

Scilab Textbook Companion for
Mechanics of Materials
by R. C. Hibbeler¹

Created by
Deepika S Namboothiri
Bachelor of Engineering
Mechanical Engineering
Cummins College of Engineering
College Teacher
Dr. Gautam S. Chandekar
Cross-Checked by
Chaitanya Potti

July 31, 2019

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

Book Description

Title: Mechanics of Materials

Author: R. C. Hibbeler

Publisher: Pearson

Edition: 6

Year: 2010

ISBN: 978-81-317-1502-4

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes	4
1 Stress	5
2 Strain	30
3 Mechanical Properties of Materials	35
4 Axial Load	44
5 Torsion	67
6 Bending	89
7 Transverse Shear	114
8 Combined Loadings	124
9 Stress Transformation	133
10 Strain Transformation	146
11 Design of Beams and Shafts	164
12 Deflection of Beams and Shafts	171

13 Buckling of Columns	183
14 Energy Methods	200

List of Scilab Codes

Exa 1.1	S1	5
Exa 1.2	S2	6
Exa 1.3	S3	8
Exa 1.4	S4	9
Exa 1.5	S5	10
Exa 1.6	S6	13
Exa 1.7	S7	14
Exa 1.8	S8	15
Exa 1.9	S9	16
Exa 1.10	S10	17
Exa 1.11	S11	19
Exa 1.12	S12	20
Exa 1.13	S13	22
Exa 1.14	S14	23
Exa 1.15	S15	25
Exa 1.16	S16	26
Exa 1.17	S17	27
Exa 2.1	Strain1	30
Exa 2.2	Strain2	31
Exa 2.3	Strain3	32
Exa 2.4	Strain4	33
Exa 3.1	MPM1	35
Exa 3.2	MPM2	36
Exa 3.3	MPM3	37
Exa 3.4	MPM4	39
Exa 3.5	MPM5	40
Exa 3.6	MPM6	42
Exa 4.1	AL1	44

Exa 4.2	AL2	45
Exa 4.3	AL3	47
Exa 4.5	AL5	48
Exa 4.6	AL6	50
Exa 4.7	AL7	51
Exa 4.8	AL8	53
Exa 4.9	AL9	54
Exa 4.10	AL10	55
Exa 4.11	AL11	56
Exa 4.12	AL12	58
Exa 4.13	AL13	60
Exa 4.14	AL14	61
Exa 4.15	AL15	62
Exa 4.16	AL16	63
Exa 4.17	AL17	64
Exa 5.1	T1	67
Exa 5.3	T3	68
Exa 5.4	T4	69
Exa 5.5	T5	70
Exa 5.6	T6	71
Exa 5.7	T7	72
Exa 5.8	T8	73
Exa 5.9	T9	74
Exa 5.11	T11	76
Exa 5.12	T12	77
Exa 5.13	T13	79
Exa 5.15	T15	80
Exa 5.16	T16	81
Exa 5.17	T17	82
Exa 5.18	T18	84
Exa 5.19	T19	85
Exa 5.20	T20	86
Exa 5.21	T21	87
Exa 6.5	B5	89
Exa 6.11	B11	90
Exa 6.13	B13	91
Exa 6.14	B14	91
Exa 6.15	B15	93

Exa 6.16	B16	94
Exa 6.17	B17	95
Exa 6.18	B18	96
Exa 6.19	B19	98
Exa 6.20	B20	100
Exa 6.21	B21	101
Exa 6.22	B22	102
Exa 6.23	B23	104
Exa 6.24	B24	105
Exa 6.25	B25	107
Exa 6.26	B26	108
Exa 6.27	B27	109
Exa 6.28	B28	110
Exa 6.29	B29	111
Exa 6.30	B30	112
Exa 7.1	TS1	114
Exa 7.2	TS2	115
Exa 7.3	TS3	117
Exa 7.4	TS4	118
Exa 7.5	TS5	120
Exa 7.6	TS6	121
Exa 7.7	TS7	122
Exa 8.1	CL1	124
Exa 8.2	CL2	125
Exa 8.3	CL3	126
Exa 8.4	CL4	127
Exa 8.5	CL5	129
Exa 8.6	CL6	131
Exa 9.1	ST1	133
Exa 9.2	ST2	134
Exa 9.5	ST5	135
Exa 9.6	ST6	136
Exa 9.9	ST9	138
Exa 9.10	ST10	139
Exa 9.11	ST11	140
Exa 9.12	ST12	141
Exa 9.13	ST13	142
Exa 9.14	ST14	143

Exa 9.15	ST15	145
Exa 10.1	StnT1	146
Exa 10.2	StnT2	147
Exa 10.3	StnT3	148
Exa 10.4	StnT4	150
Exa 10.5	StnT5	151
Exa 10.6	StnT6	152
Exa 10.7	StnT7	153
Exa 10.8	StnT8	155
Exa 10.9	StnT9	156
Exa 10.10	StnT10	157
Exa 10.11	StnT11	159
Exa 10.12	StnT12	160
Exa 10.13	StnT13	161
Exa 10.14	StnT14	162
Exa 11.1	DBS1	164
Exa 11.2	DBS2	165
Exa 11.3	DBS3	168
Exa 11.6	DBS6	169
Exa 12.6	DefBS6	171
Exa 12.10	DefBS10	171
Exa 12.12	DefBS12	173
Exa 12.13	DefBS13	174
Exa 12.14	DefBS14	175
Exa 12.15	DefBS15	176
Exa 12.16	DefBS16	177
Exa 12.21	DefBS21	178
Exa 12.22	DefBS22	179
Exa 12.23	DefBS23	180
Exa 12.24	DefBS24	181
Exa 13.1	BoC1	183
Exa 13.2	BoC2	184
Exa 13.3	BoC3	185
Exa 13.4	BoC4	186
Exa 13.5	BoC5	187
Exa 13.6	BoC6	188
Exa 13.7	BoC7	190
Exa 13.8	BoC8	191

Exa 13.9	BoC9	192
Exa 13.10	BoC10	193
Exa 13.11	BoC11	195
Exa 13.12	BoC12	196
Exa 13.13	BoC13	197
Exa 13.14	BoC14	198
Exa 14.1	EM1	200
Exa 14.5	EM5	201
Exa 14.6	EM6	202
Exa 14.8	EM8	203
Exa 14.9	EM9	204
Exa 14.10	EM10	205
Exa 14.11	EM11	206
Exa 14.12	EM12	208
Exa 14.13	EM13	209
Exa 14.16	EM16	210
Exa 14.17	EM17	211
Exa 14.18	EM18	213
Exa 14.21	EM21	214

Chapter 1

Stress

Scilab code Exa 1.1 S1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 1.1 :")
4
5 w_varying = 270;
6 l_crossection = 9;
7 l_cb = 6;
8 l_ac = 2;
9 w_c = (w_varying/l_crossection) * l_cb //By
    proportion, load at C is found.
10 f_resultant_c = 0.5* w_c *l_cb
11 // Equations of Equilibrium
12
13 //Balancing forces in the x direction:
14 n_c = 0
15
16 //Balncing forces in the y direction:
17 v_c = f_resultant_c
18
19 // Balncing the moments about C:
20 m_c = - (f_resultant_c*l_ac)
```

```

21
22
23 // Displaying results:
24
25 printf('\n\nThe resultant force at C = %.2f N',
        f_resultant_c);
26 printf('\nThe horizontal force at C = %.2f N',n_c);
27 printf('\nThe vertical force at C = %.2f N',v_c);
28 printf('\nThe moment about C = %.2f Nm',m_c)
    ;
29
30
31 //

```

END

Scilab code Exa 1.2 S2

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 1.2 : ")
4
5 f_d = 225; //N
6 w_uniform = 800; // N/m
7 l_ac = 0.200; //m
8 l_cb = 0.05+0.1; //m
9 l_bd = 0.100; //m
10 l_bearing = 0.05; //m
11 f_resultant = w_uniform*l_cb //120N
12 l_f_resultant_b = (l_cb/2)+ l_bearing; //0.125m
13 l = l_ac + l_cb + l_bearing + l_bd
14
15
16 // This problem is solved by considering segment AC

```

```

    of the shaft.
17
18 //Support Reactions:
19
20 m_b = 0; // Net moment about
    B is zero for equilibrium . Sum Mb = 0.
21 a_y = -((f_d*l_bd) - (f_resultant*l_f_resultant_b))/
    (1 - l_bd) // finding the reaction force at A
22
23 // Refer to the free body diagram in Fig.1-5c.
24 f_c = 40 //N
25 //Balancing forces in the x direction:
26 n_c = 0
27
28 //Balancing forces in the y direction:
29 v_c = a_y - f_c // -18.75N - 40N - Vc = 0
30
31 // Balancing the moments about C:
32 m_c = ((a_y * (l_ac + 0.05)) - f_c*(0.025) ) // Mc
    +40N(0.025m)+ 18.75N(0.250m) = 0
33
34
35 // Displaying results:
36
37 printf('\n\nThe resultant force = %.2f N',
    f_resultant);
38 printf('\n\nThe reaction force at A = %.2f N',a_y);
39 printf('\n\nThe horizontal force at C = %.2f N',n_c);
40 printf('\n\nThe vertical force at C = %.2f N',v_c);
41 printf('\n\nThe moment about C = %.2f Nm',m_c);
42
43 //

