y,Vx,Vy,ax,ay)

```

Scilab code Exa 12.40 Kin of a Part

```

1 //Initilization of variables
2 d=0.1 //m
3 v=20 //m/s
4 a_g=6 //m/s^2
5 d2=0.150 //m
6 //Calculations
7 r=d/2 //m
8 w=v/r //rad/s

```

```

9 vb=d2*0.5*w //m/s
10 alpha=a_g/r //rad/s^2
11 a_t=d2*0.5*alpha //rad/s^2 tangential acceleration
12 a_n=d2*0.5*w*w //m/s^2 normal acceleration
13 a=sqrt(a_t^2+a_n^2) //m/s^2 linear acceleration
14 //Result
15 clc
16 printf('The linear velocity is %fm/s and the
acceleration is %fm/s^2',vb,a)

```

Scilab code Exa 12.41 Kin of a Part

```

1 //Initilization of variables
2 theta=40 //degrees
3 x=100 //ft
4 ax=0 //ft/s^2
5 ay=-32.2 //ft/s^2
6 //Calculations
7 //vox=vocos40....(1)
8 //voy=voy*t-1/2(32.2)t^2....(2)
9 //Simplyfying eq (1) and eq(2)
10 t_f=sqrt((x*tand(theta))/(0.5*(-ay))) //s time of
flight
11 Vo=x/(cosd(theta)*t_f) //ft/s
12 //As the max height occurs at half wat through the
flight
13 t=t_f/2 //s
14 ymax=Vo*sind(theta)*t+(0.5*ay*t*t) //ft the formula
has positive sign as ay is defined negative
15 //result
16 clc
17 printf('The max height the ball will reach is %f ft',
ymax)

```

Chapter 13

Dynamics of a Particle

Scilab code Exa 13.1 Dyna of a Part

```
1 //Initilization of variables
2 W=2 //lb
3 F=1.5 //lb
4 g=32.2 //ft/s^2
5 //Angles are with respect to the plane
6 theta1=10 //degrees
7 theta2=30 //degrees
8 //Calculations
9 //Now here the forces are considered as parallel and
   perpendicular to the plane
10 //Applying Newtond Principle
11 ax=(g/2)*(F*cosd(theta1)-(W*sind(theta2))) //ft/s^2
12 N1=(2*cosd(theta2)-(F*sind(theta1))) //lb
13 //result
14 clc
15 printf('The force on the particle is %flb\n The
   acceleration is %fft/s^2',N1,ax)
```

Scilab code Exa 13.2 Dyna of a Part

```

1 //Initialization of variables
2 m=5 //kg
3 s=12 //m
4 v=4 //m/s
5 vo=0 //m/s
6 g=9.8 //m/s^2
7 mu=0.25
8 //Calculations
9 //Using the kinematic equations of motion
10 a=(v^2-vo^2)/(2*s) //m/s^2
11 //Using Newtons Principle
12 N1=g*m //N
13 P=m*a+mu*N1 //N
14 //Result
15 clc
16 printf('The value of P is %fN',P)

```

Scilab code Exa 13.3 Dyna of a Part

```

1 //Initialization of variables
2 m=2 //kg
3 vo=0 //m/s
4 v=3 //m/s
5 s=0.8 //m
6 theta=20 //degrees
7 g=9.8 //m/s^2
8 //Calculations
9 N=m*g*cosd(theta) //N
10 a=(vo^2-v^2)/(2*s) //m/s^2
11 u=-((2*a)+(m*g*sind(theta)))/N
12 //Solving for return speed
13 //Symbol convention is different from textbook
14 a_ret=((m*g*sind(theta))-(u*N))/2 //m/s^2
15 vf=sqrt((2*a_ret*s)) //m/s
16 //Result

```

```
17 clc
18 printf('The speed is %fm/s',vf)
```

Scilab code Exa 13.4 Dyna of a Part

```
1 //Initilization of variables
2 W=1800 //lb
3 r=2000 //ft
4 v=58.7 //ft/s
5 g=32.2 //ft/s^2
6 //Calculations
7 F=(W*v*v)/(g*r) //lb
8 //Result
9 clc
10 printf('The frictional force to be exerted is %f lb',F)
```

Scilab code Exa 13.7 Dyna of a Part

```
1 //Initilization of variables
2 W=10 //lb
3 theta=30 //degrees
4 l=2 //ft
5 w=10 //rev/min
6 g=32.2 //ft/s^2
7 //Calculations
8 r=l*cosd(theta) //ft
9 a_n=r*((w*2*pi)/60)^2 //ft/s^2
10 //Applying Newtons Principle
11 //Solving by matrix method
12 A=[cosd(theta),-sind(theta);sind(theta),cosd(theta)]
13 B=[(W*a_n)/g;W]
14 C=inv(A)*B //lb
```

```
15 //Result
16 clc
17 printf('The value of T is %flb',C(1))
```

Scilab code Exa 13.8 Dyna of a Part

```
1 //Initilization of variables
2 m=4 //lb
3 v=6 //ft/s
4 r=2 //ft
5 theta1=40 //degrees
6 theta2=20 //degrees
7 g=32.2 //ft/s^2
8 //Calculations
9 a_n=v^2/r //ft/s^2
10 //Applying Newtons Principle
11 Fi=(m*a_n)/g //lb
12 //Solving by matrix method
13 A=[cosd(theta1),cosd(theta2);sind(theta1),-sind(
    theta2)]
14 B=[m;Fi]
15 C=inv(A)*B //lb
16 //Result
17 clc
18 printf('The value of T and C are %flb and %flb
    respectively',C(1),C(2))
```

Scilab code Exa 13.10 Dyna of a Part

```
1 //Initilization of variables
2 m1=2 //kg
3 theta=20 //degrees
4 m2=4 //kg
```

```

5 t=4 //s
6 g=9.8 //m/s^2
7 vo=0 //m/s
8 //Calculations
9 //Applying Newtons Principle
10 //Solving by matrix method
11 A=[1, -2; 1, 4]
12 B=[m1*g*sind(theta); m2*g]
13 C=inv(A)*B
14 a=C(2) //m/s^2
15 v=vo+a*t //m/s
16 //Result
17 clc
18 printf('The velocity of 4kg mass is %fm/s', v)

```

Scilab code Exa 13.11 Dyna of a Part

```

1 //Initialization of variables
2 m_A=20 //lb
3 m_B=60 //lb
4 u=0.3 //coefficient of friction
5 t=4 //s
6 theta1=30 //degrees
7 theta2=60 //degrees
8 g=32.2 //ft/s^2
9 vo=0 //ft/s
10 //Calculations
11 N1=m_A*cosd(theta1) //lb
12 N2=m_B*cosd(theta2) //lb
13 //Solving for T and a using matrix method
14 A=[1, -m_A/g; -1, -m_B/g]
15 B=[(m_A*sind(theta1)+u*N1); (-m_B*sind(theta2)+u*N2)]
16 C=inv(A)*B
17 a=C(2) //ft/s^2
18 v=vo+a*t //ft/s

```

```

19 //Result
20 clc
21 printf('The velocity is %fft/s',v)

```

Scilab code Exa 13.12 Dyna of a Part

```

1 //Initilization of variables
2 m_A=40 //kg
3 m_B=15 //kg
4 F=500 //N
5 g=9.8 //m/s^2
6 theta=30 //degrees
7 //Calculations
8 m=m_A+m_B //kg
9 a=(F-m*g*sind(theta))/(m) //m/s^2
10 //Summing forces parallel and perpendicular to the
    plane
11 //Simplfying equation (1) and (2)
12 Nb=m_B*g+(m_B*a*sind(theta)) //N
13 //Substituting this in eq(1)
14 u=-((m_B*g*cosd(theta)-(Nb*cosd(theta)))/(Nb*sind(
    theta)))
15 //Result
16 clc
17 printf('The value of u is %f',u)

```

Scilab code Exa 13.13 Dyna of a Part

```

1 //Initilization of variables
2 P=70 //N
3 m_A=16 //kg
4 u_AH=0.25 //coefficient of friction between Block A
    and Horizontal Plane

```

```

5 m_B=4 //kg
6 u_BH=0.5 //coefficient of friction between Block B
   and Horizontal Plane
7 theta=10 //degrees
8 g=9.8 //m/s^2
9 //Calculations
10 //Applying sum of forces to both the FBD's
11 //Solving by matrix method
12 A=[-cosd(theta),-u_AH,-m_A,0;-sind(theta),1,0,0;cosd
   (theta),0,-m_B,-u_BH;sind(theta),0,0,1]
13 B=[-P;m_A*g;0;m_B*g]
14 C=inv(A)*B //Ans
15 //Result
16 clc
17 printf('The Value of T is %fN',C(1) )

```

Scilab code Exa 13.14 Dyna of a Part

```

1 //Initilization of variables
2 theta=10 //degrees
3 v=10 //ft/s
4 v0=0 //ft/s
5 u=1/3 //coefficient of friction
6 g=32.2 //ft/s^2
7 //Calculations
8 //Equations of motion for box are
9 //Simplfying the equations by sybstitution
10 a=((u*cosd(theta))-sind(theta))*g //ft/s^2
11 //Time calculations
12 t=(v-v0)/a //s
13 //Result
14 clc
15 printf('The value of a is %fft/s^2\n The time
   required is %f seconds',a,t)

```

Scilab code Exa 13.15 Dyna of a Part

```
1 //Initialization of variables
2 g=9.8 //m/s^2
3 //Calculations
4 //Simplifying the equations we can solve for T2 and
   aA first to obtain the solution
5 //Solving by matrix method
6 A=[-1.5, -4; -3.5, 24]
7 B=[-4*g; -24*g]
8 C=inv(A)*B
9 T2=C(1) //N
10 T1=T2/2 //N
11 T3=T2/2 //N
12 //Acceleration calculations
13 a1=1*g-T1 //m/s^2
14 a2=(2*g-T1)/2 //m/s^2
15 a3=(3*g-T3)/3 //m/s^2
16 a4=(4*g-T3)/4 //m/s^2
17 //Tension in fixed cord
18 T_f=2*T2 //N
19 //Result
20 clc
21 printf('The acceleration values are a1=%f, a2=%f, a3=
   %f and a4=%f m/s^2\n The tension in the fixed
   cord is %fN', a1, a2, a3, a4, T_f)
```

Scilab code Exa 13.16 Dyna of a Part

```
1 //Initialization of variables
2 m1=14 //kg
3 m2=7 //kg
```



```

4 theta=45 //degrees
5 u_1=1/4 //coefficient of friction between mass 1 and
  plane
6 u_2=3/8 //coefficient of friction between mass 2 and
  plane
7 g=9.8 //m/s^2
8 //Calculations
9 //The equations of motion for m1 are
10 N1=m1*g*cosd(theta) //N
11 F1=u_1*N1 //N
12 //The equations of motion for m2 are
13 N2=m2*g*cosd(theta) //N
14 F2=u_2*N2 //N
15 //Now to get T and a we solve using matrix method
16 A=[-1,-m1;1,-m2]
17 B=[-(m1*g*sind(theta)-F1);-(m2*g*sind(theta)-F2)]
18 C=inv(A)*B
19 //Result
20 clc
21 printf('The Value of T is %fN',C(1))

```

Scilab code Exa 13.19 Dyna of a Part

```

1 //Initilization of variables
2 W=12 //oz
3 k=2 //oz/in
4 M=0.34 //kg
5 K=22 //N/m
6 g=32.2 //ft/s^2
7 //Calculations
8 //Part(a)
9 a=(k*W*g)/16
10 b=W/16
11 f=(1/(2*pi))*(sqrt(a/b)) //Hz for simplicity the
  numerator and denominator have been computed

```

```

    seperately as a and b
12 //Part(b)
13 F=(1/(2*%pi))*(sqrt(K/M)) //Hz
14 //Result
15 clc
16 printf('The frequency in part (a) is %f Hz and in
    part(b) is %f Hz',f,F)

```

Scilab code Exa 13.20 Dyna of a Part

```

1 //As the entire question is theoretical
2 //theta is directly computed
3 theta=acosd(2/3) //degrees
4 //result
5 clc
6 printf('The value of theta is %f degrees',theta)

```

Scilab code Exa 13.28 Dyna of a Part

```

1 //Initilization of variables
2 G=6.658*10^-8 //cm^3/g.s^2
3 //Calculations
4 G1=G*((3.281*10^2)/((2.205/32.2)*10^4)) //ft^3/slug-
    s^2
5 G2=G1 //ft^4/lb-s^4
6 //Result
7 G1
8 G2

```

Scilab code Exa 13.29 Dyna of a Part

```

1 //Initilization of variables
2 //Modifying the value of C without vo^2 in it
3 C=5000*5280
4 G=3.43*10^-8 //Gravatational Constant
5 M=4.09*10^23 //Mass of the Earth
6 a=5.31*10^8
7 //When the orbit is circular e=0
8 vo1=sqrt(a) //ft/s
9 //When the orbit is parabolic e=1
10 vo2=sqrt((C*a+G*M)/C) //ft/s
11 //Result
12 clc
13 printf('The value of vo1=%f is smaller than vo2=%f,
        hence the\n Satellite will enter a hyperbolic
        path and never return ',vo1,vo2)
14 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 13.30 Dyna of a Part

```

1 //Initilization of variables
2 r=3940+500 //mi
3 phi=0 //degrees
4 vo=36000 //ft/s
5 C=4440*5280*vo
6 G=3.43*10^-8
7 M=4.09*10^23 //kg
8 //Calculations
9 e=((C*vo)/(G*M))-1
10 //Result
11 clc
12 printf('The value of e=%f hence the path is
        Hyperbolic ',e)

```

Scilab code Exa 13.31 Dyna of a Part

```
1 //Initilization of variables
2 a=92.9*10^6 //mi
3 G=3.43*10^-8
4 T=365*24*3600 //s
5 c=5280
6 //Calculations
7 M=(4*pi^2*a^3*c^3)/(G*T^2) //slugs
8 //Result
9 clc
10 printf('The mass of the sun is %f slugs',M)
```