```

END

Scilab code Exa 1.3 S3

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 1.3 :")
4
5 // Given:
6 l_ac = 1; //m.
7 l_cd = 1.5 ; //m.
8 l_bd = 0.5; //m.
9 r_a = 0.125; //m.
10 r_d = 0.125; //m.
11 W = 2000; // N
12
13
14 // Equations of equilibrium:
15
16 //Balancing forces in the x direction:
17 n_c = -W; // N
18
19 //Balncing forces in the y direction:
20 v_c = -W; //N
21
22 // Balncing the moments about C:
23 m_c = - (W*(r_a +l_ac)- W*r_a)
24
25
26 // Displaying results:
27
28 printf('\n\nThe horizontal force at C = %.2f N',
        n_c);
29 printf('\n\nThe vertical force at C = %.2f N',v_c)
        ;
30 printf('\n\nThe moment about C = %.2f Nm',m_c
```

```
    );  
31  
32 //
```

END

Scilab code Exa 1.4 S4

```
1 clear all; clc;  
2  
3 disp(" Scilab Code Ex 1.4 :")  
4  
5 // Given:  
6 l_ag = 1; //Length of AG is 1m.  
7 l_gd = 1; //Length of GD is 1m.  
8 l_de = 3; //Length of DE is 1m.  
9 f_a = 1500; //Force at A is 1500N.  
10 l_ec = 1.5; //Length of EC is 1m.  
11 l = l_ag + l_gd + l_de;  
12 w_uniform_varying = 600; //Nm.  
13  
14 w_resultant = 0.5*l_de*w_uniform_varying;  
15 // calling point of action of resultant as P  
16 l_ep = (2/3)*l_de; //Distance between points P and E  
17  
18  
19  
20 f_ba = 7750; //N  
21 f_bc = 6200; //N  
22 f_bd = 4650; //N  
23  
24 //Free Body Diagram: Using the result for Fba, the
```

```

    left section AG of the beam is shown in Fig 1-7d.
25
26 // Equations of equilibrium:
27
28 //Balancing forces in the x direction:
29 n_g = -f_ba * (4/5); // N
30
31 //Balancing forces in the y direction:
32 v_g = -f_a + f_ba*(3/5); //N
33
34 // Balancing the moments about C:
35 m_g = (f_ba * (3/5)*l_ag) - (f_a * l_ag); //Nm
36
37
38
39 // Displaying results:
40
41
42 printf('\n\nThe horizontal force at G = %.2f N',n_g)
    ;
43 printf('\nThe vertical force at G    = %.2f N',v_g);
44 printf('\nThe moment about G        = %.2f Nm',m_g);
45
46
47 //

```

END

Scilab code Exa 1.5 S5

```

1 clear all; clc;
2
3

```



```

4 disp(" Scilab Code Ex 1.5 :")
5
6 // Given:
7 f_a = 50; //N
8 m_a = 70; // Moment at A in Nm
9 l_ad = 1.25; //Length of AD in m.
10 l_bd = 0.5; //Length of BD in m.
11 l_cb = 0.75; //Length of BC in m.
12 w_l = 2; //Kg/m
13 g = 9.81; //N/kg- acceleration due to gravity
14
15
16
17 //Free Body Diagram :
18
19 w_bd = w_l*l_bd*g; //in N. Weight of each segment of
    pipe that acts through the centre of gravity of
    each segment.
20 w_ad = w_l*l_ad*g;
21
22 // Equations of Equilibrium
23
24 //Balancing forces in the x direction:
25 f_b_x = 0; // N
26
27 //Balancing forces in the y direction:
28 f_b_y = 0; //N
29
30 //Balancing forces in the z direction:
31 f_b_z = g + w_ad + f_a; //N
32
33 // Balancing Moments in the x direction:
34 m_b_x = - m_a + (f_a*l_bd) + (w_ad*l_bd) + (l_bd/2)*
    g; //Nm
35
36 // Balancing Moments in the y direction:
37 m_b_y = - (w_ad*(l_ad/2)) - (f_a*l_ad); //Nm
38

```

```

39 // Balancing Moments in the z direction:
40 m_b_z = 0; //Nm
41
42 v_b_shear = sqrt(f_b_z ^2 + 0); //Shear Force in N
43 t_b = - m_b_y; //Torsional Moment in Nm
44 m_b = sqrt(m_b_x ^2+ 0); // Bending moment in Nm
45
46
47 //Display
48
49 // Displaying results:
50
51
52 printf('\n\n The weight of segment BD
           = %.1f N',w_bd);
53 printf('\n The weight of segment AD           =
           %.1f N',w_ad);
54 printf('\n The force at B in the Z direction   =
           %.1f N',f_b_z);
55 printf('\n The moment about B in the X direction =
           %.1f Nm',m_b_x);
56 printf('\n The moment about G in the Y direction =
           %.1f Nm',m_b_y);
57 printf('\n The Shear Force at B           =
           %.1f N',v_b_shear);
58 printf('\n The Torsional Moment at B       =
           %.1f Nm',t_b);
59 printf('\n The Bending Moment at B         =
           %.1f Nm',m_b);
60
61
62
63 //

```

END

Scilab code Exa 1.6 S6

```
1 clear all; clc;
2
3
4 disp(" Scilab Code Ex 1.6 :")
5
6 //Given:
7 netf_b = 18*(10 ^3); //N Net force at B.
8 netf_c = 8*(10^3); //N Net force at C.
9 f_a = 12 *(10^3); //N Force at A.
10 f_d = 22* (10^3); //N Force at D.
11 w = 35; //mm Width.
12 t = 10; //mm Thickness.
13
14 //calculations:
15 p_bc = netf_b + f_a; //N Net force in region BC.
16 a = w*t; //m^2 The area of the cross section.
17 avg_normal_stress = p_bc/a; //Average Normal Stress.
18
19
20
21 // Displaying results:
22
23 printf('\n\n Net force in the region BC
           = %.2f N',p_bc);
24 printf('\nThe Area of cross section
           = %.2f m^2 ',a);
25 printf('\nThe Average Normal Stress in the bar when
           subjected to load = %.2f MPa',avg_normal_stress);
26
27 //
```

END

Scilab code Exa 1.7 S7

```
1 clear all; clc;
2
3
4 disp("Scilab Code Ex 1.7 :")
5
6 //Given :
7 m_lamp = 80; //Mass of lamp in Kg.
8 d_ab = 10; // Diameter of AB in mm.
9 d_bc = 8; // Diameter of BC in mm.
10 ab_h = 60 *(%pi/180); // In degrees - Angle made by
    AB with the horizontal.
11 w = m_lamp*9.81; //N
12 a_bc = (%pi/4)*(d_bc^2); //m^2 Area of cross section
    of rod BC
13 a_ab = (%pi/4)*(d_ab^2); //m^2 Area of cross section
    of rod AB
14
15
16
17 // Equations of equilibrium: Solving equilibrium
    equations simultaneously ,using matrices ,in the
    x and y directions to obtain force in BC and
    force in BA.
18
19
20 a = [(4/5) -(cos(ab_h)) ; (3/5) (sin(ab_h))];
21 b = [0 ; w];
22 f = zeros(1)
23
24 f = a\b;
```

```

25 f_bc = f(1); // Force in BC in N.
26 f_ba = f(2); //Force in BA in N.
27 avg_normal_stress_a = f_ba / a_ab; //Mpa Average
    Normal Stress in AB
28 avg_normal_stress_c = f_bc/ a_bc;// Mpa Average
    Normal Stress in BC
29
30
31 // Displaying results:
32
33
34 printf('\n\nThe Weight of lamp = %.2f N',w);
35 printf('\n\nThe Net force in BC = %.2f N',f_bc);
36 printf('\n\nTheNet force in BA = %.2f N',f_ba);
37 printf('\n\nThe Average Normal Stress in AB when
    subjected to load = %.2f MPa',avg_normal_stress_a
    );
38 printf('\n\nThe Average Normal Stress in BC when
    subjected to load = %.2f MPa',avg_normal_stress_c
    );
39
40 //

```

END

Scilab code Exa 1.8 S8

```

1 clear all; clc;
2
3
4 disp(" Scilab Code Ex 1.8 :")
5
6 //Given:

```

```

7 h_above_ab = 0.8;
8 h_below_ab = 0.2;
9 d_a = 0.2;
10 d_b = 0.1;
11 sp_w = 80;
12
13 // Equation of Equilibrium:
14
15
16 a = %pi* (d_a^2); // Area of cross section in m^2
17 p = sp_w * h_above_ab * a;
18 avg_comp_stress = p/a; // The average compressive
    stress in kN/m^2
19
20 //Display:
21
22 printf('\nThe internal Axial force P      = %.2 f kN',
    p);
23 printf('\nThe average compressive stress = %.2 f kN/m
    ^2', avg_comp_stress);
24
25
26 //

```

END

Scilab code Exa 1.9 S9

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 1.9 : ")
4
5 //Given :