Chapter 14

Kinematics of a Rigid Body in Plane Motion

Scilab code Exa 14.2 Kin of rig body in PM

```
1 //Initialization of variables
2 d=500 //mm
3 wo=0 //rpm
4 w=300 //rpm
5 t=20 //s
6 t1=2 //s
7 //Calculations
8 alpha=(2*pi*(1/60)*(w-wo))/t //rad/s^2
9 w1=wo+alpha*t1 //rad/s
10 v=(d/(2*1000))*w1 //m/s
11 a_n=(d/(2*1000))*w1^2 //m/s^2
12 a_t=(d/(2*1000))*alpha //m/s^2
13 a=sqrt(a_n^2+a_t^2) //m/s^2
14 theta=acosd(a_n/a) //degrees
15 //Result
16 clc
17 printf('The computed values are\n alpha=%frad/s^2,w1
    =%frad/s,v=%frad/s\n a=%fm/s^2 and the angle made
    is %fdegrees ',alpha,w1,v,a,theta)
```

Scilab code Exa 14.3 Kin of rig body in PM

```
1 //Initilization of variables
2 s_BC=2 //m
3 s_C=2.5 //m
4 //Calculations
5 s_B=sqrt(s_BC^2+s_C^2) //m
6 theta=atand(s_BC/s_C) //degrees
7 //Result
8 clc
9 printf('The absolute displacement is %fm and the
        angle made by the vector is %fdegrees',s_B,theta)
```

Scilab code Exa 14.4 Kin of rig body in PM

```
1 //Initilization of variables
2 V_A=20 //mi/h
3 V_B=70 //mi/h
4 theta1=60 //degrees
5 phi=45 //degrees
6 //Result
7 //Vector's in matrix form
8 v_A=[-V_A*cosd(phi),V_A*sind(phi)] //mi/h
9 v_B=[V_B*cosd(theta1),V_B*sind(theta1)] //mi/h
10 a=v_A(1)+v_B(1) //mi/h
11 b=v_A(2)+v_B(2) //mi/h
12 v_ab=sqrt(a^2+b^2) //mi/h
13 theta=atand(b/a) //degrees
14 //The relative velocity v_ba is just different in
    sign while the magnitude stays the same
15 //Result
```

```

16 clc
17 printf('The relative velocity is %fmi/h making an
    angle %fdegrees ',v_ab,theta)

```

Scilab code Exa 14.9 Kin of rig body in PM

```

1 //Initilization of variables
2 l=2.5 //m
3 v_A=4 //m/s
4 a_A=5 //m/s^2
5 theta=30 //degrees
6 //Calculations
7 //Vector triangle yields v_a.b=2.93 m/s
8 v_ab=2.93 //m/s
9 w=v_ab/l //rad/s (clockwise)
10 //Ploygon yields alpha_a/b=2.75 m/s^2
11 alpha_ab=2.75 //m/s^2
12 alpha=alpha_ab/l //rad/s^2 (counterclockwise)
13 //Result
14 clc
15 printf('The value of angular velocity is %frad/s and
    that of angular acceleration is %frad/s^2',w,
    alpha)

```

Scilab code Exa 14.10 Kin of rig body in PM

```

1 //Initilization of variables
2 w=(2*pi*120)/60 //rad/s
3 l=24 //in
4 l_c=4 //in
5 th=30 //degrees
6 //Calculations
7 v=(l_c/12)*w //ft/s

```

```

8  betaa=asind((l_c*sind(th))/l) //degrees
9  theta=60-betaa //degrees
10 //Component of velocity along connecting rod is
11 v1=v*cosd(theta) //ft/s
12 v_p=v1/cosd(betaa) //ft/s
13 //Result
14 clc
15 printf('The absolute velocity is %fft/s',v_p)

```

Scilab code Exa 14.13 Kin of rig body in PM

```

1 //Initialization of variables
2 v_pc=3.68 //ft/s
3 l=2 //ft
4 //Calculations
5 w=v_pc/l //rad/s counterclockwise
6 //Result
7 clc
8 printf('The angular velocity is %frad/s',w)

```

Scilab code Exa 14.14 Kin of rig body in PM

```

1 //This problem is a combination of numerical and
   graphical solution
2 //The program only deals with the numerical solution
   parts the rest can be verified by graphical
   solution
3 //Initialization of variables
4 r=4/12 //ft
5 w=4*pi //rad/s
6 l=2 //ft
7 w2=1.84 //rad/s
8 //Calculations

```



```

9 ac_n=r*w^2 //ft/s^2
10 a_pc_n=l*w2^2 //ft/s^2
11 //Result
12 clc
13 printf('The value of ac_n is %fft/s^2 and that of
        a_pc_n is %fft/s^2',ac_n,a_pc_n)

```

Scilab code Exa 14.15 Kin of rig body in PM

```

1 //Initilization of variables
2 w_bc=10 //rad/s
3 AB=250 //mm
4 BC=150 //mm
5 AC=179 //mm
6 AD=200 //mm
7 theta1=45 //degrees
8 //Calculations
9 v_c=(BC/1000)*w_bc //m/s
10 AC=sqrt((AB^2+BC^2)-(2*AB*BC*cosd(theta1))) //m
11 betaa=asind((BC*sind(theta1))/AC) //degrees
12 gammaa=asind((AB*sind(theta1))/AC) //degrees answer
    in the textbook is incorrect
13 ang=60-betaa //degrees
14 CD=sqrt(AD^2+AC^2-(2*AD*AC*cosd(ang))) //mm
15 D=asind((AC*sind(ang))/CD) //degrees
16 theta=asind((AD*sind(D))/AC) //degrees
17 n=360-(theta+gammaa+90) //degrees
18 v_cd=v_c*cosd(n) //m/s
19 del=180-(90+D) //degrees
20 v_D=v_cd/cosd(del) //m/s
21 w_AD=v_D/(AD/1000) //rad/s
22 //Result
23 clc
24 printf('The angular Velocity of AD is %frac/s',w_AD
        ) //Negative sign indicates clockwise orientation

```

25 //Answer in the textbook is incorrect

Scilab code Exa 14.18 Kin of rig body in PM

```
1 //Initilization of variables
2 theta1=73.9 //degrees
3 V=900 //mm/s
4 theta2=60 //degrees
5 theta3=46.1 //degrees
6 //Calculations
7 BC=sqrt((350*350)+(86.6*86.6)) //mm
8 CD=400 //mm
9 v_cb=(V*sind(theta2))/(sind(theta1)) //mm/s
10 v_c=((V*sind(theta3)))/(sind(theta1)) //mm/s
11 w_dc=v_c/CD //rad/s
12 w_cb=v_cb/BC //rad/s
13 //Result
14 clc
15 printf('The angular velocities are w_dc=%frad/s,w_bc
    =%frad/s',w_dc,w_cb)
```

Scilab code Exa 14.19 Kin of rig body in PM

```
1 //Calculations
2 //After equating the i and j terms we obtain
    simplified equations
3 //Solving by matrix method
4 A=[346,86.7;200,-350]
5 B=[-3700;-1790]
6 C=inv(A)*B
7 //Result
8 clc
```

```

9 printf('The angular accelerations are alpha_DC=%frac
    /s^2 and alpha_BC=%frac/s^2 ',C(1),C(2))
10 //The signs only indicate that the originally
    assumed orientations are incorrect and are
    opposite to those assumed

```

Scilab code Exa 14.20 Kin of rig body in PM

```

1 //Initilization of variables
2 d=3 //m
3 w=8 //rad/s (clockwise)
4 alpha=4 //rad/s^2 (counterclockwise)
5 r=d/2 //m
6 //Calculations
7 vo=r*w //m/s
8 ao=r*alpha //m/s^2
9 //Here OB is r
10 OB=r //m
11 v_bo=OB*w //m/s
12 v_B=v_bo+vo //m/s
13 //Also
14 a_bo=r*alpha //m/s^2 (directed left)
15 a_bo_n=r*w^2 //m/s^2
16 a_h=ao+a_bo //m/s^2
17 a_v=a_bo_n //m/s^2
18 a_B=sqrt((a_h^2)+(a_v^2)) //m/s^2
19 phi=atand(a_h/a_v) //degrees
20 //Result
21 clc
22 printf('The linear velocity at B is %fm/s and the
    acceleration is %fn/s^2 making an angle of %f
    degrees with horizontal ',v_B,a_B,phi)

```

Scilab code Exa 14.21 Kin of rig body in PM

```
1 //Initilization of variables
2 OA=0.6 //m
3 w=8 //rad/s
4 theta=30 //degrees
5 v_0=12 //m/s
6 alpha=4 //rad/s^2
7 a_0=6 //m/s^2
8 //Calculations
9 //Velocity Calculations
10 v_A0=OA*w //m/s
11 v_Ah=v_A0*sind(theta)+v_0 //m/s horizontal component
12 v_Av=v_A0*cosd(theta) //m/s
13 v_A=sqrt((v_Ah^2)+(v_Av^2)) //m/s
14 phi=atand(v_Av/v_Ah) //degrees
15 //Acceleration Calculations
16 a_A0t=OA*alpha //m/s^2
17 a_A0n=OA*w^2 //m/s^2
18 a_Ah=-a_0-a_A0n*cosd(theta)-a_A0t*sind(theta) //m/s
    ^2
19 a_Av=-a_A0n*sind(theta)+a_A0t*cosd(theta) //m/s^2
20 a_A=sqrt((a_Ah^2)+(a_Av^2)) //m/s^2
21 phi2=atand(a_Av/a_Ah) //degrees
22 //Result
23 clc
24 printf('The velocity is %fm/s making an angle %f
    degrees with horizontal\n The acceleration is %fm
    /s^2 making an angle %fdegrees with horizontal',
    v_A,phi,a_A,phi2)
```

Scilab code Exa 14.22 Kin of rig body in PM

```
1 //Initilization of variables
2 AL=5 //ft
```

```

3 d=10 //ft displacement
4 //Calculations
5 theta=d/AL //radians
6 s_o=3*theta//ft
7 //Result
8 clc
9 printf('The displacement So is %i ft ',s_o)

```

Scilab code Exa 14.23 Kin of rig body in PM

```

1 //Initilization of variables
2 //Speed and acceleration at the center
3 v=12 //in/s
4 a=18 //in/s^2
5 //Calculations
6 v_D=((a+v*0.5)/a)*v //in/s
7 //Speed at point F
8 v_F=((v/2)/v)*v_D //in/s
9 //Acceleration at D
10 a_D=(24/a)*a //in/s^2
11 //Acceleration at F
12 a_F=((v/2)/v)*24 //in/s^2
13 //Result
14 clc
15 printf('The velocity and acceleration of weight A
        are %iin/s and %iin/s^2 respectively ',v_F,a_F)

```

Scilab code Exa 14.24 Kin of rig body in PM

```

1 //Calculations
2 //Speed and acceleration of D
3 sD=((18-6)/18)*12 //in/s
4 aD=(12/18)*18 //in/s^2

```

```

5 //Speed and acceleration of F
6 sF=(6/12)*8 //in/s
7 aF=(6/12)*12 //in/s^2
8 //Result
9 clc
10 printf('The velocity and acceleration of weight A
        are %iin/s and %iin/s^2 respectively ',sF,aF)

```

Scilab code Exa 14.26 Kin of rig body in PM

```

1 //Initilization of variables
2 v_BG=300 //mm/s
3 v_G=300 //mm/s
4 a_BGt=500 //mm/s^2
5 a_BGn=3600 //mm/s^2
6 a_Gh=500 //mm/s^2
7 a_Bv=1800 //mm/s^2
8 //Calculations
9 w=((75-25)/25)*6 //rad/s
10 alpha=((75-25)/25)*10 //rad/s^2
11 v_B=sqrt(v_BG^2+v_G^2) //mm/s
12 a_v=a_Bv-a_BGt //mm/s^2
13 a_h=a_BGn-a_Gh //mm/s^2
14 a_B=sqrt(a_v^2+a_h^2) //mm/s^2
15 //Result
16 clc
17 printf('The velocity and acceleration of point B are
        %imm/s and %imm/s^2 respectively ',v_B,a_B)

```

Chapter 15

Moment of Inertia

Scilab code Exa 15.11 MI

```
1 //Initialization of variables
2 y1=1 //in
3 y2=4 //in
4 d1=2.2-1 //in
5 d2=4-2.2 //in
6 A1=12 //in^2
7 A2=8 //in^2
8 b1=6 //in
9 b2=2 //in
10 h1=2 //in
11 h2=4 //in
12 //Calculations
13 y_bar=(A1*y1+A2*y2)/(A1+A2) //in
14 I1=(1/12)*(b1)*(h1^3) //in^4
15 I2=(1/12)*(b2)*(h2^3) //in^4
16 //Using Parallel Axes Theorem
17 I=(I1+A1*d1^2)+(I2+A2*d2^2) //in^4
18 //Result
19 clc
20 printf('The moment of inertia is %f in^4',I)
```

Scilab code Exa 15.12 MI

```
1 // Initialization of variables
2 d=60 //mm diameter of the hole
3 // Areas
4 At=100*100 //mm^2
5 Ab=200*100 //mm^2
6 Ac=((%pi/4)*d^2) //mm^2
7 bt=100 //mm
8 ht=100 //mm
9 bb=200 //mm
10 hb=100 //mm
11 //Distance of centroids of each area
12 yt=150 //mm
13 yb=50 //mm
14 yc=150 //mm
15 // Calculations
16 y_bar=((At*yt)+(Ab*yb)-(Ac*yc))/(At+Ab-Ac) //mm
17 // Distances
18 dt=yt-y_bar //mm
19 db=y_bar-yb //mm
20 dc=yc-y_bar //mm
21 // Values of Inertia
22 It=(1/12)*(bt)*(ht^3) //mm^4
23 Ib=(1/12)*(bb)*(hb^3) //mm^4
24 Ic=(1/4)*(%pi)*((d/2)^4) //mm^4
25 //Moment of inertia
26 I=(It+At*dt^2)+(Ib+Ab*db^2)-(Ic+Ac*dc^2) //mm^4
27 // Result
28 clc
29 printf('The moment of inertia is %f mm^4',I)
```

Scilab code Exa 15.14 MI

```
1 // Initialization of variables
2 b1=2 //in
3 b2=4 //in
4 h1=8 //in
5 h2=2 //in
6 bo=8 //in
7 ho=8 //in
8 bi=4 //in
9 hi=4 //in
10 // Calculations
11 I1=(1/12)*(b1)*(h1^3) //in^4
12 I2=(1/12)*(b2)*(h2^3) //in^4
13 I=2*(I1+I2) //in^4
14 Io=(1/12)*(bo)*(ho^3) //in^4
15 Ii=(1/12)*(bi)*(hi^3) //in^4
16 I_bar=Io-Ii //in^4
17 // Result
18 clc
19 printf('The moment of inertia is %f in^4',I_bar)
```

Scilab code Exa 15.15 MI

```
1 // Initialization of variables
2 b1=75 //mm
3 b2=12 //mm
4 h1=12 //mm
5 h2=162 //mm
6 d1=75 //mm
7 // Calculations
8 A=(h2*b2)+(2*b1*h1) //mm^2
9 I1=(1/12)*(b1)*(h1^3)+(b1*h1*d1^2) //mm^4
10 I2=(1/12)*(b2)*(h2^3) //mm^4
11 I_bar=2*I1+I2 //mm^4
```

```

12 k=sqrt(I_bar/A) //mm
13 //Result
14 clc
15 printf('The radius of gyration is %f mm',k)

```

Scilab code Exa 15.20 MI

```

1 //Initilization of variables
2 r=50 //mm
3 //Calculations
4 Ixy=(1/8)*(50^4) //mm^4
5 //Result
6 clc
7 printf('The moment of inertia is %f mm^4',Ixy)

```

Scilab code Exa 15.24 MI

```

1 //The notation has been changed for ease
2 //Calculations
3 x=(5*1*3.5+8*1*0.5)/(5*1+8*1) //in
4 y=(5*1*0.5+8*1*4)/13 //in
5 //Moment of inertia
6 Ix=(1/12)*(5)*(1^3)+(5*2.15*2.15)+(1/12)*(1*8^3)
   +(8*1.35^2) //in^4
7 Iy=(1/12)*(1)*(5^3)+(5*1.85*1.85)+(1/12)*(8)*(1^3)
   +(8*1.15^2) //in^4
8 Ixy=(8*1*(-1.15)*1.35)+(5*1*1.85*(-2.15)) //in^4
9 //Mohr circle calculations
10 d=0.5*(Ix+Iy) //distance to center of the circlce
11 r=sqrt((21^2)+(32.3^2))
12 maxI=d+r //in^4
13 theta=atand(32.3/21) //degrees maxI occurs at this
   angle

```

```

14 minI=d-r //in^r
15 //Result
16 clc
17 printf('The moment of inertias are as follows\n Ix=
    %f in^4,Iy=%f in^4,Ixy=%f in^4\n Imax=%fin^4 and
    Imin=%f in^4 ',Ix,Iy,Ixy,maxI,minI)

```

Scilab code Exa 15.25 MI

```

1 //Notations have been changed
2 //Calculations
3 x=-(25*125*0.5*125+25*100*0.5*25)/(25*125+25*100) //
    mm
4 y=(25*125*0.5*25+25*100*75)/5625 //mm
5 Iy=(1/12)*25*125^3+25*125*(62.5-40.3)^2+(1/12)
    *100*25^3+100*25*(40.3-12.5)^2 //mm^4
6 Ix=Iy //mm^4 for L-section
7 //The second computation checks the first
8 Ixy=(125*25*22.2*27.8)+(100*25*(-27.8)*(-34.7)) //mm
    ^4
9 //Mohr Circle analysis
10 Imax=Ix+Ixy //mm^4
11 Imin=Ix-Ixy //mm^4
12 //Result
13 clc
14 printf('The values of moment of inertia are\n Ix=
    %fmm^4,Iy=%fmm^4,Ixy=%fmm^4\n Imax=%fmm^4 and
    Imin=%fmm^4 ',Ix,Iy,Ixy,Imax,Imin)

```

Scilab code Exa 15.30 MI

```

1 //Initilization of variables
2 rho=490 //lb/ft^3

```

```

3 t=0.02 //in
4 d=4 //in
5 r=d/2 //in
6 g=32.2 //ft/s^2
7 //Calculations
8 W=(%pi*r^2*t*rho)/1728 //lb
9 //Mass
10 m=W/g //slugs
11 //Moment of inertia
12 I=(1/4)*m*(r/12)^2 //slug-ft^2
13 //Result
14 clc
15 printf('the moment of inertia is %fslug-ft^2',I)

```

Scilab code Exa 15.36 MI

```

1 //Initilization of variables
2 //The integration involves variables hence the
   direct formula is being used in this coding
3 m=500 //kg
4 R=0.25 //m
5 h=0.5 //m
6 //Calculations
7 Ix=(3/10)*m*R^2 //kg.m^2
8 Iy=(3/5)*m*((1/4)*R^2+h^2) //kg.m^2
9 //Result
10 clc
11 printf('Hence proved that Ix=%fkg-m^2 and Iy=%fkg-m
   ^2',Ix,Iy)

```

Scilab code Exa 15.37 MI

```

1 //Initilization of variables

```

```

2 del=450 //lb/ft^3
3 h1=9/12 //ft
4 h2=10/12 //ft
5 ro1=4/12 //ft
6 ri1=2/12 //ft
7 ro2=18/12 //ft
8 ri2=16/12 //ft
9 a=2.5/24 //ft
10 b=3.5/24 //ft
11 l=1 //ft
12 g=32.2 //ft/s^2
13 //Calculations
14 Whub=(%pi*ro1^2-%pi*ri1^2)*h1*del //lb
15 Wrिम=(%pi*ro2^2-%pi*ri2^2)*h2*del //lb
16 //For one spoke
17 Wspoke=(%pi*a*b*l*del) //lb
18 //Moment of inertia calculations
19 Ihub=0.5*(Whub/g)*(ro1^2+ri1^2) //lb-s^2-ft
20 Irim=0.5*(Wrिम/g)*(ro2^2+ri2^2) //lb-s^2-ft
21 Ispoke=6*((1/12)*(Wspoke/g)*l^2+(Wspoke/g)*h2^2) //
    lb-s^2-ft
22 Iwheel=Ihub+Irim+Ispoke //lb-s^2-ft
23 //result
24 clc
25 printf('The moment of inertia of the wheel is %flb-s
    ^2-ft ',Iwheel)

```

Chapter 16

Dynamics of a Rigid Body in Plane Motion

Scilab code Exa 16.2 Dyna of rig body in PM

```
1 //Intilization of variables
2 W=600 //lb
3 d=30 //in
4 theta=25 //degrees
5 g=32.2 //ft/s^2
6 //Calculations
7 m=W/g //lb-s^2/ft
8 //Moment of inertia
9 I=0.5*m*((d/2)/12)^2 //lb-s^2-ft
10 //Applying Newtons law and coservation of angular
    momentum and rolling
11 //Solving by matrix method
12 A=[1,m,0,0;0,0,0,1;((d/2)/12),0,-I,0;0,1,-((d/2)/12)
    ,0]
13 B=[W*sind(theta);W*cosd(theta);0;0]
14 C=inv(A)*B
15 //Result
16 clc
17 printf('The Frictional Force is %f lb and the
```

acceleration is %f ft/s²',C(1),C(2))

Scilab code Exa 16.3 Dyna of rig body in PM

```
1 //Initialization of variables
2 m=18 //kg
3 d=0.6 //m
4 vo=3 //m/s
5 theta=20 //degrees
6 g=9.8 //m/s2
7 //Calculations
8 //Moment of Inertia
9 I=0.5*m*(d/2)2 //
10 //Applying Newtons second Law a
11 A=[1,m,0,0;0,0,1,0;d/2,0,0,-I;0,1,0,(-d/2)]
12 B=[g*m*sind(theta);g*m*cosd(theta);0;0]
13 C=inv(A)*B
14 //Storing the answers in variables
15 F=C(1) //N
16 ax=C(2) //m/s2
17 Na=C(3) //N
18 alpha=C(4) //rad/s2
19 //Time Calculations
20 v=0 //m/s2
21 t=(vo)/ax //s
22 //Result
23 clc
24 printf('It takes %f s to reach the highest point of
    travel',t)
```

Scilab code Exa 16.5 Dyna of rig body in PM

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

```

2 m=20 //kg
3 F1=40 //N
4 ro=0.6 //m
5 ri=0.45 //m
6 g=9.8 //m/s^2
7 //Calculations
8 //Moment of inertia
9 I=(2/5)*m*ro^2 //kg-m^2
10 //Applying Newtons Law and conservation of angular
    Momentum
11 //Solving by matrix method
12 A=[1,m;ro,-I/ro]
13 B=[F1;F1*ri]
14 C=inv(A)*B
15 //Storing answers in variables
16 F=C(1) //N
17 a=C(2) //m/s^2
18 //Result
19 clc
20 printf('The acceleration is %f m/s^2 and F=%f N',a,F
    )
21 //The solution in the textbook is incorrect

```

Scilab code Exa 16.6 Dyna of rig body in PM

```

1 //Initilization of variables
2 W=16.1 //lb
3 u=0.10 //co-efficient of friction
4 g=32.2 //ft/s^2
5 theta=30 //degrees
6 F=1.39 //lb
7 //Calculations
8 //Applying Newtons Second Law
9 //Using F=1.39 lb
10 a=(W*sind(theta)-F)/(W/g) //ft/s^2

```



```

11 alpha=(F*0.5*5/2)/((W/g)*(0.5^2)) //rad/s^2
12 //Result
13 clc
14 printf('The value of a is %f ft/s^2 and alpha is %f
        rad/s^2.\n Hence the sphere will both roll and
        slip.',a,alpha)

```

Scilab code Exa 16.8 Dyna of rig body in PM

```

1 //Initilization of variables
2 theta=30 //degrees
3 W=80 //lb
4 Ww=100 //lb
5 I=4 //slug-ft^2
6 r=0.5 //ft
7 v= 20 //ft/s
8 vo=0 //ft/s
9 g=32.2 //ft/s^2
10 //Calculations
11 //Using Equations of motion
12 //Solving the system of linear equatinons by matrix
    method
13 A=[-1,0,-W/g;1,-1,-Ww/g;0,r,-2*I]
14 B=[-W;Ww*sind(theta);0]
15 C=inv(A)*B
16 //Storing values in variables
17 T=C(1) //lb
18 F=C(2) //lb
19 a=C(3) //ft/s^2
20 //Time calculations
21 t=(v-vo)/a //s
22 //Result
23 clc
24 printf('The time required is %f s',t)

```

Scilab code Exa 16.9 Dyna of rig body in PM

```
1 //Initilization of variables
2 M=70 //kg
3 ko=0.4 //m
4 ri=0.45 //m
5 ro=0.6 //m
6 theta=30 //degrees
7 m=35 //kg
8 g= 9.8 //m/s^2
9 //Calculations
10 I=M*ko^2 //kg-m^2
11 //Using Equations of motion
12 //Solving the equations by matrix method
13 A=[-1,-m*0.15,0;1,-M*ro,-1;-ri,-I,ro]
14 B=[-m*g;M*g*sind(theta);0]
15 C=inv(A)*B
16 F=C(3) //N
17 Na=M*g*cosd(theta) //N
18 //Required coefficient of friction
19 u=F/Na //coefficient of friction
20 //Result
21 clc
22 printf('The value of alpha is %f rad/s^2 and Tension
        is %f N\n F=%f N and Na=%f N also u=%f',C(2),C
        (1),F,Na,u)
```

Scilab code Exa 16.10 Dyna of rig body in PM

```
1 //Initilization of variables
2 m=200 //kg
3 g=9.8 //m/s^2
```

```

4 r=1.2 //m
5 F1=1000 //N
6 F2=1400 //N
7 // Calculations
8 N=m*g //N
9 I=(2/5)*(m)*r^2 //kg-m^2
10 //Using equations of motion
11 //Solving for F and alpha using matrix method
12 //Applying equations of motion
13 A=[1,-m;-r,-I/r]
14 B=[F1-F2;F1*r]
15 C=inv(A)*B
16 //Storing values
17 F=C(1) //N
18 alpha=C(2) //rad/s^2
19 a=r*alpha //m/s^2
20 //Result
21 clc
22 printf('The value of a is %f m/s^2 and F is %f N',a,
        F)
23 //The negative signs indicate that the direction is
        opposite to what was originally assumed

```

Scilab code Exa 16.11 Dyna of rig body in PM

```

1 //Initilization of variables
2 Wa=161 //lb
3 Wb=193.2 //lb
4 Wc=300 //lb
5 ka=3 //ft
6 kb=2.5 //ft
7 theta1=30 //degrees
8 theta2=45 //degrees
9 g=32.2 //ft/s^2
10 //Calculations

```

```

11 //Moment of inertia Calculations
12 Ia=(Wa/g)*ka^2 //lb-s^2-ft
13 Ib=(Wb/g)*kb^2 //lb-s^2-ft
14 //Using equations of motion for A and B and C
15 //Solving by matrix method
16 A=[1,1,-Wa/g,0,0;1,-4,-Ia*(1/4),0,0;-2,0,-Ib*(5/8)
    ,4,0;0,0,-(Wc/g)*(5/2),-1,-0.25;0,0,0,0,1]
17 B=[Wa*sind(theta1);0;0;-Wc*cosd(theta2);Wc*sind(
    theta2)]
18 C=inv(A)*B
19 //Storing values in the variables
20 T1=C(1) //lb
21 T2=C(4) //lb
22 a=C(3) //ft/s^2
23 //Result
24 clc
25 printf('The value of a is %f ft/s^2 and T1=%f lb and
    that of T2 is %f lb ',a,T1,T2)