```

```

6 f = 3000; //N Force acting at distance x from AB.
7 l_ac = 200; //Length of AC in mm.
8 a_ab = 400; //Cross sectional area of AB in mm^2.
9 a_c = 650; // area of C in mm^2.
10
11
12 f_ans = zeros(3)
13
14 k = [1 1 0;0 l_ac -f; 1.625 -1 0]
15 l = [f ; 0 ; 0 ]
16 f_ans = k\l;
17
18 f_ab = f_ans(1)
19 f_c = f_ans(2)
20 x = f_ans(3)
21
22 //Display:
23
24 printf('\n\nThe Net force on AB          = %.2 f N',ceil
      (f_ab));
25 printf('\nNet force on C                = %.2 f N',f_c);
26 printf('\nDistance of force from AB = %.2 f mm',ceil(
      x));
27
28
29 //

```

END

Scilab code Exa 1.10 S10

```

1 clear all; clc;
2

```

```

3 disp(" Scilab Code Ex 1.10 : ")
4
5 //Given:
6 af = 800; //N Axial force along centroidal axis
7 t = 0.040; //m thickness of square cross section
8 ang_b = 30 *(%pi/180) ;
9 ang_b_comp = 60 *(%pi/180);
10 a = t^2; //m^2 Area of cross section
11 a_new = ((t*1000)^2)/(sin(ang_b_comp)); // mm^2 Area
    of section at b-b
12
13 //Part(a)
14
15 //Internal Loading: The bar is sectioned , Fig 1-24b,
    and the internal resultant loading consists of
    only axial force .
16
17 // Average Stress:
18 avg_stress = af/(a* 1000);
19
20 //Shear Force at the section is zero.
21 //The average normal stress distribution over the
    cross section is shown in Fig 1-24c.
22
23
24 //Part(b)
25
26
27 //solve the two equations for two unknowns:
28
29 N = af * cos(ang_b);
30 V = af * sin(ang_b);
31 avg_normal_stress = (N*1000)/ a_new; // kPa
32 avg_shear_stress = (V*1000)/a_new; //kPa
33
34 //Display
35
36 printf('\n\nThe average stress for section a-a

```



```

        = %.2f kPa', avg_stress);
37 printf('\nThe Normal Force for section b-b
        = %.2f N', N);
38 printf('\nThe Shear Force for section b-b
        = %.2f N', V);
39 printf('\nThe Average Normal Stress for section b-b
        = %.2f kPa', avg_normal_stress);
40 printf('\nThe Average Shear Stress for section b-b
        = %.2f kPa', ceil(avg_shear_stress));
41
42 //

```

END

Scilab code Exa 1.11 S11

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 1.11 : ")
4
5 //Given :
6 f = 5000; //N
7 d_rod = 10; //Diameter of steel rod in mm.
8 l_bc = 20; //Length of side bc in mm.
9 l_bd = 40; //Length of side bd in mm.
10 a_rod = (%pi/4)* (d_rod^2); //Area of cross section
    of the rod in mm^2.
11 a_strut = l_bc*l_bd ; //Area of strut in mm^2.
12
13
14 //Average shear stress
15
16 avg_shear_rod = f/a_rod; //for rod in Mpa

```

```

17 avg_shear_strut = (f/2)/a_strut; //for strut
18
19 //Display:
20
21 printf('\n\nThe average shear stress for the rod
    = %.2f MPa',avg_shear_rod);
22 printf('\n\nThe average shear stress for the strut =
    %.2f MPa',avg_shear_strut);
23
24
25
26 //

```

END

Scilab code Exa 1.12 S12

```

1 clear all; clc;
2
3
4 disp("Scilab Code Ex 1.12 : ")
5
6 //Given:
7 l_bc = 50; //Length of BC in mm.
8 l_db = 75; // mm.
9 l_ed = 40; // mm.
10 l_ab = 25; // mm.
11 f_diagonal = 3000; //N
12 a1 = l_ab*l_ed; //Area of face AB in mm^2.
13 a2 = l_bc*l_ed ; //mm^2.
14 a3 = l_db*l_ed ; // mm^2.
15
16 //Internal loadings – The free body diagram of the

```

```

    inclined member is shown in 1-26b.
17
18 //Equilibrium Equations
19
20 //Balancing forces along the x- direction.
21 f_ab = f_diagonal*(3/5); //Force on segment AB in N
22 V = f_ab; //Shear force acting on the sectioned
    horizontal plane EDB in N
23
24 //Balancing forces along the Y direction.
25 f_bc = f_diagonal*(4/5); //Force on segment BC in N.
26
27 //Average compressive stresses along the horizontal
    and vertical planes:
28
29 avg_comp_ab = f_ab/a1; // N/mm^2
30 avg_comp_bc = f_bc/a2; // N/mm^2
31
32 //Average shear stress acting on the horizontal
    plane defined by EDB :
33
34 avg_shear = f_ab/a3; // N/mm^2
35
36 //Display:
37
38
39 printf('\n\nThe Force on segment AB
    = %.2f N',f_ab);
40 printf('\nThe Shear Force on sectioned plane EDB =
    %.2f N',V);
41 printf('\nThe Force on segment BC =
    %.2f N',f_bc);
42 printf('\nThe average compressive stress along AB =
    %.2f N/mm^2',avg_comp_ab);
43 printf('\nThe average compressive stress along BC =
    %.2f N/mm^2',avg_comp_bc);
44 printf('\nThe average shear stress along EDB =
    %.2f N/mm^2',avg_shear);

```

```
45
46 //


---


END


---


```

Scilab code Exa 1.13 S13

```
1
2 clear all; clc;
3
4
5 disp("Scilab Code Ex 1.13 : ")
6
7 //Given:
8 shear_allow = 90; //MPa
9 tensile_allow = 115; //MPa
10
11 l_AP = 2; //m
12 l_PB = 1; //m
13 resultant_A = 5.68; //kN
14 resultant_B = 6.67; //kN
15 v_a = 2.84; //kN
16 v_b = 6.67; //kN
17
18
19 //Diameter of the Pins:
20 A_A = (v_a*10^3)/(shear_allow*10^6); //Area of pin A
21 da = (sqrt((4*A_A)/%pi))*10^3 // d = (square root of
    (area*4/pi)) in mm
22 A_B = (v_b*10^3)/(shear_allow*10^6) ; //Area of pin
    B
23 db = (sqrt((4*A_B)/%pi))*10^3 // Area = (%pi\4)d^2
    in mm^2
```

```

24
25 chosen_da = ceil(da);
26 chosen_db = ceil(db);
27
28 //Diameter of Rod:
29 A_bc = (resultant_B*10^3)/(tensile_allow*10^6); //
    Area of BC
30 dbc = (sqrt((4*A_bc)/%pi)*10^3); // Area = %pi\4)d
    ^2
31 chosen_dbc = ceil(dbc);
32
33 //Displaying Results:
34
35 printf ("\n\n The diameter of pin A = %.3 f mm", da)
    ;
36 printf ("\n The diameter of pin B = %.3 f mm", db);
37 printf ("\n The diameter of rod BC = %.2 f mm", dbc);
38 printf ("\n\n\nThe chosen diameters are: ");
39 printf ("\n The diameter of pin A = %.3 f mm",
    chosen_da);
40 printf ("\n The diameter of pin B = %.3 f mm",
    chosen_db);
41 printf ("\n The diameter of rod BC = %.2 f mm",
    chosen_dbc);
42
43 //

```

END

Scilab code Exa 1.14 S14

```

1 clear all; clc;
2

```

```

3
4 disp(" Scilab Code Ex 1.14 : ")
5
6 //Given:
7 shear_allow = 55; //MPa
8 l_ac = 200; //mm
9 l_cd= 75; //mm
10 l_de = 50; //mm
11 l_ce = l_cd + l_de;
12 load_d =15; //kN
13 load_e = 25; //kN
14
15 //Internal Shear Force:
16 //summation Mc = 0
17
18 f_ab = ((load_d*l_cd +load_e*(3/5)*l_ce)/l_ac);
19 c_x =-load_d + (load_e*(4/5)); //resolving C in x
    dir
20 c_y = load_d + (load_e*(3/5)); //resolving C in y
    dir
21
22 f_c = sqrt(c_x^2 + c_y^2); //kN
23 V = f_c/2;
24
25 //Required Area
26 A = ((V*10^3)/(shear_allow)); //A = V/Allowable
    shear in mm^2
27 d = ((sqrt((4*A)/%pi))) // Area = (%pi\4)d^2 in mm^2
28
29 chosen_d = ceil(ceil(d))+1;
30
31 //Displaying Results:
32
33
34 printf("\n\nThe force at AB                = %.2 f kN",
    f_ab);
35 printf("\nThe resultant force at C = %.2 f kN",f_c);
36 printf("\nThe area of pin                = %.2 f mm^2",A);

```

```

37 printf("\nThe diameter of pin      = %.2 f mm",
        chosen_d);
38
39 //

```

END

Scilab code Exa 1.15 S15