```

Scilab code Exa 16.12 Dyna of rig body in PM

```

1 //Initilization of variables
2 W=644 //lb
3 F=30 //lb
4 theta=30 //degrees
5 r=1.5 //ft
6 g=32.2 //ft/s^2
7 //Calculations
8 //Using equations of motion
9 //Solving by matrix method
10 A=[1,-W/g;-r,-(1/2)*(W/g)*(2*2)*(1/r)]
11 B=[W*sind(theta)-F*cosd(theta);-F*2]
12 C=inv(A)*B
13 a=C(2) //ft/s^2
14 //Result

```

```
15 clc
16 printf('The value of a is %f ft/s^2',a)
```

Scilab code Exa 16.14 Dyna of rig body in PM

```
1 //Initilization of variables
2 W=20 //lb
3 g=32.2 //ft/s^2
4 vb=0.5 //rad/s
5 //Calculations
6 //Using equations of motion
7 //Solving the three equations simultaneously by
  matrix method
8 X=[0,1,-(W/g)*5.2;-1,0,-(W/g)*3;3,-3,-(1/12)*(W/g)
  *12^2]
9 Y=[-0.75*(W/g);(W/g)*1.3-W;0]
10 C=inv(X)*Y
11 A=C(1) //lb
12 B=C(2) //lb
13 alpha=C(3) //rad/s^2
14 //Result
15 clc
16 printf('The value of alpha is %f rad/s^2 and of A
  and B are %f lb \nand %f lb respectively',alpha,A
  ,B)
```

Scilab code Exa 16.16 Dyna of rig body in PM

```
1 //Initilization of variables
2 mc=7.25 //kg
3 d=0.9 //m
4 la=0.2 //m
5 ma=9 //kg
```

```

6 F=45 //N
7 ay=0 //m/s^2
8 g=9.8 //m/s^2
9 //Calculations
10 I=2*(0.5*mc*(d/2)^2)+0.5*ma*(la/2)^2 //kg-m^2
11 //Using the equations of motion
12 Na=(2*mc+ma)*g //N
13 //Simplfying using radial velocity formula
14 //Solving the two equations using matrix method
15 A=[-1,-(2*mc+ma);(d/2),-I/(d/2)]
16 B=[-F;F*(la/2)]
17 C=inv(A)*B
18 F=C(1) //N
19 ax=C(2) //m/s^2
20 //Result
21 clc
22 printf('The computation yields ax=%f m/s^2',ax)

```

Scilab code Exa 16.18 Dyna of rig body in PM

```

1 //Initilization of variables
2 r=0.05 //m cylinder radius
3 g=9.8 //m/s^2
4 //Calculations
5 //Here the equation has been solved in terms of the
  variables
6 //Hence we directly consider the final result
7 av=(2*g)/3 //m/s^2
8 //Result
9 clc
10 printf('The value of av is %f m/s^2',av)

```

Scilab code Exa 16.21 Dyna of rig body in PM

```

1 //initilization of variables
2 W=16.1 //lb
3 v=9 //ft/s
4 phi=30 //degrees
5 r=0.5 //ft
6 g=32.2 //ft/s^2
7 OG=4.5 //ft
8 //Calculations
9 //Using equations of motion
10 an=v^2/OG //ft/s^2
11 //Solving for alpha we get
12 N=(W/g)*an+W*cosd(phi) //lb
13 //Using equations of motion
14 A=[1,-r;-1,-r*r]
15 B=[W*sind(phi);0]
16 C=inv(A)*B
17 F=C(1) //lb
18 at=C(2) //ft/s^2
19 //Result
20 clc
21 printf('The value of N and F are %f lb and %f lb
        respectively ',N,F)

```

Scilab code Exa 16.24 Dyna of rig body in PM

```

1 //Initilization of variables
2 m_abc=20 //kg
3 m_cd=10 //kg
4 l_abc=3 //m
5 l_cd=2 //m
6 x=0.75 //m
7 y=1.5 //m
8 theta=60 //degrees
9 F=1000 //N
10 g=9.8 //m/s^2

```

```

11 //Simplfying constants
12 a=26
13 b=28.3
14 c=49
15 //Calculations
16 //After the rigorous simplification we arrive at the
    following
17 //Bx=26*alpha
18 //By=49-28.3*alpha
19 //Summing moments about A
20 alpha=(m_abc*g*x+F*y+c)/((1/3)*m_abc*l_abc^2+a*tand(
    theta)+b) //rad/s
21 //Result
22 clc
23 printf('The value of angular acceleration is: %frac/
    s^2',alpha)

```

Scilab code Exa 16.25 Dyna of rig body in PM

```

1 //Initilization of variables
2 W=50 //lb
3 P=10 //lb
4 t=5 //s
5 vo=0 //ft/s
6 g=32.2 //ft/s^2
7 //Calculations
8 //Using equations of motion
9 ax=(P*g)/W //ft/s^2
10 //Solving by matrix method for A and B
11 F=[1,1;-4,4]
12 Q=[W;P]
13 R=inv(F)*Q
14 //Velocity calculations
15 v=vo+ax*t //ft/s
16 A=R(1) //lb

```



```

17 B=R(2) //lb
18 //Result
19 clc
20 printf('The velocity of the door after 5s is %f ft/s
        and A=%f lb and B=%f lb ',v,A,B)

```

Scilab code Exa 16.26 Dyna of rig body in PM

```

1 //Initilization of variables
2 AB=2 //m
3 m=2 //kg
4 F=20 //N
5 g=9.8 //m/s^2
6 //Calculations
7 //Using equation of motion
8 a=F/m //m/s^2
9 //Solving by matrix method for Na and Nb
10 A=[1, -1;4/5,4/5]
11 B=[m*g;F*(3/5)]
12 C=inv(A)*B
13 //Result
14 clc
15 printf('The value of a is %f m/s^2 and the reactions
        are\n Na=%f N and Nb=%f N',a,C(1),C(2))

```

Scilab code Exa 16.27 Dyna of rig body in PM

```

1 //Initilization of variables
2 vo=0 //ft/s
3 //Calculations
4 s=(0.011*5280*2)/(2*0.004)
5 //Result
6 clc

```

```
7 printf('It travels %f ft before coming to rest',s)
8 //Answer in the textbook is incorrect by 20ft
```

Scilab code Exa 16.28 Dyna of rig body in PM

```
1 //Initilization of variables
2 u=0.3 //coefficient of friction
3 m=70 //kg
4 g=9.8 //m/s^2
5 //Calculations
6 //CASE 1
7 //Using equations of motion
8 Na=m*g //N
9 ah=(u*Na)/m //m/s^2
10 //CASE 2
11 //Applying sum of moments equal to zero
12 F=(Na*0.3)/1.2 //N
13 a_h=F/m //m/s^2
14 //Result
15 //Intutive insights can be attained after we get
    these results
16 clc
17 printf('The value of Na is %f N and that of
    acceleration are \ntwo values 1)%f m/s^2 2)%f m/s
    ^2 each for tipping and sliding respectively\n F
    is %f N',Na,ah,a_h,F)
```

Scilab code Exa 16.29 Dyna of rig body in PM

```
1 //Initilization of variables
2 m=60 //kg
3 me=660 //kg
4 a=6 //m/s^2
```

```

5 g=9.8 //m/s^2
6 //Calculations
7 //Using equations of motion
8 P=m*a+m*g //N
9 //Scale reading
10 R=P/g //kg
11 //Increase in mass
12 I=R-m //kg
13 //Tension
14 T=me*a+me*g //N
15 //Result
16 clc
17 printf('The value of P is %f N \n Apparent increase
    in weight is %f kg\n Tension in the cable is %f N
    . ',P,I,T)
18 //Answer in the textbook is off by 28 //N in Tension

```

Scilab code Exa 16.30 Dyna of rig body in PM

```

1 //Initilization of variables
2 u=0.2 //coefficient of friction
3 ma=1.2 //kg
4 mb=2 //kg
5 g=9.8 //m/s^2
6 //Calculations
7 Nb=mb*g //N
8 F=u*Nb //N
9 //Using equations of motion
10 //Solving for T and a
11 A=[-1,-ma;1,-mb]
12 B=[-ma*g;F]
13 C=inv(A)*B
14 T=C(1) //N
15 a=C(2) //m/s^2
16 //Taking the sum of the moments

```

```

17 x_m=-(F*0.15+T*0.15)/Nb //m
18 x=x_m*1000 //mm
19 //Result
20 clc
21 printf('The acceleration of block A is %f m/s^2 and
Nb acts at a distance %f mm\n Negative sign
indictaes that the side assumed is incorrect',a,x
)

```

Scilab code Exa 16.31 Dyna of rig body in PM

```

1 //Initilization of variables
2 a=2.5 //m/s^2
3 mA=3 //kg
4 mB=7 //kg
5 g=9.8 //m/s^2
6 //Calculations
7 F=(mA+mB)*a //N
8 //Using equations of motion
9 Py=mB*g //N
10 //Solving for Px and H
11 A=[1,1;-0.0375,0.0375]
12 B=[mB*a;Py*0.05]
13 C=inv(A)*B
14 Px=C(1) //N
15 H=C(2) //N
16 //Result
17 clc
18 printf('The value of H is %f N',H)

```

Scilab code Exa 16.32 Dyna of rig body in PM

```

1 //Initilization of variables

```

```

2 m=20 //kg
3 g=9.8 //m/s^2
4 vo=3 //m/s
5 v=0 //m/s
6 s=4 //m
7 //Calculations
8 //Using equations of motion
9 Na=m*g //N
10 F=(Na*0.075)/0.125 //N
11 a=F/m //m/s^2
12 //Displacement
13 d=-(v^2-vo^2)/(2*a) //m
14 displ=s-d //m
15 v_f=sqrt(2*a*displ) //m/s
16 //Result
17 clc
18 printf('The final velocity is %f m/s ',v_f)

```

Scilab code Exa 16.33 Dyna of rig body in PM

```

1 //Initilization of variables
2 mA=30 //kg
3 mB=45 //kg
4 u_ab=1/3 //coefficient of friction between two
   blocks
5 u_bp=1/10 //coefficient of friction between block
   and horizontal plane
6 g=9.8 //m/s^2
7 //Calculations
8 //By inspection
9 Na=mA*g //N
10 Nb=Na+mB*g //N
11 a=(u_ab*Na-u_bp*Nb)/mB //m/s^2
12 P=(mA*a+u_ab*Na) //N
13 //For block A

```

```

14 //Solving for P,F and a
15 A=[1,-1,-mA;-0.05,-0.075,0;0,1,-mB]
16 B=[0;-Na*0.050;Nb*u_bp]
17 C=inv(A)*B
18 P_new=C(1) //N
19 //Result
20 //As p < p_new
21 clc
22 printf('The maximum value of P is %f N',P)

```

Scilab code Exa 16.34 Dyna of rig body in PM

```

1 //Initilization of variables
2 Vo=1.5 //m/s
3 V=0 //m/s
4 g=9.8 //m/s^2
5 //Calculations
6 a=(g*0.2)/0.75 //m/s^2
7 t=-(V-Vo)/a //s
8 //Result
9 clc
10 printf('The maximum accelerayion is %f m/s^2 and
    minimum time is %f s',a,t)

```

Scilab code Exa 16.36 Dyna of rig body in PM

```

1 //Initilization of variables
2 vo=0 //mi/h
3 v=60 //mi/h
4 t=13.8 //s
5 W=3385 //lb
6 xb=46 //in
7 xf=66 //in

```

```

8 xv=31 //in
9 g=32.2 //ft/s^2
10 //Calculations
11 a=((v*88*60)/3600)-vo)/t //ft/s^2
12 //Summing horizontal forces
13 F=(W/g)*a //lb
14 //Solving for Rf and Rr
15 A=[1,1;-xf,xb]
16 B=[W;-F*xv]
17 C=inv(A)*B
18 Rr=C(1) //lb
19 Rf=C(2) //lb
20 //Result
21 clc
22 printf('The value of reactions are Rf=%f lb and Rr=
%f lb ',Rf,Rr)

```

Scilab code Exa 16.43 Dyna of rig body in PM

```

1 //Initilization of variables
2 W=161 //lb
3 F=16.1 //lb
4 r=18 //ft radius
5 t=2 //s
6 g=32.2 //ft/s^2
7 wo=0 //rad/s
8 //Calculations
9 //Using equations of motion
10 //Solving for T and alpha
11 A=[r/12,-0.5*(W/g)*(r/12)^2;-1,-F/g]
12 B=[0;-F]
13 C=inv(A)*B
14 alpha=C(2) //rad/s^2
15 w=wo+alpha*t //rad/s
16 //Result

```

```

17 clc
18 printf('The angular speed is %f rad/s',w)
19 //The decimal accuray causes the discrepancy

```

Scilab code Exa 16.47 Dyna of rig body in PM

```

1 //Initilization fo variables
2 r=2000 //ft
3 g=32.2 //ft/s^2
4 d=4.71 //ft
5 v=176 //ft/s
6 //Calculations
7 e=(d*v^2)/(g*r) //ft
8 //Result
9 clc
10 printf('The superelevation is %f ft',e)
11 //Watch the unit in the final answer

```

Scilab code Exa 16.48 Dyna of rig body in PM

```

1 //Initilization of variables
2 a=5 //ft/s^2
3 C=50 //lb-ft
4 W=161 //lb
5 g=32.2 //ft/s^2
6 //Calculations
7 T=0.5*(W/g)*1^2*a+C //lb
8 Ox=-T*(2/sqrt(a)) //lb
9 Oy=T*(1/sqrt(a))+W //lb
10 Wa=T/(1-(a/g)) //lb
11 //Result
12 clc