```

1  clear all; clc;
2
3  disp("Scilab Code Ex 1.15 : ")
4
5  //Given:
6  P= 20; //kN
7  d_hole = 40; //mm
8  normal_allow = 60; //MPa
9  shear_allow = 35; //MPa
10
11
12 //Diameter of Rod:
13 area1 = (P*10^3)/(normal_allow*10^6); //Area in m^2
14 d = ((sqrt((4*area1)/%pi))*1000); // Area = (%pi\4)d
    ^2
15
16
17 //Thickness of disc:
18 V = P;
19 area2 = (V*10^3)/(shear_allow*10^6); //Area in m^2
20 thickness = (area2*10^6)/(d_hole*%pi); // A = pi*d*t
21
22
23 printf("\n\nThe cross sectional area of disc = %.8

```

```

    f m^2",area1);
24 printf("\nThe diameter of rode           = %.2 f
    mm",d);
25 printf("\nThe thickness of disc         = %.2 f
    mm",thickness);
26
27 //

```

END

Scilab code Exa 1.16 S16

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 1.16 : ")
4
5 //Given:
6 bearing_allow = 75; //MPa
7 tensile_allow = 55; //MPa
8 d_shaft = 60; //mm
9 r_shaft = d_shaft/2; //mm
10 area_shaft = %pi*(r_shaft^2); //Area = pi*r^2
11 d_collar = 80; //mm
12 r_collar = d_collar/2; //mm
13 area_collar = %pi*(r_collar^2); //Area = pi*r^2
14 thick_collar = 20; //mm
15
16 //Normal Stress:
17 P1 = (tensile_allow* area_shaft)/3; //Tensile stress
    = 3P/A.
18 P1_kN = P1/1000;
19
20

```



```

21 //Bearing Stress:
22 bearing_area = area_collar-area_shaft; //mm^2
23 P2 = (bearing_allow*bearing_area)/3; //Bearing
    stress = 3P/A.
24 P2_kN= P2/1000;
25
26 if(P2_kN<P1_kN)
27     big = P2_kN;
28 else big = P1_kN;
29     end
30
31 //Displaying Results:
32
33 printf("\n\nThe load calculated by Normal Stress
        = %.1f kN",P1_kN);
34 printf("\nThe load calculated by Bearing Stress
        = %.1f kN",P2_kN);
35 printf("\nThe largest load that can be applied to
        the shaft = %.1f kN",big);
36
37 //

```

END

Scilab code Exa 1.17 S17

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 1.17 : ")
4
5 //Given:
6 d_ac= 20; //mm
7 area_ac = %pi*(d_ac/2)^2; //Area = (%pi\4)d^2
8 area_al = 1800; //mm^2
9 d_pins = 18; //mm

```

```

10 area_pins = %pi*(d_pins/2)^2;
11 st_fail_stress = 680; //MPa
12 al_fail_stress = 70; //MPa
13 shear_fail_pin = 900; //MPa
14 fos = 2; //Factor of safety
15 l_ab = 2; //m
16 l_ap = 0.75; //m
17
18
19 st_allow= st_fail_stress /fos; //MPa
20 al_allow = al_fail_stress/fos; //MPa
21 pin_allow_shear = shear_fail_pin/fos; //MPa
22
23 //Rod AC
24 f_ac = (st_allow*area_ac)/1000;
25 P1 = ((f_ac*l_ab)/(l_ab-l_ap));
26
27 //Block B
28 f_b = (al_allow*area_al)/1000;
29 P2 = ((f_b*l_ab)/l_ap);
30
31 //Pin A or C:
32 V = (pin_allow_shear*area_pins)/1000;
33 P3 = (V*l_ab)/(l_ab-l_ap);
34
35 if(P1<P2 & P1<P3)
36     big = P1;
37 else if(P2<P1 & P2<P3)
38     big = P2;
39 else big = P3;
40 end
41
42 //Displaying Results:
43
44 printf("\n\nThe load allowed on rod AC
         = %.1f kN",round(P1));
45 printf("\n\nThe load allowed on block B
         = %.1f kN",P2);

```

```
46 printf("\nThe load allowed on pins A or C
          = %.1f kN",P3);
47 printf("\nThe largest load that can be applied to
          the bar = %.1f kN ",big);
48
49 //
```

END

Chapter 2

Strain

Scilab code Exa 2.1 Strain1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 2.1 : ")
4
5 //Given:
6 e_z= 4;
7 ab = 0.200; //m
8
9
10 //Calculations:
11
12 //Part a)
13
14 z=integrate('1+(40*10^-3)*(sqrt(z))', 'z', 0, ab); //
    Strain formula for short line segment = delta(
    sdash) =(1+e_z)delta(s)
15 deltaB= z-ab;
16 deltaB_mm= deltaB*1000;
17
18 //Part b)
19
```

```

20 e_avg = deltaB/ab; // Normal strain formula : e = (
    delta(sdash) -delta(s))/delta(s)
21
22 //Display:
23
24
25 printf("\n\nThe value of integration is
    =%10.5 f m",z);
26 printf("\nThe displacement at the end of the rod is
    = %0.2 f mm",deltaB_mm);
27 printf("\nThe average normal strain in the rod is
    =%10.4 f mm/mm",e_avg);
28
29 //

```

END

Scilab code Exa 2.2 Strain2

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 2.2 : ")
4
5 //Given:
6 theta = 0.002; //radians
7 bc=1; //m
8 ba = 0.5; //m
9
10 //Calculations:
11
12 bb_dash = theta*ba;
13 avg_normal_strain = bb_dash/bc; //m/m
14

```

```

15 //Display :
16
17
18 printf("\n\nThe average normal strain =%10.3f m/m",
        avg_normal_strain);
19
20 //-----END

```

Scilab code Exa 2.3 Strain3

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 2.3 : ")
4
5 //Given:
6
7 ab= 250; //mm
8 bbdash_x = 3; //mm
9 bbdash_y = 2; //mm
10 ac = 300; //mm
11
12 //calculations:
13
14 //Part(a)
15 abdash = sqrt((ab - bbdash_y)^2 + (bbdash_x)^2); //
        Pythagoras theorem
16 avg_normal_strain = (abdash-ab)/ab;
17
18 //Part(b)
19 gamma_xy = atan(bbdash_x/(ab - bbdash_y)); //shear
        strain formula
20
21 //Display :

```

```

22
23 printf("\n\nThe average normal strain along AB is
        =%10.5 f mm/mm", avg_normal_strain);
24 printf("\nThe average shear strain          =
        %10.5 f rad", gamma_xy);
25
26 //

```

END

Scilab code Exa 2.4 Strain4

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 2.4 : ")
4
5 //Given:
6 ab = 150; //mm
7 bc = 150; //mm
8 disp_cd= 2; //mm
9 ab_half = ab/2;
10 addash_half = (bc+disp_cd)/2 ;
11
12 //Calculations:
13
14 //Part(a)
15
16 ac = sqrt((ab)^2 + (bc)^2); //Pythagoras theorem in
    mm
17 ac_m = ac/1000; //in m
18 acdash = sqrt((ab)^2 + (bc+disp_cd)^2); //
    Pythagoras theorem in mm
19 acdash_m = acdash/1000; //in m
20

```

```

21 avg_strain_ac = (acdash_m - ac_m)/ac_m; //Normal
    strain formula
22
23 //Part(b)
24
25 theta_dash = 2* atan((addash_half)/(bc/2)); //theta
    found in radians
26 gamma_xy = (%pi / 2)- theta_dash; //shear strain
    formula
27
28 //Display:
29
30
31 printf("\n\nThe average normal strain along the
    diagonal AC is    =%10.5 f mm/mm",avg_strain_ac);
32 printf("\nThe shear strain at E relative to the x,y
    axes            = %10.5 f rad",gamma_xy);
33
34 //

```

END

Chapter 3

Mechanical Properties of Materials

Scilab code Exa 3.1 MPM1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 3.1 : ")
4
5 //Given:
6 offset = 0.2; //%
7 a_x = 0.0016; //mm/mm
8 a_y = 345; //Mpa
9
10 //Refer to the given graph.
11
12 //Calculations:
13
14 //Modulus of Elasticity
15 E = a_y/(a_x*10^3); //E is the slope in GPa.
16
17 //Yield Strength:
18 sigma_ys = 469; //Graphically , for a strain of 0.002
    mm/mm
```

```

19
20 //Ultimate Stress:
21 sigma_u = 745.2; //Mpa B is the peak of stress
    strain graph.
22
23 //Fracture Stress:
24 ep_f = 0.23; //mm/mm
25 sigma_f = 621; //Mpa from the graph.
26
27 //Display:
28
29 printf("\n\nThe Modulus of Elasticity is          =
    %10.1f GPa",E);
30 printf("\nThe Yield Strength from the graph      =
    %0.2f MPa",sigma_ys);
31 printf("\nThe Ultimate Stress from the graph is =%10
    .1f MPa",sigma_u);
32 printf("\nThe Fracture Stress from the graph is =%10
    .1f MPa",sigma_f);
33
34 //

```

END

Scilab code Exa 3.2 MPM2

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 3.2 : ")
4
5 //Given:
6 stress_b = 600; //MPa
7 strain_b = 0.023; //mm/mm

```

```

8 stress_a = 450; //Mpa
9 strain_a = 0.006; //mm/mm
10
11 //Calculations:
12
13 //Permanent Strain:
14 E = stress_a/strain_a;
15 strain_cd = stress_b/E; //The recovered elastic
    strain
16 perm_strain = strain_b - strain_cd; //mm/mm
17
18 //Modulus of Resilience:
19 ur_initial = (0.5*stress_a*strain_a); //MJ/m^3
20 ur_final = (0.5*stress_b*strain_cd); //MJ/m^3
21
22 //Display:
23
24 printf("\n\nThe Permanent Strain is          =
    %10.5 f mm/mm",perm_strain);
25 printf("\nThe Initial Modulus of Resilience is = %0
    .2 f MJ/mm^3",ur_initial);
26 printf("\nThe Final Modulus of Resilience is   = %0
    .2 f MJ/mm^3",ur_final);
27
28
29 //

```

END

Scilab code Exa 3.3 MPM3

```

1 clear all; clc;
2

```

```

3 disp(" Scilab Code Ex 3.3 : ")
4
5 //Given:
6 p = 10000; //N
7 E_al = 70*(10^3); //MPa
8 l_ab = 600; //mm
9 d_ab = 20; //mm
10 l_bc = 400; //mm
11 d_bc = 15; //mm
12
13 //Calculations:
14
15 a_ab = (%pi/4)*(d_ab^2); // Area of AB
16 a_bc = (%pi/4)*(d_bc^2);
17 stress_ab = p/a_ab; // Stress = load/area
18 stress_bc = p/a_bc;
19
20 e_ab = stress_ab/E_al; //Hookes' s Law. Elastic
    strain.
21 e_bc = 0.045; //mm/mm . From the graph for stress_bc
22
23 elongation = (l_ab*e_ab)+ (l_bc*e_bc);
24 strain_rec = stress_bc/E_al; //Strain Recovery
25
26 e_og = e_bc-strain_rec; // mm/mm
27 rod_elong = e_og*l_bc;
28
29 //Display:
30
31 printf("\n\nThe elongation of the rod when load is
    applied          =%10.1f mm",elongation);
32 printf("\n\nThe permanent elongation of the rod when
    load is removed = %0.1f mm",rod_elong);
33
34 //

```

END

Scilab code Exa 3.4 MPM4

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 3.4 : ")
4
5 //Given:
6 P = 80; //kN
7 l_z = 1.5; //m
8 l_y = 0.05; //m
9 l_x = 0.1; //m
10
11 //Calculations:
12 A= l_x*l_y;
13 normal_stress_z = (P*(10^3))/A; //Pa
14
15 Est = 200; //GPa - from the tables.
16 strain_z = (normal_stress_z)/(Est*(10^9)); // Strain
    = stress/modulus of elasticity
17
18 axial_elong = strain_z*l_z; //elongation in the y
    direction
19
20 nu_st = 0.32; //Poisson's Ratio - from the tables.
21 strain_x = -(nu_st)*(strain_z); //strain in the x
    direction.
22 strain_y = strain_x;
23
24 //Elongations:
25 delta_x = strain_x*l_x;
26 delta_y = strain_y*l_y;
27
28 //Display:
```

```

29
30 printf("\n\nThe change in the length (z direction)
      = %10.8f m", axial_elong);
31 printf("\nThe change in the cross section (x
      direction)= %10.8f m', delta_x);
32 printf("\nThe change in the cross section (y
      direction)= %10.8f m', delta_y);
33
34 printf("\n\nIn the standard form:")
35 printf("\nThe change in the length (z direction)
      = %10.2f x10^6m", (axial_elong*10^6));
36 printf("\nThe change in the cross section (x
      direction)= %10.2f x10^6m', (delta_x*10^6));
37 printf("\nThe change in the cross section (y
      direction)= %10.2f x10^6m', (delta_y*10^6));
38
39 //

```

END

Scilab code Exa 3.5 MPM5

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 3.5 : ")
4
5 //Given:
6 //Refer to the graph of shear stress-strain of
  titanium alloy.
7 x_A = 0.008; //rad - x co-ordinate of A
8 y_A = 360; //MPa - y co-ordinate of A
9 height = 50; //mm
10 l = 75; //mm

```

```

11 b = 100; //mm
12
13
14 //Calculations:
15
16 //Shear Modulus:
17 G = y_A/x_A;
18
19 //Proportional Limit:
20 tou_pl = 360; //Mpa Point A
21
22 //Ultimate Stresss:
23 tou_u = 504; //MPa – Max shear stress at B
24
25 //Maximum Elastic Displacement:
26 tanA= x_A;// tan theta is approximated as theta.
27 d = tanA*height;
28
29 //Shear Force:
30 A = l*b;
31 V = tou_pl*A;
32
33 //Display:
34
35
36 printf("\n\nThe Shear Modulus           = %10.2
    f MPa",G);
37 printf("\nThe Proportional Limit       = %10.2 f
    Mpa",tou_pl);
38 printf("\nThe Ultimate Shear Stress    = %10.2 f
    MPa ",tou_u);
39 printf("\nThe Maximum Elastic Displacement = %10.2 f
    mm",d);
40 printf("\nThe Shear Force               = %10.2 f
    kN ",(V/1000));
41
42 //

```

END

Scilab code Exa 3.6 MPM6

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 3.6 : ")
4
5 //Given:
6 d_o = 0.025; //m
7 l_o =0.25; //m
8 F =165; //kN
9 delta = 1.2; //mm
10 G_al = 26; //GPa
11 sigma_y = 440; //MPa
12
13 //Calculations:
14
15 //Modulus of Elasticity:
16 A = (%pi/4)*(d_o^2);
17 avg_normal_stress = (F*10^3)/A;
18 avg_normal_strain = delta/l_o;
19 E_al = avg_normal_stress/ avg_normal_strain;
20
21 E_al = E_al/10^6;
22
23 //Contraction of Diameter:
24 nu = (E_al/(2*G_al))-1;
25 strain_lat = nu*(avg_normal_strain) ;
26 d_contraction = strain_lat* d_o ;
27
28
29 //Display :
```



```
30
31 printf("\n\nThe Modulus of Elasticity
           = %10.1f GPa",E_a1);
32 printf("\nThe contraction in diameter due to the
           force = %10.4f mm",d_contraction);
33
34 //
```

END

Chapter 4

Axial Load

Scilab code Exa 4.1 AL1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 4.1 : ")
4
5 //Given:
6 a_ab = 600; //mm^2
7 a_bd = 1200; //mm^2
8 a_bc = a_bd;
9 p = 75; //kN
10 l_ab = 1; //m
11 l_bc = 0.75; //m
12 l_cd = 0.5; //m
13
14 // Calculations:
15
16 //Internal Forces: By method of Sections
17 P_bc = 35; //kN
18 P_cd = 45; //kN
19
20 // Displacement:
21 E_st = 210*(10^3); //From the tables
```

```

22
23 P = [p P_bc -P_cd];
24 A =[a_ab a_bc a_bd];
25 L= [l_ab l_bc l_cd];
26 E = []
27 n = length(P)
28
29 delta_sum =0;
30
31 for i = 1:n;
32     delta_sum = delta_sum + (P(i)*L(i)*(10^6))/(A(i)
        *E_st);
33 end
34
35 delta_bc = (P_bc*l_bc*10^6)/(a_bc*E_st);
36
37
38
39 //Display:
40
41 printf("\n\nThe vertical displacement of end A      =
        +%1.2f mm",delta_sum);
42 printf("\nThe displacement of B relative to C is = +
        %1.3f mm",delta_bc);
43
44 //

```

END

Scilab code Exa 4.2 AL2

```

1 clear all; clc;
2

```

```

3 disp(" Scilab Code Ex 4.2 : ")
4
5 //Given:
6 a_ab = 400; //mm^2
7 d_rod = 10; //mm
8 r_rod = d_rod/(2*1000); //radius in m
9 P = 80; //kN
10 E_st = 200*(10^9); //Pa
11 E_al = 70*(10^9); //Pa
12 l_ab = 400; //mm
13 l_bc = 600; //mm
14
15 //Calculations:
16
17 //Internal forces: tension = compression = 80kN.
18
19 //Displacement:
20
21 //delta =PL/AE
22 numerator1 = P*(10^3)*(l_bc/1000);
23 denominator1 = (%pi*r_rod^2*E_st);
24 delta_cb = numerator1/denominator1; //to the right
25
26 numerator2 = -P*(10^3)*(l_ab/1000);
27 denominator2 = (a_ab* 10^-6 *E_al);
28 delta_a = -numerator2/denominator2; //to the right
29
30 delta_c = delta_a+delta_cb;
31
32 //Display:
33
34
35
36 printf("\n\nThe displacement of C with respect to B
          = +%1.6f m', delta_cb);
37 printf("\n\nThe displacement of B with respect to A
          = +%1.6f m", delta_a);
38 printf('\n\nThe displacement of C relative to A

```