```



```
13 printf('The values are T=%f lb ,Wa=%f lb ,Ox=%f lb and
        Oy=%f lb ',T,Wa ,Ox ,Oy)
```

Scilab code Exa 16.49 Dyna of rig body in PM

```
1 //Initilization of variables
2 m=100 //kg
3 mr=20 //kg
4 w=8 //rad/s
5 l1=300 //mm
6 l2=600 //mm
7 g=9.8 //m/s^2
8 //Calculations
9 r_bar=(mr*l1+m*750)/120 //mm
10 I=(1/3)*mr*(l2/1000)^2+(2/5)*m*(l1/2000)^2+m*(0.75)
    ^2 //kg.m^2
11 alpha=(m+mr)*g*(r_bar/1000)/I //rad/s^2
12 On=(m+mr)*(r_bar/1000)*w^2 //N
13 Ot=((m+mr)*(r_bar/1000)*alpha)-(m+mr)*g //N
14 //Result
15 clc
16 printf('The angular acceleration is %f rad/s^2 and
        On=%f N and Ot=%f N',alpha,On,Ot)
17 //Due to decimal accuracy there is discrepancy in
    answers with the textbook
```

Scilab code Exa 16.50 Dyna of rig body in PM

```
1 //Initilization of variables
2 W=40 //lb
3 w=10 //rad/s
4 alpha=2 //rad/s^2
5 r=2 //in
```

```

6 g=32.2 //ft/s^2
7 //Calculations
8 //Using equations of motion
9 On=(W/g)*(1/6)*w^2 //lb
10 Ot=(W/g)*(1/6)*alpha
11 Io=(0.5*(W/g)*0.5^2)*2+((W/g)*(1/6)^2)*2
12 //Result
13 clc
14 printf('The reaction components are On=%f lb and Ot=
    %f lb ',On,Ot)

```

Scilab code Exa 16.51 Dyna of rig body in PM

```

1 //Initilizatin of variables
2 W=6 //lb
3 l=8 //ft
4 v=10 //ft/s
5 g=32.2 //ft/s^2
6 theta1=60 //degrees
7 theta2=30 //degrees
8 //Calculations
9 Fe=(W*v^2)/(g*l*0.5) //lb
10 //Using equations of motion
11 //Solving for C and T
12 A=[cosd(theta1),-cosd(theta2);cosd(theta2),cosd(
    theta1)]
13 B=[-Fe;W]
14 P=inv(A)*B //lb
15 C=P(1) //lb
16 T=P(2) //lb
17 //Result
18 clc
19 printf('The value of C is %f lb and T is %f lb ',C,T)

```

Scilab code Exa 16.52 Dyna of rig body in PM

```
1 //Initilization of variables
2 W=32.2 //lb
3 T=120 //lb
4 m=1 //slug
5 r=6/12 //ft
6 //Calculations
7 w=sqrt((T*(3/5)*4)/(m*r*3)) //rad/s
8 //Result
9 clc
10 printf('The angular speed permissible is %f rad/s',w
    )
```

Scilab code Exa 16.53 Dyna of rig body in PM

```
1 //Initilization of variables
2 m=30 //kg
3 k=0.45 //m
4 g=9.8 //m/s^2
5 //Using equations of motion
6 //Solving for T1,T2 and alpha
7 A=[1,0,-m;0,-1,-45;-0.6,0.3,-m*k^2]
8 B=[50*g;-150*g;0]
9 C=inv(A)*B
10 //Result
11 clc
12 printf('The values are T1=%f N T2=%f N and alpha=%f
    rad/s^2 ',C(1),C(2),C(3))
```

Scilab code Exa 16.54 Dyna of rig body in PM

```
1 //Initilization of variables
2 Wc=28 //lb
3 v=16 //ft/s
4 Ib=12 //ft-lb-s^2
5 u=0.4 //coefficient of friction
6 t=2 //s
7 g=32.2 //ft/s^2
8 //Calculations
9 T=Wc+(Wc/g)*8 //lb
10 alpha=8/(15/12) //rad/s^2
11 F=(Ib*alpha+T*1.25)/t //lb
12 N=F/u //lb
13 //Summing moments about D
14 P=(N*8+F*3)/40 //lb
15 //Summing forces horizontally and vertically
16 Dx=151-P //lb
17 Dy=-F //lb
18 //Result
19 clc
20 printf('The reactions at D are Dx=%f lb and Dy=%f lb
        ',Dx ,Dy)
```

Scilab code Exa 16.55 Dyna of rig body in PM

```
1 //Initilization of variables
2 m=8 //kg
3 n=90 //rpm
4 g=9.8 //m/s^2
5 //Calculations
6 Fg=m*g //N
7 w=2*%pi*n/60 //rad/s
8 //using equations of motion
9 By=m*g //N
```

```

10 //Solving for Bx and C
11 A=[1,1;-0.3,0.9]
12 B=[m*0.3*w^2;By*0.3]
13 C=inv(A)*B //N
14 //Result
15 clc
16 printf('The solution is Bx=%f N ,By=%f N and C=%f N'
        ,C(1),By,C(2))

```

Scilab code Exa 16.56 Dyna of rig body in PM

```

1 //Initilization of variables
2 m=8 //kg
3 n=90 //rpm
4 g=9.8 //m/s^2
5 r=0.3 //m
6 //calculations
7 w=2*pi*n/60 //rad/s
8 //Using equations of motion
9 C=(m*g*0.3+m*r*w^2*r)/1.2 //N
10 Bx=C-m*r*w^2 //N
11 By=m*g //N
12 //Result
13 clc
14 printf('The solution is Bx=%f N ,By=%f N and C=%f N'
        ,Bx,By,C)

```

Scilab code Exa 16.57 Dyna of rig body in PM

```

1 //Initilization of variables
2 Na=294 //N
3 Nb=735 //N
4 //Calculations

```

```

5 a=(1/10*Nb-1/3*Na)/45 //m/s^2
6 P=(1/3*Na)-30*a //N
7 //result
8 clc
9 printf('The solution is P=%f N and a=%f m/s^2',P,a)

```

Scilab code Exa 16.58 Dyna of rig body in PM

```

1 //Initilization of variables
2 W=50 //lb
3 g=32.2
4 //Calculations
5 //Using equations of motion
6 a=(10/(W/g)) //ft/s^2
7 B=((2.5*(W/g)*a)+4*W-1.5*10)/8 //lb
8 A=50-B //lb
9 //Result
10 clc
11 printf('The solution is A=%f lb B=%f lb and a=%f ft/
    s^2',A,B,a)

```

Scilab code Exa 16.60 Dyna of rig body in PM

```

1 //Initilization of variables
2 g=9.8 //m/s^2
3 r1=0.3 //m
4 m1=20 //kg
5 m2=100 //kg
6 r2=0.75 //m
7 //Calculations
8 alpha=(m1*g*r1+m2*g*r2)/(m1*r1^2+(m1/12)*0.6^2+m2*r2
    ^2+(2/5)*m2*0.15^2) //rad/s^2
9 //Result

```

```

10 clc
11 printf('The angular acceleration is %f rad/s2',
        alpha)

```

Scilab code Exa 16.61 Dyna of rig body in PM

```

1 //Initilization of variables
2 r=15/12 //ft
3 W=600 //lb
4 theta=25 //degrees
5 //calculations
6 ax=(r*W*sind(theta))/((1/r)*14.5+r*18.6) //ft/s2
7 F=(W*sind(theta))-18.6*9.09 //lb
8 //Result
9 clc
10 printf('The solution is F=%f lb and ax=%f ft/s2',F,
        ax)

```

Scilab code Exa 16.62 Dyna of rig body in PM

```

1 //Initilization of variables
2 m=7 //kg
3 g=9.8 //m/s2
4 r=0.5 //m
5 I=0.875 //kg.m2
6 //Calculations
7 //Solving for alpha and T
8 alpha=(m*g*r)/(I+m*r*0.5) //rad/s2
9 T=(I*alpha)/r //N
10 //Result
11 clc
12 printf('The souldion is alpha =%f rad/s2 and T=%f N
        ',alpha,T)

```


Chapter 17

Work and Energy

Scilab code Exa 17.4 Work

```
1 //Initilization Of Variables
2 s1=2 //Lower Limit of the Integral
3 s2=5 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 k=20 //lb/in
6 //Calculation
7 //Using Trapezoidal Rule for Intergration
8 function [I1]=Trap_Composite1(f,s1,s2,n)
9     h=(s2-s1)/n
10    s=linspace(s1,s2,n+1)
11    I1=(h/2)*((2*sum(f(s)))-f(s(1))-f(s(n+1)))
12 endfunction
13 deff(' [y]=f(s) ', 'y=k*s ')
14 //Result
15 clc
16 printf('The work done is %f in-lb ',Trap_Composite1(f
    ,s1,s2,n) )
```

Scilab code Exa 17.6 Work

```

1 //Initialization of variables
2 m=5 //kg
3 d=6 //m
4 theta1=30 //degrees
5 theta2=10 //degrees
6 u=0.2 //coefficient of friction
7 g=9.8 //m/s^2
8 F=70 //N
9 //Calculations
10 //Using free body diagram
11 Na=m*g*cosd(theta1)-(F*sind(theta2)) //N
12 //work done by each force
13 W=[F*cosd(theta2),-m*g*sind(theta1),0,-u*Na*d] //N.m
14 //Total Work Done
15 W_tot=W(1)+W(2)+W(3)+W(4) //N.m
16 //Using resultant
17 R=F*cosd(theta2)-(u*Na)-(m*g*sind(theta1)) //N
18 W_d=R*d //N.m (Work Done)
19 //Result
20 clc
21 printf('The work done is %f N.m',W_d)

```

Scilab code Exa 17.7 Work

```

1 //Initialization of variables
2 m=20 //kg
3 d=1.5 //m
4 theta=30 //degrees
5 u=0.25 //coefficient of friction
6 g=9.8 //m/s^2
7 F=130 //N
8 //Calculations
9 W=F*d-(m*g*sind(theta)*d) //N.m
10 //Result
11 clc

```

```
12 printf('The work done is %i N.m',W)
```

Scilab code Exa 17.9 Work

```
1 //Initialization of variables
2 d=6/12 //ft
3 l=8/12 //ft
4 l_c=3.2 //in
5 y=1.82 //in^2
6 //Calculations
7 V=(1/4)*%pi*d^2*l //ft^3
8 //One horizontal inch
9 h_i=V/l_c //ft^3
10 //One vertical inch
11 v_i=100*144 //lb/ft^2
12 //Then 1.82 in^2 represents
13 x=y*v_i*h_i //ft-lb
14 //Result
15 clc
16 printf('The work capacity is %f ft-lb',x)
```

Scilab code Exa 17.10 Work

```
1 //Initialization of variables
2 speed=90000 //m/h
3 P=100*1000 //N
4 //Calculations
5 Power=P*((speed)/3600) //J/s
6 //Result
7 clc
8 printf('The power developed is %fJ/s',Power)
```

Scilab code Exa 17.11 Work

```
1 // Initialization of variables
2 d=0.6 //m
3 T_t=800 //N
4 T_s=180 //N
5 w=200 //rpm
6 // Calculations
7 r=d/2 //m radius
8 // Torque
9 M=(T_t-T_s)*r //N.m
10 // Power
11 w_new=(2*pi*w)/60 //rad/s
12 Power=M*(w_new) //W
13 // Result
14 clc
15 printf('The power transmitted is %f W',Power)
```

Scilab code Exa 17.12 Work

```
1 // Initialization of variables
2 P=25.6 //lb
3 w=600 //rpm
4 a=36 //in
5 b=12 //in
6 // Calculations
7 M=P*(((b/2)+a)/12) //lb-ft
8 w_new=(2*pi*w)/60 //rad/s
9 Hp=(M*w_new)/550 //hp
10 // Result
11 clc
12 printf('The power being transmitted is %fhp',Hp)
```

Scilab code Exa 17.13 Work

```
1 // Initilization of variables
2 Pout=3.8 //bhp
3 Pin=4.1 //ihp
4 // Calculations
5 Efficiency=round((Pout/Pin)*100) // Percent
6 // Result
7 clc
8 printf('The efficiency of the engine is %ipercent',
        Efficiency)
```

Scilab code Exa 17.15 Work

```
1 // Initilization Of Variables
2 a=3 //Lower Limit of the Integral
3 b=6 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 g=9.8 //m/s^2
6 w=4/16
7 // Calculation
8 //Using Trapezoidal Rule for Intergration
9 function [I1]=Trap_Composite1(f,a,b,n)
10     h=(b-a)/n
11     x=linspace(a,b,n+1)
12     I1=(h/2)*((2*sum(f(x)))-f(x(1))-f(x(n+1)))
13 endfunction
14 deff(' [y]=f(x) ', 'y=-3*x^-1')
15 an=-Trap_Composite1(f,a,b,n) //ft-lb
16 v=sqrt((an*g)/(0.5*w)) //ft/s
17 // Result
```

```
18 clc
19 printf('The speed of the disk is %fft/s',v)
20 //The answer in the textbook is incorrect
```

Scilab code Exa 17.16 Work

```
1 //Initilization of variables
2 l=2 //m
3 m=4 //kg
4 w_1=20 //rpm
5 w_2=50 //rpm
6 rev=10 //no of revolution
7 //Calculations
8 Io=(1/3)*(m)*l^2 //kg.m^2
9 w1=(2*%pi*w_1)/60 //rad/s
10 w2=(2*%pi*w_2)/60 //rad/s
11 theta=2*%pi*rev //rad
12 M=(0.5*Io*(w2^2-w1^2))/theta //N.m
13 //result
14 clc
15 printf('The constant moment required is %fN.m',M)
```

Scilab code Exa 17.18 Work

```
1 //Initilization of variables
2 W=1000 //lb
3 w_w=200 //lb weight of the individual wheel
4 d_w=2.5 //ft diameter of the wheel
5 v=22 //ft/s
6 t=2 //minutes
7 //Calculations
8 //T1=Initial Kinetic Energy and T2=Final Kinetic
  Energy
```

```

9 F=(-0.5*W*32.2^-1*v^2-4*0.5*w_w*32.2^-1*(v^2+0.5*v
    ^2))/(10560) //lb
10 //Negative sign in the answer tells it opposes the
    motion
11 //Result
12 clc
13 printf('The rolling resistance is %flb',F)

```

Scilab code Exa 17.19 Work

```

1 //Initilization of variables
2 W=100 //lb
3 lo=4 //ft
4 theta=45 //degrees
5 g=32.2 //ft/s^2
6 l=8/3 //ft
7 //Calculations
8 //Taking moment about point O and equating it to
    zero
9 alpha=(W*(lo*0.5)*cosd(theta))/((W/g)*(l)*2) //rad/s
    ^2
10 //Summing forces in the t direction
11 Ot=(W*cosd(theta))-((W/g)*lo*0.5*alpha) //lb
12 //Work Done
13 Work=W*(lo*0.5*cosd(theta)) //ft/lb
14 //Moment of inertia
15 Io=(1/3)*(W/g)*(lo^2) //kg-ft^2
16 //Using the concept for work done=chane in K.E
17 w=sqrt(Work/(0.5*Io)) //rad/s
18 //Summing forces along the bar
19 On=-(-(W/g)*lo*0.5*w^2)-(W*cosd(theta)) //lb
20 //Result
21 clc
22 printf('The bearing reaction at O on the rod is %flb
    ',On)

```

Scilab code Exa 17.21 Work

```
1 //Initilization of variables
2 vo=9 //m/s
3 theta=30 //degrees
4 g=9.8 //m/s^2
5 //Calculations
6 x=((7/10)*vo^2)/(g*sind(theta)) //m
7 //Result
8 clc
9 printf('The ball will roll %f m up the plane',x)
10 //The textbook wrongly mentions the unit of
    displacement as in
```

Scilab code Exa 17.22 Work

```
1 //Initilization of variables
2 W=322 //lb
3 F=12 //lb
4 a=0 //lower limit (where the cyliner starts rolling)
5 b=%pi/2 //Upper Limit (where the cyliner stops
    rolling)
6 d=3.2 //ft
7 g=32.2 //ft/s^2
8 //Calculations
9 dR=1.6 //Differential Radius
10 d_U=2*dR*F //differential work done
11 //Integration Calculations
12 //As it is a simple integration we can resort to
    this
13 U=d_U*(b-a) //ft-lb
```



```

14 //Determination of K.E
15 w=sqrt(U/((0.5*(W/g)*(1/(d/2)^2))+((0.5*0.5)*(W/g)*(
    d/2)^2))) //rad/s
16 //Result
17 clc
18 printf('the angular velocity of the is %f rad/s',w)

```

Scilab code Exa 17.23 Work

```

1 //Initilization of variables
2 m=90 //kg
3 k=450 //N/m
4 lo=0.6 //m
5 r=0.15 //m
6 x=0.9 //m
7 y=0.4 //m
8 //Calculations
9 //Initial KE=0
10 I=0.5*m*r^2 //kg.m^2
11 s1=sqrt((lo^2)+(x^2)) //m
12 s2=sqrt((lo^2)+(y^2)) //m
13 V1=0.5*k*(s1-lo)^2 //N.m
14 V2=0.5*k*(s2-lo)^2 //N.m
15 //Applying Conservation of Energy
16 w=sqrt((V1-V2)/(0.5*m*r^2+0.5*I)) //rad/s
17 //Result
18 clc
19 printf('The value of angular speed is: %f rad/s',w)

```

Scilab code Exa 17.24 Work

```

1 //Initilization of variables
2 Wa=161 //lb

```

```

3 Wb=193.2 //lb
4 Wc=322 //lb
5 v1=5 //ft/s
6 lc=6 //in
7 k=6 //lb/ft
8 l=4 //ft
9 u=0.2 //coefficient of friction
10 g=32.2 //ft/s^2
11 //Calculations
12 Ib=(1/2)*(Wb/g)*(1/2)^2 //Moment of inertia
13 w1=v1/0.5 //rad/s
14 T1=(0.5*(Wc/g)*v1^2)+(0.5*Ib*w1^2)+(0.5*(Wa/g)*v1^2)
    //ft-lb
15 //Work Done on the system
16 //The textbook is ambiguous on the calculations
    hence the result is dispalyed directly
17 U=26.4 //ft-lb
18 //Velocity Calculations
19 v=sqrt((T1+U)/9) //ft/s
20 //Result
21 printf('The velocity of the block is %f',v)

```

Scilab code Exa 17.25 Work

```

1 //Initilization of variables
2 Mm=70 //kg
3 Mc=45 //kg
4 R=0.6 //m
5 g=9.8 //m/s^2
6 l=5 //m
7 theta=50 //degrees
8 //Calculations
9 //T2 calculations except for v term in it as it
    cannot be declared as a number
10 T2=68.7 //without the v term in it

```

```

11 v=sqrt((g*Mm*l-g*Mc*l*sind(theta))/T2) //m/s
12 //Result
13 clc
14 printf('The speed is %fm/s',v)

```

Scilab code Exa 17.26 Work

```

1 //The textbook has a typo in printing the question
  number
2 //Initilization of variables
3 W1=96.6 //lb
4 W2=128.8 //lb
5 v=8 //ft/s
6 g=32.2 //ft/s^2
7 theta=30 //degrees
8 //Calculations
9 //Initial KE of the system is T1=0
10 T2=(0.5*(W1/g)*v^2)+(0.5*(W2/g)*(v/2)^2) //ft-lb
11 //Work Done without s term
12 U=-(W1*sind(theta))+W2*0.5
13 //S calculations
14 s=T2/U //ft
15 //Result
16 clc
17 printf('The block attains a speed of 8ft/s in %f ft',
  ,s)

```

Scilab code Exa 17.28 Work

```

1 //Initilization Of Variables
2 a=0 //Lower Limit of the Integral
3 b=6 //Upper Limit of the Integral
4 n=10 //Interval of the integral

```

```

5 m=50 //kg
6 l=6 //m
7 g=9.8 //m/s^2
8 //Calculation
9 //Gravataional Force is
10 Fg=g*(m/l) //dx
11 //Using Trapezoidal Rule for Intergration
12 function [I1]=Trap_Composite1(f,a,b,n)
13     h=(b-a)/n
14     t=linspace(a,b,n+1)
15     I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
16 endfunction
17 deff(' [y]=f(t) ', 'y=Fg*(6-t) ')
18 //Result
19 clc
20 printf('The Work done is %f N.m',Trap_Composite1(f,a
    ,b,n))

```

Scilab code Exa 17.31 Work

```

1 //Initilization Of Variables
2 x1=150 //mm
3 x2=450 //mm
4 a=0 //Lower Limit of the Integral
5 b=(x2-x1) //Upper Limit of the Integral
6 n=10 //Interval of the integral
7 k=0.044 //N/m
8 //Calculation
9 //Using Trapezoidal Rule for Intergration
10 function [I1]=Trap_Composite1(f,a,b,n)
11     h=(b-a)/n
12     t=linspace(a,b,n+1)
13     I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
14 endfunction
15 deff(' [y]=f(t) ', 'y=k*t ')

```

```
16 //Result
17 clc
18 printf('The Work done is %f N.m',Trap_Composite1(f,a
    ,b,n)/1000)
```

Scilab code Exa 17.32 Work

```
1 //Initilization of variables
2 m=10 //kg
3 d=1.2 //m
4 g=9.8 //m/s^2
5 //Calculations
6 //Initilial KE is zero
7 //Final KE is (without v^2 term in it)
8 KE2=(3/4)*10
9 //Work Done
10 U=m*g*d //N.m
11 //Velocity calculations
12 v=sqrt(U/KE2) //m/s
13 //Result
14 clc
15 printf('The velocity is %fm/s',v)
```

Scilab code Exa 17.33 Work

```
1 //Initilization of variables
2 W=161 //lb
3 wa=150 //lb
4 wb=100 //lb
5 la=2 //ft
6 lb=4 //ft
7 //Calculations
8 //Work Done
```

```

 9 T1=wb*lb-wa*la //ft-lb
10 //Final KE=zero
11 T2=0 //ft-lb
12 //Work Done on the system=T2-T1
13 //Hence the equation becomes
14 //50x-50x^2+100=0
15 //where
16 a=-50
17 b=50
18 c=100
19 //Solution
20 d=sqrt(b^2-4*a*c)
21 x1=(-b+d)/(2*a) //ft
22 x2=(-b-d)/(2*a) //ft
23 //Result
24 clc
25 printf('The stretch of the spring is %f',x2)
26 //Here even x1 could have been the solution ,but the
    stretch in the string is elongation not
    compression hence x2 is the valid answer

```

Scilab code Exa 17.34 Work

```

1 //Initilization of variables
2 I=100 //slug-ft^2
3 w=4 //rad/s
4 theta=6 //rad
5 Mc=64.4 //lb
6 g=32.2 //ft/s^2
7 //Calculations
8 vb=2*w //ft/s
9 vc=0.5*w //ft/s
10 Mb=(0.5*I*w^2+0.5*(Mc/g)*vc^2+0.5*Mc*theta)/(2*theta
    -(0.5*vb^2*(1/g))) //lb
11 //Result

```

```
12 clc
13 printf('The weight of the block B is %f lb',Mb)
```

Scilab code Exa 17.35 Work

```
1 //Initialization of variables
2 Wa=96.6 //lb
3 Wb=128.8 //lb
4 g=32.2 //ft/s^2
5 I=12 //slug-ft^2
6 v=16 //ft/s
7 ratio=1/3 //ratio of Sb/Sa
8 r=3//ft
9 va=6 //ft/s
10 vb=2 //ft/s
11 //Calculations
12 //Work Done without S in it
13 W=Wa-(ratio*Wb)
14 //System has zero KE initially and final KE is given
    by
15 w=va/r //rad/s
16 T2=(0.5*(Wa/g)*va^2+0.5*I*w^2+0.5*(Wb/g)*vb^2) //ft-
    lb
17 //Distance Calculations
18 S=T2/W //ft
19 //Result
20 clc
21 printf('The distance through which A falls is %f ft',
    ,S)
```

Scilab code Exa 17.36 Work

```
1 //initilization of variables
```

```

2 u=0.25 //coefficient of friction
3 k=2800 //N/m
4 x=0.075 //m
5 g=9.8 //m/s^2
6 m=7 //kg
7 theta=30 //degrees
8 //Calculations
9 //Normal Reaction
10 N=g*m*cosd(theta) //N
11 //Frictional Force
12 Fr=u*N //N
13 //Component of force along the plane
14 F=g*m*sind(theta) //N
15 //Spring work is
16 W=0.5*k*x*x //N.m
17 s=(W+Fr*x-F*x)/(F-Fr) //m
18 S=round(s*1000) //mm
19 //Result
20 clc
21 printf('The value of S is %i mm',S)

```

Scilab code Exa 17.37 Work

```

1 //Initialization of variables
2 m=5 //kg
3 l=2 //m
4 k=10000 //N/m
5 x=0.1 //m
6 g=9.8 //m/s^2
7 //Calculations
8 drop=l+x //m mass drop length
9 //Work Done by Gravity
10 Wg=g*m*drop //N.m
11 //Work Done by Spring
12 Ws=0.5*k*x^2 //N.m

```



```

13 //Increase in KE is without v^2
14 KE=0.5*m //kg
15 //Velocity Calculations
16 v=sqrt((Wg-Ws)/KE) //m/s
17 //Result
18 clc
19 printf('The speed is %f m/s',v)

```

Scilab code Exa 17.38 Work

```

1 //Initilization of variables
2 l=6 //ft
3 k=20 //lb/in
4 x=8 //in
5 //Calculations
6 //Work Done by Gravity
7 Wg=(1*12+x) //in without W
8 //Work Done by Spring
9 Ws=0.5*k*x^2 //in-lb
10 //Change in the kinetic energy is zero
11 W=Ws/Wg //lb
12 //Result
13 clc
14 printf('The weight is %i lb',W)

```

Scilab code Exa 17.40 Work

```

1 //Initilization of variables
2 W=8 //lb
3 //Calculations
4 //work done by the spring woithout k
5 Ws=0.5*((9/12)^2-(1/12)^2)
6 //Work done by gravity

```

```
7 Wg=W*(10.5/12) //ft-lb
8 //Change in KE is zero
9 k=Wg/Ws //lb/ft
10 //Result
11 clc
12 printf('The value of k is %f lb/ft ',k)
```

Scilab code Exa 17.41 Work

```
1 //Initialization of variables
2 Wc=100 //lb
3 r= 1 //ft
4 F=80 //lb
5 k=50 //lb/ft
6 s=6 //in
7 g=32.2 //ft/s^2
8 //Calculations
9 //Work done on the system
10 U=-0.5*k*(1)+F*(s/12) //ft-lb
11 //Initial KE is zero
12 Vo=sqrt(U/(0.5*(Wc/g+0.5*(Wc/g)*r))) //ft/s
13 //Result
14 clc
15 printf('The initial speed is %f ft/s ',Vo)
```

Chapter 18

Impulse and Momentum

Scilab code Exa 18.8 Imp

```
1 //Initialization of variables
2 W=100 //lb
3 u=0.2 //coefficient of friction
4 t=5 //s
5 v1=5 //ft/s
6 v2=10 //ft/s
7 g=32.2 //ft/s^2
8 ll=0 //lower limit of integration
9 ul=5 //upper limit of integration
10 //Calculations
11 Fr=u*W //lb
12 //Using The impulse momentum theorem
13 //Since the integration is just subtraction of
    limits we can skip that
14 F=((W/g)*v2-(W/g)*v1+Fr*ul)/ul //lb
15 //Result
16 clc
17 printf('The Force is %f lb ',F)
```