```

39         = +%1.5 f m', delta_c);
40 //-----
    END
-----

```

Scilab code Exa 4.3 AL3

```

1  clear all; clc;
2
3  disp("Scilab Code Ex 4.3 : ")
4
5  //Given:
6  d_ac = 20; //mm
7  r_ac = d_ac/(2*1000); //radius in m
8  d_bd =40; //mm
9  r_bd = d_bd/(2*1000); //radius in m
10 P = 90; //kN
11 E_st = 200*(10^9); //Pa
12 E_al = 70*(10^9); //Pa
13 l_af = 200; //mm
14 l_fb = 400; //mm
15 l_bd = 300; //mm
16 l_ac = l_bd;
17
18 //Calculations:
19
20 //Internal Force:
21 P_ac = 60; //kN
22 P_bd = 30; //kN
23
24 //Displacement:
25
26 //Post AC: delta = PL/AE

```

```

27 num1 = -(P_ac*10^3*(l_ac/1000));
28 denom1 = %pi* r_ac^2*E_st;
29 delta_a = -num1/denom1; //downwards
30 delta_a = delta_a*1000; //in m
31
32 //Post BD: delta = PL/AE
33 num2 = -(P_bd*10^3*(l_bd/1000));
34 denom2 = %pi* r_bd^2*E_al;
35 delta_b = -num2/denom2; //downwards
36 delta_b = delta_b*1000; //in m
37
38
39 delta_f = delta_b + (0.184)*(l_fb/(l_af+l_fb)); //By
    similar triangles from the figure.
40
41 //Display:
42
43 printf('\n\nThe displacement of Post AC      = +%1
    .3f mm downwards',delta_a);
44 printf('\n\nThe displacement of Post BD      = +%1.3 f
    mm downwards',delta_b);
45 printf('\n\nThe displacement of point F      = +%1.3 f
    mm downwards',delta_f);
46
47 //

```

END

Scilab code Exa 4.5 AL5

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 4.5 : ")

```

```

4
5 //Given:
6 d_ab = 5; //mm
7 A = (%pi/4)*(d_ab/1000)^2;
8 gap = 1; //mm
9 P = 20; //kN
10 E_st = 200; //GPa
11 l_ac = 0.4; //m
12 l_cb = 0.8; //m
13 l_ab = l_ac+l_cb;
14
15 //Calculations:
16
17 //Equilibrium:
18 // Eqn1: -Fa - Fb +P*10^3 = 0;
19
20 //Compatibility:
21 delta_ba = gap/1000; //in m
22
23 delta = delta_ba*(A*E_st*10^9); //delta_ba* Lac/AE
24
25
26 //Eqn2: (L/AE)*Fa -(Lb/AE)*Fb = delta_ba
27
28 //Solving Equations 1 and 2 by matrices:
29 coeff_F = [1 1; l_ac -l_cb];
30 b =[P*10^3 ; delta];
31 F = coeff_F\b;
32
33 F_a = F(1)/1000;
34 F_b = F(2)/1000;
35
36 //Display:
37
38
39 printf("\n\nThe reaction force at A = %1.1f kN",F_a)
;
40 printf("\n\nThe reaction force at B = %1.2f kN",F_b);

```

```
41
42 //


---


END


---


```

Scilab code Exa 4.6 AL6

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 4.6 : ")
4
5 //Given:
6 P = 45; //kN
7 E_al = 70*10^3;
8 E_br = 105*10^3;
9 h = 0.5; //m
10 ri = 25/1000; //m
11 ro = 50/1000; //m
12 A = (%pi*(ro^2 -ri^2));
13 Ai = %pi*ri^2;
14
15 //Calculations:
16
17 //Equilibrium: Eqn1: F_al +F_br = P
18
19 //Compatibility:
20 coeff_F_br = (A*E_al)/(Ai*E_br); // delta_al =
    delta_brass
21
22 //Eqn2 : F_al- (coeff_F_br*F_br) = 0
23
24 //Solving equations 1 and 2 using matrices:
25
```



```

26 coeff_F = [1 1; 1 -coeff_F_br];
27 b = [P; 0];
28 F = coeff_F\b;
29
30 F_al =F(1);
31 F_br =F(2);
32
33 avg_stress_al = F_al/A;
34 avg_stress_br = F_br/Ai;
35
36 avg_stress_al = avg_stress_al/1000;
37 avg_stress_br = avg_stress_br/1000;
38
39 //Display:
40
41
42 printf("\n\nThe axial force experienced by Al      =
      %1.1f kN",F_al);
43 printf("\n\nThe axial force experienced by Brass = %1
      .2f kN",F_br);
44 printf('\n\nThe average normal stress in Al      = %1
      .2f MPa',avg_stress_al);
45 printf('\n\nThe average normal stress in Al Brass = %1
      .2f MPa',avg_stress_br);
46
47 //

```

END

Scilab code Exa 4.7 AL7

```

1 clear all; clc;
2

```

```

3 disp(" Scilab Code Ex 4.7 : ")
4
5 //Given:
6 P = 15; //kN
7 a_ab = 25; //mm^2
8 a_ef = a_ab;
9 a_cd = 15; //mm^2
10 l_ef = 0.5; //m
11 l_ce = 0.4; //m
12 l_ac = 0.4; //m
13
14 //Calculations:
15
16 //Equilibrium:
17 //In the y direction ; F_a +F_c +F_e = P
18 //of moments: -F_a(l_ac)+ P(l_ac/2) +F_e(l_ce) = 0
19
20 //Compatibility equation for displacemnts:
21 coeff_Fc = (1/a_cd); //coefficient of Fc
22 coeff_Fa = (0.5/a_ab); //coefficient of Fa
23 coeff_Fe = (0.5/a_ef); //coefficient of Fe
24
25 //Using matrices to solve the 3 Equations:
26 A = [1 1 1; -l_ac 0 l_ce; coeff_Fa -coeff_Fc
      coeff_Fe];
27 b = [P ; -P*(l_ac/2); 0];
28 F = A\b;
29
30
31 F_a = F(1);
32 F_b = F(2);
33 F_c = F(3);
34
35 //Display:
36
37
38 printf("\n\nThe force in rod AB          = %1.2 f kN' ,
      F_a);

```

```

39 printf('\n\nThe force in rod CD      = %1.2 f kN', F_b)
    ;
40 printf('\n\nThe force in rod EF      = %1.2 f kN', F_c)
    ;
41
42 //-----
    END

```

Scilab code Exa 4.8 AL8

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 4.8 : ")
4
5  //Given:
6  r_o = 10; //mm
7  r_i = 5; //mm
8  l = 60; //mm
9  a_t = (%pi)*(r_o^2 - r_i^2); //Area of thread
10 a_b = (%pi*(r_i^2)); // Area of bolt
11 one_turn = 20/20;
12 E_am = 45; //GPa
13 E_al = 75; //GPa
14
15 //calculations:
16
17 //Equilibrium:
18 // In Y direction: F_b - F_t = 0
19
20 //Compatibility:
21 half_turn = one_turn/2;
22 coeff_Ft = 1/(a_t*E_am*10^3); // delta = PL/AE
23 coeff_Fb = 1/(a_b*E_al*10^3);

```

```

24
25 //Solving the two simultaneous equations for F_b and
    F_t:
26 A = [1 -1; coeff_Fb coeff_Ft];
27 b = [0 ; half_turn];
28 F = A\b;
29
30 F_b =F(1);
31 F_t = F(2);
32
33 stress_b = F_b/a_b;
34 stress_t = F_t/a_t;
35
36 F_b = F_b/1000; //in kN
37 F_t = F_t/1000; //in kN
38
39 //Display:
40
41
42 printf('\n\nThe force experienced by threads      =
    %1.2f kN',F_t);
43 printf('\nThe force experienced by the bolt      =
    %1.2f kN',F_b);
44 printf('\nThe stress in the screw                =
    %1.1f MPa',stress_t);
45 printf('\nThe stress in the bolt                 =
    %1.1f MPa',stress_b);
46
47 //

```

END

Scilab code Exa 4.9 AL9

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 4.9 : ")
4
5 //Given:
6 l_ab = 800 + 400; //mm
7 P = 20; //kN
8 d = 5/1000; //m
9 area = (%pi/4)*d^2; //Cross sectional area
10 l_bbdash = 1/1000; //m
11 E = 200; //GPa
12
13 //Calculations:
14
15 //Compatibility
16 delta_p = (P*10^3*0.4)/(area*E*10^9); //delta = PL/
    AE
17 delta_b = delta_p - l_bbdash;
18 F_b = (delta_b*area*E*10^9)/(l_ab/1000);
19 F_b = F_b/1000;
20
21 //Equilibrium:
22 F_a = P - F_b;
23
24 //Display:
25
26 printf("\n\nThe reaction at A      = %1.1f kN', F_a);
27 printf('\n\nThe reaction at B      = %1.1f kN', F_b);
28
29 //-----
    END
    -----
    -----

```

Scilab code Exa 4.10 AL10

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 4.10 : ")
4
5 //Given:
6 T1 = 30; //degree celcius
7 T2 = 60; //degress celcius
8 l_ab = 1; //m
9 area = 10*10*10^-6; //m^2
10 alpha = 12*10^-6; // per degree celcius
11 E = 200*10^6; //kPa
12
13 //Equilibrium:
14 //F_a = F_b = F
15
16 del_T = T2-T1;
17 F = alpha*del_T*area*E; //Thermal Stress Formula
18
19 avg_normal_comp_stress = (F*10^-3)/area; // sigma =
    F/A
20
21 //Display:
22
23 printf("\n\nThe force at A and B           =
    %1.1f kN",F);
24 printf('\n\nThe average normal compressive stress =
    %1.1f MPa',avg_normal_comp_stress);
25
26
27 //