Scilab code Exa 18.9 Imp

```
1 //Initialization of variables
2 m=10 //kg
3 theta=30 //degrees
4 u=0.3 //coefficient of kinetic friction
5 t=5 //s
6 g=9.8 //m/s^2
7 //Calculations
8 //Asthere is no motion in the vertical direction
9 //Summing forces along vertical direction
10 Na=m*g*cosd(theta) //N
11 //Using impulse momentum theorem
12 vx=(m*g*sind(theta)-u*Na)*(t/m) //m/s
13 //Result
14 clc
15 printf('The speed after 5s is %f m/s',vx)
```

Scilab code Exa 18.10 Imp

```
1 //Initialization Of Variables
2 a=1 //Lower Limit of the Integral
3 b=5 //Upper Limit of the Integral
4 n=10 //Interval of the integral
5 W=80 //lb
6 us=0.25 //coefficient of static friction
7 uk=0.20 //coefficient of kinetic friction
8 g=32.2 //ft/s^2
9 //Calculation
10 //Limiting Force
11 F=W*uk //lb
12 //Using Trapezoidal Rule for Intergration
13 function [I1]=Trap_Composite1(f,a,b,n)
14     h=(b-a)/n
15     t=linspace(a,b,n+1)
```

```

16     I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
17 endfunction
18 deff(' [y]=f(t) ', 'y=t*20 ')
19 //Using Impulse momentum theorem
20 l=Trap_Composite1(f,a,b,n) //storing integration
    value in a variable
21 v=(g/W)*(1-F*(b-a)) //ft/s
22 //Result
23 clc
24 printf('The speed of the block is %f ft/s',v)

```

Scilab code Exa 18.11 Imp

```

1 //Initilization of variables
2 m=1.5 //kg
3 i1=2 //Integral constant obtained after integrating
4 i2=3.33 //Integral constant obtained after
    integrating
5 //Calculations
6 //As indefinite integrals are not possible to code
7 //We directly consider the intergral which is a
    simple integral
8 //v=t^2-1.11*t^3
9 //After we derivate this expression we obtain
10 t=i1/i2 //s
11 //Now using the formula for v we get
12 v=t^2-((i2/3)*t^3) //ft/s
13 //Result
14 clc
15 printf('The maximum velocity is:%f m/s',v)

```

Scilab code Exa 18.12 Imp

```

1 //Initialization of variables
2 m1=40 //kg
3 m2=10 //kg
4 m3=15 //kg
5 v0=2.5 //m/s
6 vf=5 //m/s
7 t=12 //s
8 u=0.1 //coefficient of friction
9 g=9.8 //m/s^2
10 theta=45 //degrees
11 //Calculations
12 //Applying Impulse–Momentum Theorem
13 P=((m1+m2+m3)*(vf-v0))+(t*(-m2*g*sind(theta)+u*g*m2
    *cosd(theta)+u*g*m3+g*m1))/t //N
14 //Result
15 clc
16 printf('The value of P is %f N',P)

```

Scilab code Exa 18.13 Imp

```

1 //Initialization of variables
2 W1=4 //lb
3 W2=2 //lb
4 t2=0.04 //s
5 W3=-2 //lb
6 t3=0.02 //s
7 t=3 //s
8 g=32.2 //ft/s^2
9 //Calculations
10 //Algebraic sum of two areas
11 A=t2*W2+t3*W3 //lb-s
12 //Using Impulse Momentum Theorem
13 v=(A*g)/W1 //ft/s
14 //Result
15 clc

```

```
16 printf('The speed after 3s is %f ft/s',v)
```

Scilab code Exa 18.14 Imp

```
1 //Initialization of variables
2 f_r=1 //in/s rate of fall of mercury
3 l1=18 //in length of left column
4 l2=22 //in length of right column
5 rho=850 //lb/ft^3
6 d=1/4 //in
7 g=32.2 //ft/s^2
8 //Calculations
9 //Applying Impulse momentum theorem
10 M=((d*pi*d^2*4)/12^3)*(rho/g)*(1/12) //lb-s
11 //Result
12 clc
13 printf('The upward momentum of mercury is %f lb-s',M
    )
```

Scilab code Exa 18.15 Imp

```
1 //Initialization of variables
2 m=2000 //kg
3 k=1.200 //m
4 w=120 //rpm
5 t=200 //s
6 //Calculations
7 //Applying Angular Momentum theorem
8 M=((m*k^2*(w*2*pi))/60)/t //N.m
9 //Result
10 clc
11 printf('The Momentum necessary is %f N.m',M)
```

Scilab code Exa 18.17 Imp

```
1 // Initialization of variables
2 m1=5 //kg
3 m2=7 //kg
4 mp=5 //kg
5 r=0.6 //m
6 k=0.45 //m
7 vi=3 //m/s
8 vf=6 //m/s
9 g=9.8 //m/s^2
10 // Calculations
11 I=m1*k^2 //kg.m^2
12 wnet=(vf/r)-(vi/r) //rad/s
13 //Solving the system of linear equations
14 //Simplifying the equation we get
15 t=((I*wnet)+m1*(vf-vi)+m2*(vf-vi))*r/(r*(m2-m1)*g)
    //s
16 //Result
17 clc
18 printf('The time required is %f s',t)
19 //Decimal accuracy causes discrepancy in answers
```

Scilab code Exa 18.18 Imp

```
1 // Initialization of variables
2 m=50 //kg
3 vo=4 //m/s
4 vf=8 //m/s
5 t=6 //s
6 g=9.8 //m/s^2
7 r=0.8 //m
```



```

8 u=0.25 //coefficient of friction
9 I=30 //kg-m^2
10 //Calculations
11 Na=m*g //N
12 F=u*Na //N
13 //Angular Speeds
14 wo=vo/r //rad/s
15 wf=vf/r //rad/s
16 //Applying impulse momentum theorem
17 mb=(I*wf+m*vf*r-I*wo-m*vo*r+F*r*t)/(vo*r+g*r*t-vf*r)
    //kg
18 //Result
19 clc
20 printf('The mass of block B is %f kg',mb)

```

Scilab code Exa 18.19 Imp

```

1 //Initialization of variables
2 Ws=250 //lb
3 W1=500 //lb
4 W3=161 //lb
5 W4=64.4 //lb
6 wo=100 //rpm
7 wf=300 //rpm
8 r1=3 //ft
9 rs=2 //ft
10 g=32.2 //ft/s^2
11 //Calculations
12 //Moment Of Inertia
13 I=(0.5*(W1/g)*r1^2+0.5*(Ws/g)*rs^2) //slug-ft^2
14 //Change in angular Momentum
15 change1=I*((wf-wo)*2*(%pi/60)) //lb-s-ft
16 //Change in angular momentum about G for 161lb
17 change2=2*((W3/g)*(wf-wo)*(4/60)*%pi) //lb-s-ft
18 //Similarly change in 64lb is

```

```

19 change3=3*((W4/g)*(wf-wo)*(6/60)*%pi) //lb-s-ft
20 //Change in linear impulse
21 //Without t term in it
22 m1=2*W3
23 m2=-3*W4
24 //Total angular impulse
25 t=(change1+change2+change3)/(m1+m2) //s
26 //Result
27 clc
28 printf('The time required is %f s',t)

```

Scilab code Exa 18.21 Imp

```

1 //Initialization of variables
2 d=3 //ft
3 W=300 //lb
4 theta=20 //degrees
5 F=250 //lb
6 t=6 //s
7 vo=0 //ft/s
8 g=32.2 //ft/s^2
9 //Calculations
10 //Applying linear impulse momentum theorem
11 //Solving by matrix method
12 A=[-W/g,1*t;-(0.5*(W/g)*d^2*0.5^2)/(d/2),-t*d/2]
13 B=[-F*t+W*sind(theta)*t;-F*(d/2)*6]
14 C=inv(A)*B
15 //Result
16 clc
17 printf('The speed after 6s is %f ft/s',C(1))

```

Scilab code Exa 18.22 Imp

```

1 //Initilization of variables
2 theta=30 //degrees
3 vo=20 //ft/s
4 r=4 //ft
5 vf=0 //ft/s
6 g=32.2 //ft/s^2
7 //Calculations
8 wo=vo/r //rad/s
9 wf=vf/r //rad/s
10 //Applying impulse momentum theorem
11 //Solving simultaneous equations
12 t=-((3/(2*g))*(r^2)*(wf-wo))/(r*sind(theta))//s
13 //Result
14 clc
15 printf('The time t is %f s',t)

```

Scilab code Exa 18.23 Imp

```

1 //Initilization of variables
2 mw=75 //kg
3 k=0.9 //m
4 wi=10 //rad/s
5 wf=6 //rad/s
6 r=1.2 //m
7 m=30 //kg
8 g=9.8 //m/s^2
9 //Calculations
10 //Initial speed
11 vi=-r*wi //m/s
12 vf=r*wf //m/s
13 //Initial speed of B is
14 vib=-0.8*wi+vi //m/s
15 //Similarly
16 vfb=12 //m/s
17 //Applying impulse momentum theorem

```

```

18 //Solving by matrix method
19 A=[1, -1, -(mw*(vf-vi));0.8, 1.2, -(mw*(k^2)*(wf+wi))
    ; -1, 0, -(m*(vfb-vi))]
20 B=[0;0; -g*m]
21 C=inv(A)*B
22 //Here t is calculated as 1/t for simplicity
23 //Result
24 clc
25 printf('The time required is %f s',1/C(3))
26 //Decimal accuracy causes discrepancy in answers

```

Scilab code Exa 18.24 Imp

```

1 //Initialization of variables
2 d=8 //in
3 W=96.6 //lb
4 w=36 //rad/s
5 u=0.15 //coefficient of friction
6 g=32.2 //ft/s^2
7 //Calculations
8 r=(d/2)/12 //m
9 N=W //lb
10 F=u*N //lb
11 m=W/g //slugs
12 I=0.5*m*(r^2) //slug-ft^2
13 //Applying the impulse momentum theorem
14 //Solving the two equations simultaneously
15 A=[F, -m; F*r, I*(1/r)]
16 B=[0; w*I]
17 C=inv(A)*B
18 //Distance travelled
19 s=0.5*C(2)*C(1) //ft
20 t=C(1) //s
21 //Result
22 printf('The time required is %f s, and it travels %f

```

ft',t,s)

Scilab code Exa 18.25 Imp

```
1 //Initilisation of variables
2 d=2/12 //ft
3 v=80 //ft/s
4 g=32.2 //ft/s^2
5 //Calculations
6 //Mass flow reate without time
7 m=(1/4)*%pi*d^2*v*(62.4/g)
8 //Let P=force of plate on mass m of water
9 P=m*(0-v) //lb
10 //Result
11 clc
12 printf('The force water exerts on the plate is %f lb
        ',-P )
```

Scilab code Exa 18.26 Imp

```
1 //Initilization of variables
2 v1=20 //ft/s
3 vw=80 //ft/s
4 d=2/12 //ft
5 g=32.2 //ft/s^2
6 //Calculations
7 v=vw-v1 //ft/s
8 //mass flow rate without t
9 m=(1/4)*(%pi*d^2)*(62.4/g)*v
10 //Applying impulse momentum theorem
11 P=m*v //lb
12 //Result
13 clc
```

```
14 printf('The force exerted by water on the plate is
    %f lb ',P)
```

Scilab code Exa 18.27 Imp

```
1 //Initilisation of variables
2 W=150 //lb
3 v=20 //ft/s
4 A=0.2 //in^2
5 t=60 //s
6 g=32.2 //ft/s^2
7 //Calculations
8 //Mass flow
9 m=(A/12^2)*v*(62.4/g)
10 //Force
11 F=m*(0-v) //lb
12 //At t=60s the tank holds
13 Ww=(A/12^2)*v*t*62.4 //lb
14 //Total reading on scale
15 S=-F+W+Ww //lb
16 //Result
17 clc
18 printf('The scale reading is %f lb ',S)
```

Scilab code Exa 18.28 Imp

```
1 //Initilisation of variables
2 W=150 //lb
3 v=20 //ft/s
4 A=0.2 //in^2
5 t=60 //s
6 g=32.2 //ft/s^2
7 //Calculations
```

```

8 //Mass flow
9 m=(A/12^2)*v*(62.4/g)
10 //Force
11 F=m*(0-v) //lb
12 //At t=60s the tank holds
13 Ww=(A/12^2)*v*t*62.4 //lb
14 //Total reading on scale
15 S=-F+W+Ww //lb
16 //Result
17 clc
18 printf('The scale reading is %f lb',S)

```

Scilab code Exa 18.29 Imp

```

1 //Initilization of variables
2 Wp=130 //lb
3 Wb=150 //lb
4 Wbullet=2/16 //lb
5 g=32.2 //ft/s^2
6 vbullet=1200 //ft/s
7 //Calculations
8 v=((-Wbullet/g)*vbullet)/((Wb+Wp)/g) //ft/s
9 //Result
10 clc
11 printf('The speed of the boat is %f ft/s',v)
12 //Negative sign indicates direction opposite to that
    of bullet

```

Scilab code Exa 18.30 Imp

```

1 //Initilization of variables
2 mb=0.06 //kg
3 ms=50 //kg

```

```

4 h=0.03 //m
5 g=9.8 //m/s^2
6 // Calculations
7 //Speed of bag+bullet
8 v2=sqrt(2*g*h) //m/s
9 //Applying conservation of momentum
10 v1=((mb+ms)*v2)/mb //m/s
11 //Result
12 clc
13 printf('The speed of bullet as it entered the bag
        was %f m/s ',v1)

```

Scilab code Exa 18.32 Imp

```

1 //Initilization of variables
2 mb=0.06 //kg
3 vb=500 //m/s
4 mblock=5 //kg
5 vblock=30 //m/s
6 // Calculations
7 //Applying conservation of momentum
8 v=(mb*vblock+mblock*vblock)/(mb+mblock) //m/s
9 //Result
10 clc
11 printf('The speed of the system is %f m/s ',v)

```

Scilab code Exa 18.33 Imp

```

1 //Initilization Of Variables
2 W1=2 //lb
3 W2=3 //lb
4 a=0 //Lower Limit of the Integral
5 b=2 //Upper Limit of the Integral

```



```

6 n=10 //Interval of the integral
7 k=12/12 //lb/ft
8 g=32.2 //ft/s^2
9 //Calculation
10 //Work Done by the spring
11 //Using Trapezoidal Rule for Intergration
12 function [I1]=Trap_Composite1(f,a,b,n)
13     h=(b-a)/n
14     t=linspace(a,b,n+1)
15     I1=(h/2)*((2*sum(f(t)))-f(t(1))-f(t(n+1)))
16 endfunction
17 deff(' [y]=f(t) ', 'y=k*(2-t) ')
18 W=Trap_Composite1(f,a,b,n) //ft-lb
19 //Solving the simultaneous equations
20 v3=sqrt(W/(0.5*(W2/g)+0.5*(W1/g)*(-W2/W1)^2)) //ft/s
21 v2=-(W2/W1)*v3 //ft/s
22 //Result
23 clc
24 printf('The speed of 2lb block is %f ft/s and that
        of 3lb block is %f ft/s',v2,v3)

```

Scilab code Exa 18.34 Imp

```

1 //Initilization of variables
2 //Here the integration is indefinite hence it will
  be computed manually and entered
3 W=10 //lb
4 l=4 //ft
5 w=2 //rad/s
6 g=32.2 //ft/s^2
7 //Calculations
8 //Part (a)
9 wf=1.5 //rad/s
10 t=sqrt(((W/g)*(1*w*1))-((W/g)*(1*wf*1))) //s
11 //Part (b)

```

```

12 //Applying conservation of angular momentum
13 r=((W/g)*l*wf*1)/((W/g)*l*w) //ft
14 //Result
15 clc
16 printf('The answer for part (a) is %f s\n and the
        answer for part (b) is %f ft ',t,r)

```

Scilab code Exa 18.35 Imp

```

1 //Initilization of variables
2 W=2.5 //lb
3 w=36 //rad/s
4 Idisk=0.4 //slug-ft^2
5 g=32.2 //ft/s^2
6 //Calculations
7 Ii=Idisk+(2*(W/g)*(3/12)^2) //slug-ft^2
8 If=Idisk+(2*(W/g)*(11/12)^2) //slug-ft^2
9 //Since no external moments act
10 //Applying conservation of momentum
11 wf=(Ii*w)/If //rad/s
12 //Result
13 clc
14 printf('The final angular speed is %f rad/s ',wf)

```

Scilab code Exa 18.39 Imp

```

1 //Initilization of variables
2 u1=6 //ft/s
3 u2=-8 //ft/s
4 e=0.8 //coefficient of restitution
5 //Calculations
6 //Solving both simultaneous equations
7 A=[1, -1;1, 1]

```

```

8 B=[11.2;-2]
9 C=inv(A)*B //ft/s
10 //Result
11 clc
12 printf('The velocities are v1=%f ft/s and v2=%f ft/s
        ',C(2),C(1))

```

Scilab code Exa 18.40 Imp

```

1 //Initialization of variables
2 h1=20 //m
3 h2=14 //m
4 g=9.8 //m/s^2
5 //Calculations
6 u1=sqrt(2*g*h1) //m/s
7 u2=0 //m/s
8 v1=-sqrt(2*g*h2) //m/s
9 v2=0 //m/s
10 e=(v2-v1)/(u1-u2) //coefficient of restitution
11 //Result
12 clc
13 printf('The value of coefficient of restitution is
        %f',e)

```

Scilab code Exa 18.41 Imp

```

1 //Initialization of variables
2 u=6.55 //ft/s
3 g=32.2 //ft/s^2
4 L=6 //ft
5 W=5 //lb
6 c=0.7 //fraction of impulse acting in second phase
7 //Calculations

```

```

8 //Impulse
9 I=(W/g)*(u/3) //N.s
10 //Second Phase
11 v=(u*2-(c*(W*u*(1/3)))) //ft/s
12 wprime=(1.09*60+c*(W*u*(1/3)*6))/60 //rad/s
13 //Result
14 clc
15 printf('The value of v=%f ft/s and that of w is %f
        rad/s',v,wprime)
16 //The value of w is incorrect in the textbook

```

Scilab code Exa 18.42 Imp

```

1 //Initilization of variables
2 m1=9 //kg
3 m2=5.5 //kg
4 u1=-3 //m/s
5 u2=1.77 //m/s
6 e=0.8 //coefficient of restitution
7 //Calculations
8 //Solving by matrix method after we get the two
    equations
9 A=[-1,1;m1,m2]
10 B=[(e*u1-e*u2);m1*u1+m2*u2]
11 C=inv(A)*B //m/s
12 //Result
13 clc
14 printf('The 9kg ball will rebound up the speed of %f
        m/s\n and the 5.5kg ball will move to the right
        and down \nwith components of %f m/s and %f m/s
        respectively ',C(1),u2,-C(2))

```

Scilab code Exa 18.46 Imp

```

1 //Initilization of variables
2 v=4 //m/s
3 m=9 //kg
4 s=1.5 //m
5 //Calculations
6 Io=(2/3)*(m*s^2) //kg.m
7 w=(m*v*s*0.5)/Io //rad/s
8 //Result
9 clc
10 printf('The angular velocity of the box is %i rad/s'
        ,w)

```

Scilab code Exa 18.48 Imp

```

1 //Initilization of variables
2 m=2000 //kg
3 mf=8500 //kg
4 vr=2000 //m/s
5 a=9.8 //m/s^2
6 //Calculations
7 //Applying Newtons Second Law
8 dm_dt=-(-(m+mf)*a)-(m+mf)*a/(-vr) //kg/s
9 //Result
10 clc
11 printf('dm/dt=%f kg/s',dm_dt)
12 //The negative sign indicates the loss in the mass
    of the system

```

Scilab code Exa 18.50 Imp

```

1 //Initilization of variables
2 W=4000 //lb
3 k=3 //ft

```

```

4 wp=(1/60)*2*%pi //rad/s
5 ws=(300/60)*2*%pi //rad/s
6 d=3.5 //ft
7 g=32.2 //ft/s^2
8 //Calculations
9 I=(W/g)*k^2 //slug-ft^2
10 M=I*ws*wp //lb-ft
11 //Now equating M to Rf-Rr gives one equations and
    vertical sum yields other
12 //solving them by matrix method
13 A=[1,-1;1,1]
14 B=[M*(2/d);W]
15 C=inv(A)*B //lb
16 //Result
17 clc
18 printf('The weight of Rf and Rr are %f lb and %f lb
    respectively ',C(1),C(2))

```

Scilab code Exa 18.51 Imp

```

1 //As the integration is indefinite we will directly
    consider the equation with R
2 //Initillization of variables
3 GM=1.41*10^16 //ft^3/s^2
4 r=2640000 //ft
5 theta=60 //degrees
6 R=21120000 //ft
7 //Calculations
8 v1=sqrt((GM*((1/R)-(1/(R+r))))/2.031) //ft/s
9 //Result
10 clc
11 printf('The speed required will be %f ft/s',v1)

```

Scilab code Exa 18.52 Imp

```
1 //Initialization of variables
2 k=4 //lb/ft
3 so=1 //ft
4 W=1/2 //lb
5 g=32.2 //ft/s^2
6 vo=5 //ft/s
7 //Calculations
8 m=W/g //kg
9 //Angular momentum is conserved
10 v=sqrt((0.5*k*so^2*2*2*g)+vo^2) //ft/s
11 //Using vd=15
12 d=15/v //ft
13 //Result
14 clc
15 printf('The ball passes %f ft close to the fixed pin
        ',d)
```

Chapter 19

Mechanical Vibration

Scilab code Exa 19.2 Vibrations

```
1 //Initialization of variables
2 k=18 //lb/in
3 g=386 //in/s^2
4 W=35 //lb
5 //Calculations
6 f=(1/(2*pi))*sqrt((k*g/W)) //cps
7 period=1/f //s
8 //Result
9 clc
10 printf('The period of vibration is %f s',period)
```

Scilab code Exa 19.11 Vibrations

```
1 //Initialization of variables
2 ds=0.2 //m
3 ts=0.05 //m
4 rhos=7850 //kg/m^3 density of steel
5 dw=0.002 //m
```



```

6 lw=0.9 //m
7 G=80*10^9 //Pa
8 // Calculations
9 //Torsional Constant
10 K=(%pi*dw^4*G)/(32*lw) //m/rad
11 //Mass Calculations
12 m=(1/4)*%pi*(ds^2)*ts*rhos //kg
13 //Moment of Inertia
14 Io=(1/2)*m*(ds/2)^2 //kg.m^2
15 //Frequency
16 f=(1/(2*%pi))*(sqrt(K/Io)) //Hz
17 //Result
18 clc
19 printf('The natural frequency of the system is %f Hz
        ',f)

```

Scilab code Exa 19.13 Vibrations

```

1 //Initilization of variables
2 m=120 //kg
3 k=0.3 //m
4 ls=0.6 //m
5 ds=0.05 //m
6 G=80*10^9 //Pa
7 // Calculations
8 //Polar Moment of Inertia
9 J1=m*k^2 //kg.m^2
10 J2=J1 //kg.m^2
11 J=(1/32)*%pi*(ds^4) //m^4
12 //Frequency
13 f=(1/(2*%pi))*(sqrt((J*G*(J1+J2))/(ls*J1*J2))) //Hz
14 //Result
15 clc
16 printf('The natural frequency of the torsional
        oscillation is %f Hz',f)

```

Scilab code Exa 19.14 Vibrations

```
1 //Initialization of variables
2 ds=2 //in
3 L=15 //in
4 Wf1=300 //lb
5 k1=6 //in
6 Wf2=100 //lb
7 k2=4 //in
8 G=12*10^6 //Pa
9 g=386 //in/s^2
10 //Calculations
11 //Moment of inertia of flywheel
12 Jf=(Wf1/g)*k1^2 //lb-s^2-in
13 //Moment of inertia of the rotor
14 Jr=(Wf2/g)*k2^2 //lb-s^2-in
15 //Moment of inertia of the shaft cross section
16 J=(1/32)*%pi*ds^4 //in^4
17 //Frequency
18 f=(1/(%pi*2))*(sqrt((J*G*(Jf+Jr))/(L*Jf*Jr))) //cps
19 //result
20 clc
21 printf('The natural frequency of the system is %f
        cps ',f)
```

Scilab code Exa 19.15 Vibrations

```
1 //Initialization of variables
2 W=10 //lb
3 A=2 //in^2
4 //Calculations
```

```

5 wn=sqrt(((A/144)*5*62.4*5)/2.59) //Hz
6 //Result
7 clc
8 printf('The frequency of oscillation is %f Hz',wn)

```

Scilab code Exa 19.16 Vibrations

```

1 //Initialization of variables
2 f=50 //cps
3 g=386 //in/s^2
4 E=30*10^6 //lb/in^2
5 l=4 //in
6 I=2.08*10^-6 //in^4
7 //Calculations
8 W=(3*E*I*g)/(((f*2*pi)^2)*l^3) //lb
9 //Result
10 clc
11 printf('The value of W is %f lb',W)

```

Scilab code Exa 19.19 Vibrations

```

1 //Initialization of variables
2 F=10 //lb
3 v=20 //in/s
4 g=386 //in/s
5 W=12 //lb
6 k=20 //lb/in
7 //Calculations
8 //Coefficient of damping
9 c=F/v //lb-s/in
10 //Natural Frequency
11 wn=sqrt((k*g)/W) //rad/s
12 //Critical Damping coefficient

```

```

13 cr=(2*W/g)*wn //lb-s/in
14 //Damping Coefficient
15 d=c/cr
16 //Frequency of damped vibrations
17 wd=sqrt(1-d^2)*wn //rad/s
18 //Result
19 clc
20 printf('The frequency of damped vibrations is %f rad
/s ',wd)

```

Scilab code Exa 19.20 Vibrations

```

1 //Initilization of variables
2 wn=25.4 //rad/s
3 t=0.261 //s
4 d=0.316
5 //Calculations
6 del=d*t*wn //logarithmic decay
7 //Result
8 clc
9 printf('The rate of decay is %f',del)

```

Scilab code Exa 19.24 Vibrations

```

1 //Initilization of variables
2 F=9 //N
3 m=5 //kg
4 k=6000 //N/m
5 f1=1 //Hz
6 f2=5.4 //Hz
7 f3=50 //Hz
8 //Calculations
9 //Natural Frequency

```

```

10 fn=(1/(%pi*2))*(sqrt(k/m)) //Hz
11 deltax=F/(k/1000) //mm
12 //Part (a)
13 r1=f1/fn
14 amp1=deltax/(1-r1^2) //mm amplitude
15 //Part (b)
16 r2=f2/fn
17 amp2=deltax/(1-r2^2) //mm amplitude
18 //Part (c)
19 r3=f3/fn
20 amp3=deltax/(1-r3^2) //mm amplitude
21 //Result
22 clc
23 printf('The amplitudes in part (a),(b) and (c)
        respectively are \n %f mm,%f mm and %f mm
        respectively ',amp1 ,amp2 ,amp3)

```

Scilab code Exa 19.25 Vibrations

```

1 //Initilization of vvariables
2 g=386 //in/s^2
3 W=20 //lb
4 w=600 //rpm
5 ratio=1/12
6 //Calculations
7 r=sqrt((1/ratio)+1)
8 fn=((w/60)/r) //cps
9 k=((fn*2*%pi)^2*W)/(g) //lb/in
10 //Result
11 clc
12 printf('The value of k is %f lb/in ',k)

```

Scilab code Exa 19.28 Vibrations

```
1 //Initialization of variables
2 X=12 //mm
3 me_M=1.3 //mm
4 //Calculations
5 d=(me_M)/(2*X)
6 //Result
7 clc
8 printf('The damping ratio is %f',d)
```