```

END

Scilab code Exa 4.11 AL11

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 4.11 : ")
4
5 //Given:
6 area_sleeve = 600*10^-6; //m^2
7 area_bolt = 400*10^-6; //m^2
8 T1 = 15; //degree celcius
9 T2 = 80; //degree celcius
10 alpha_bolt = 12*10^-6; //per degree celcius
11 alpha_sleeve = 23*10^-6; //per degree celcius
12 l = 0.15; //m
13 E_bolt = 200*10^9; //N/m^2
14 E_sleeve = 73.1*10^9; //N/m^2
15
16 //Equilibrium:
17 //F_s = F_b
18
19 //Compatibility:
20 del_T = T2 - T1; // temperature difference
21 delb_T = alpha_bolt*del_T*l;
22 delb_F = 1/(area_bolt*E_bolt);
23 dels_T = alpha_sleeve*del_T*l;
24 dels_F = 1/(area_sleeve*E_sleeve);
25
26 //delb_T + F_b*delb_F = dels_T + F_s*dels_F
27
28 F_b = (dels_T-delb_T)/(delb_F+dels_F);
29 F_b = F_b/1000; //in kN
30 F_s= F_b;
31
32 sigma_b = F_b/(area_bolt*10^3); //Average Normal
    Stress
33 sigma_s = F_s/(area_sleeve*10^3); //Average Normal
    Stress
34
```

```

35 //Display:
36
37
38 printf("\n\nThe force experienced by sleeve and bolt
        = %1.2f kN",F_s);
39 printf('\n\nThe average normal stress on bolt
        = %1.1f MPa',sigma_b);
40 printf('\n\nThe average normal stress on sleeve
        = %1.1f MPa',sigma_s);
41
42
43 //

```

END

Scilab code Exa 4.12 AL12

```

1
2 clear all; clc;
3
4 disp(" Scilab Code Ex 4.12 : ")
5
6 //Given:
7 h = 0.250; //m
8 T1 = 20; //degree celcius
9 udl = 150; //kN/m
10 T2 = 80; //degree celcius
11 len = 0.3; //m
12 dia_steel = 0.04; //m
13 r_steel = 0.02;
14 dia_aluminium = 0.06; //m
15 r_al = dia_aluminium/2;
16 area_st = %pi*(r_steel^2);

```



```

17 area_al = %pi*(r_al^2);
18 F = 90*10^3; //N
19 alpha_st = 12*10^-6; //per degree celcius
20 alpha_al = 23*10^-6; //per degree celcius
21 E_st = 200*10^9; // N/m^2
22 E_al = 73.1*10^9; // N/m^2
23
24 //Equilibrium:
25 //From the free body diagram: Eqn1 : 2F_st + F_al-
26
27
28 // -delst_T + F_st*delst_F = -delal_T + F_al*delal_F
29
30 //Eqn2 : 165.9*10^3 = 1.216F_al - F_st F = 0
31
32 //Compatibility:
33 delst_T = alpha_st*(T1+T2)*h;
34 delst_F = h/(area_st*E_st);
35 delal_T = alpha_al*(T1+T2)*h;
36 delst_F = h/(area_al*E_al);
37
38 coeffMat = [2 1; -1 1.216]
39 b= [90*10^3 ; 165.9*10^3]
40 F = coeffMat\b;
41 F_st = F(1)/1000;
42 F_al =F(2)/1000;
43 F_al =ceil(F_al);
44
45 //Display:
46
47
48 printf("\n\nThe force on the steel post          =
      %1.1f kN",F_st);
49 printf('\n\nThe force on the aluminium post      =
      %1.1f kN',F_al);
50
51 //

```

END

Scilab code Exa 4.13 AL13

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 4.13 : ")
4
5 //Given:
6 sigma_allow = 115; //MPa
7
8 //Determinng the stress concentration factor:
9
10 r_n =10/20;
11 w_h = 40/20;
12 k = 1.4; //from graph
13 sigma_avg = sigma_allow/k;
14 P =sigma_avg*20*10;
15 P = P/1000;
16
17 //Display:
18
19 printf("\n\nThe largest axial force that the bar can
        carry      = %1.2f kN",P);
20
21 //
```

END

Scilab code Exa 4.14 AL14

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 4.14 : ")
4
5 //Given:
6 P = 80*10^3; //N
7 yield_stress = 700; //MPa;
8 E = 200*10^9; //N/mm^2
9 l1 = 0.3; //m
10 l2 = 0.8; //m
11
12 //Maximum Normal Stress:
13 r_h = 6/20;
14 w_h = 40/20;
15 K = 1.6;
16
17 area2 = 0.02*0.01; //m^2 note its not 0.001.
18 max_stress = (K*P)/area2;
19 max_stress = (max_stress/10^6); // converting to MPa
20
21 //Displacement:
22 area1 = 0.04*0.01;
23 del_ad_1 = (P*l1)/(area1*E);
24 del_ad_2 = (P*l2)/(area2*E);
25 del_ad = (2*del_ad_1)+ del_ad_2;
26 del_ad = del_ad*1000; //converting m to mm
27
28 //Display:
29
30
31 printf("\n\nThe maximum normal stress
           = %1.1f MPa",
           max_stress);
32 printf('\n\nThe displacement of one end with respect
           to the other = %1.2f mm',del_ad);
```

Scilab code Exa 4.15 AL15

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 4.15 : ")
4
5 //Given
6 weight = 15; //kN
7 l_ab = 5; //m
8 l_ac = 5.0075; //m
9 area = 30; //mm^2
10
11 //calculations:
12 strain_ab = (l_ac-l_ab)/l_ab;
13 max_strain = 0.0017;
14
15 stress_ab = (350*strain_ab)/max_strain;
16 F_ab = stress_ab*area; // F= stress*area
17 E_st = 350/max_strain; //Modulus of elasticity
18
19 del1 = l_ab/(area*10^-6*E_st*10^3); //del = PL/AE
20 del2 = l_ac/(area*10^-6*E_st*10^3); //del = PL/AE
21
22 //Eqn1 = T_ab + T_ac = weight
23 //Eqn2 = del1*T_ab - del2*T_ac = (l_ac-l_ab)
24
25 //Solving using matrices:
26 A = [1 1;del1 -del2];
27 b = [weight; (l_ac-l_ab)];
28 T = A\b;
29
30 T_ab = T(1);
31 T_ac = T(2);
32
```

```

33 stress_in_ab = (T_ab*10^3)/area;
34
35 if(stress_in_ab>350)
36     T_ab = (350*area)/1000;
37 end
38
39 T_ac = 15-T_ab;
40 stress = (T_ac*10^3)/area;
41 strain_ac = (stress*max_strain)/350;
42
43 elong_ac = strain_ac*l_ac; //m
44 elong_ab = (l_ac-l_ab)+elong_ac; //m
45
46
47
48 //Display:
49
50 printf('\n\nThe force experienced by wire AB = %1.1
      f kN',T_ab);
51 printf('\n\nThe force experienced by wire AC = %1.1 f
      kN',T_ac);
52 printf('\n\nThe elongation in wire AB          = %1.5 f
      m',elong_ab);
53 printf('\n\nThe elongation in wire AC          = %1.5 f
      m',elong_ac);
54
55 //

```

END

Scilab code Exa 4.16 AL16

```

1 clear all; clc;

```

```

2
3 disp(" Scilab Code Ex 4.16 : ")
4
5 //Given:
6 yield = 250; //MPa
7 r = 4; //mm
8 width = 40; //mm
9 thick = 2; //mm
10
11 //a)
12 r_h = r/(width - (2*r));
13 w_h = width/(width - (2*r));
14 K = 1.75;
15 area = (thick*(width - (2*r))*10^-6);
16 P_y = (yield*10^6*area)/K;
17 P_y = P_y/1000;
18
19 //b)
20 P_p = (yield*10^6*area);
21 P_p = P_p/1000;
22
23 //Display:
24
25 printf("\n\nThe maximum load P that does not cause
        the steel to yield      = %1.2f kN",P_y);
26 printf('\n\nThe maximum load that the bar can support
        = %1.2f kN',P_p);
27
28 //

```

END

Scilab code Exa 4.17 AL17

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 4.17 : ")
4
5 //Given:
6 r = 5/1000; //m
7 yield = 420; //MPa
8 E = 70; //GPa
9 P = 60; //kN
10 l_ac = 100/1000; //m
11 l_cb = 300/1000; //m
12 F_a = 45; //kN by elastic analysis
13 F_b = 15; //kN by elastic analysis
14
15 //Calculations:
16 area = %pi*(r^2)
17 sigma_ac = F_a/(area*1000);
18 sigma_ac1 = sigma_ac;
19 sigma_cb = F_b/(area*1000);
20 sigma_cb1 = sigma_cb;
21
22 if(sigma_ac>yield)
23     F_a_y = yield*10^3*area;
24     F_b = P - F_a_y;
25
26     sigma_ac = yield;
27     sigma_cb = F_b/(area*1000);
28 end
29
30 //Residual Stress:
31 defl_c = (F_b*l_cb)/(area*E*10^6);
32 strain_cb = defl_c/l_cb;
33 strain_ac = -defl_c/l_ac;
34
35 sigma_ac_r = -sigma_ac+ sigma_ac1;
36 sigma_cb_r = sigma_cb - sigma_cb1;
37
38     sigma = sigma_cb_r;

```

```
39
40 //Permanent Displacement:
41 res_strain_cb = (sigma*10^6)/(E*10^9);
42 perm_defl_c = res_strain_cb*l_cb*1000;
43
44
45 //Display:
46
47 printf("\n\nThe residual stress in AC
          = %1.1f MPa",sigma_ac_r)
      ;
48 printf("\n\nThe residual stress in CB
          = %1.1f MPa",sigma_cb_r)
      ;
49 printf("\n\nThe permanent displacement of the collar
          at C      = %1.3f mm",perm_defl_c);
50
51 //
```

END

Chapter 5

Torsion

Scilab code Exa 5.1 T1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 5.1 : ")
4
5 //Given:
6 r = 50; //mm
7 J = (%pi/2)*(r^4); //polar moment of inertia
8 tou_max = 56; //MPa
9 T = (tou_max*J)/(r*10^6); //toumax = Tc/J
10
11 //Display:
12
13
14 printf("\n\nThe resultant internal torque      = %1.0 f
15         kNm' ,T);
16 //-----
17         END
```

Scilab code Exa 5.3 T3

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.3 : ")
4
5 //Given:
6 T1 = 4250; //kNmm
7 T2 = -3000; //kNm
8 T3 = T1+T2; //kNm
9 r = 75; //mm
10
11 //Section Property:
12 J = (%pi/2)*(r^4); //polar moment of inertia
13
14 //Shear Stress:
15 c_a = 75; //mm
16 tou_a = (T3*c_a*1000)/J; //tou = Tc/J
17
18 c_b = 15; //mm
19 tou_b = (T3*c_b*1000)/J; //tou = Tc/J
20
21 //Display:
22
23 printf('\n\nThe shear stress developed at A = %1.2
24         f MPa', tou_a);
25 printf('\n\nThe shear stress developed at B = %1.3 f
26         MPa', tou_b);
27
28 //
```

END

Scilab code Exa 5.4 T4

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.4 : ")
4
5 //Given:
6 di = 80; //mm
7 ri = 40/1000; //m
8 d0 = 100; //mm
9 ro = d0/2000; //m
10 F = 80; //N
11 l1 = 0.2; //m
12 l2 = 0.3; //m
13
14 //Internal Torque:
15 T = F*(l1+l2);
16
17 //Section Property:
18 J = (%pi/2)*((ro^4)-(ri^4));
19
20 //Shear Stress:
21 c_o = 0.05; //m
22 tou_o = (T*c_o)/(J*10^6);
23
24 c_i = 0.04; //m
25 tou_i = (T*c_i)/(J*10^6);
26
27 //Display:
28
29
30 printf('\n\nThe shear stress in the inner wall    =
    %1.3f MPa', tou_i);
31 printf('\n\nThe shear stress in the outer wall    = %1
```

```
        .3 f MPa', tou_o);  
32  
33  
34 //
```

END

Scilab code Exa 5.5 T5

```
1 clear all; clc;  
2  
3 disp("Scilab Code Ex 5.5 : ")  
4  
5 //Given:  
6 P = 3750; //W  
7 N = 175; //rpm  
8 allow_shear = 100; //MPa  
9  
10 //Calculations:  
11 ang_vel = (2*pi*N)/60; // rad/s  
12 T = P/ang_vel; //P = T*angular velocity  
13  
14 c = ((2*T*1000)/(pi*allow_shear))^(1/3);  
15 d = round(2*c);  
16  
17 //Display:  
18  
19  
20 printf('\n\nThe required diameter of the shaft =  
    %1.0f mm', d);  
21  
22 //
```

END

Scilab code Exa 5.6 T6

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.6 : ")
4
5 //Given:
6 di = 30; //mm
7 ri= (di/2000); //m
8 d0 = 42; //mm
9 ro = (d0/2000); //m
10 P = 90; //kW
11 max_shear = 50; //MPa
12
13 //Calculations:
14 c = ro; //m
15 J = (%pi/2)*((ro^4)-(ri^4)); //Polar moment of
    inertia of hollow shaft
16 T = (max_shear*J)/c; //tau max = Tc/J
17
18 //P = 2(%pi)fT
19 f = (P)/(2*%pi*T*10^3);
20
21 //Display:
22
23
24 printf('\n\nThe required frequency of rotation of
    the shaft = %1.1f Hz',f);
25
26 //
```

END

Scilab code Exa 5.7 T7

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.7 : ")
4
5 //Given:
6 E = 80*10^3; //MPa
7 d = 14/1000; //m
8 r = d/2; //m
9 R = 100; //mm
10 l_ac = 0.4; //m
11 l_cd = 0.3; //m
12 l_de = 0.5; //m
13 T_c = 280; //Nm
14 T_a = 150; //Nm
15 T_d = 40; //Nm
16 T_ac = T_a; //Nm
17 T_cd = T_ac - T_c;
18 T_de = T_cd - T_d;
19
20 //Angle of Twist:
21 J = (%pi/2)*(r^4);
22
23 T = [T_ac T_cd T_de];
24 l = [l_ac l_cd l_de];
25
26 sumTwist = 0;
27
28 for i= 1:3
29     sumTwist = sumTwist+ ((T(i)*l(i))/(J*E*10^6));
```

```

30 end
31
32 displacement = - sumTwist*R;
33
34 //Display:
35
36
37 printf('\n\nThe angle of twist of the shaft
          = %1.3f rad',sumTwist);
38 printf('\n\nThe displacement of tooth P on gear A    =
          %1.1f mm',displacement);
39
40 //

```

END

Scilab code Exa 5.8 T8

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 5.8 : ")
4
5 //Given:
6 T = 45; //N
7 G = 80; //GPa
8 d = 20/1000; //m
9 r = d/2; //m
10 l_dc = 1.5; //m
11 l_ab = 2; //m
12 r1 = 75/1000; //m
13 r2 = 150/1000; //m
14
15 //Internal Torque:

```

```

16 F = T/r2;
17 T_d_x = F*r1;
18
19 //Angle of twist:
20 J = (%pi/2)*(r^4);
21 phi_c = (T*l_dc)/(2*J*G*10^9);
22 phi_b = (phi_c*r1)/r2;
23
24 phi_ab = (T*l_ab)/(J*G*10^9);
25
26 phi_a = phi_b + phi_ab;
27
28 //Display:
29
30
31 printf('\n\nThe angle of twist of end A of shaft AB
        = + %1.4f rad',phi_a);
32
33 //

```

END

Scilab code Exa 5.9 T9

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 5.9 : ")
4
5 //Given:
6 d = 50; //mm
7 r = d/2;
8 c = d/2;
9 l_buried = 600; //mm

```



```

10 G = 40*10^3; //MPa
11 F = 100; //N
12 l_handle= 150; //mm
13 l_ab = 900; //mm
14
15 //Internal Torque:
16 T_ab = F*2*l_handle;
17 t = T_ab/l_buried;
18
19 //Maximum Shear Stress:
20 J = (%pi/2)*(r^4);
21 tou_max = (T_ab*c)/(J);
22
23 //Angle of Twist:
24
25 x0=0;
26 x1=l_buried;
27 X=integrate('x','x',x0,x1);
28
29 phi_a = ((T_ab*l_ab)+(50*X))/(J*G);
30
31 //Display:
32
33
34
35 printf('\n\nThe maximum shear stress in the post
          = %1.2 f N/mm^2 ',tou_max);
36 printf('\n\nThe angle of twist at the top of the post
          = %1.5 f rad ',phi_a);
37
38 //

```

END

Scilab code Exa 5.11 T11

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.11 : ")
4
5 //Given:
6 d = 20/1000; //m
7 r = d/2;
8 l_bc = 0.2;
9 l_cd = 1.5;
10 l_da = 0.3;
11 T_c = 800; //Nm
12 T_d = -500; //Nm
13
14 //Equilibrium:
15 //Eqn 1 :      300 = T_a + T_b
16
17 //Compatibility:
18 //Eqn 2:
19 coeff_Tb = -l_bc;
20 coeff-Ta = l_cd + l_da;
21
22 //Solving Equations simultaneously using matrices:
23 C = [1 1; coeff_Tb coeff-Ta];
24 b = [300 ; -750];
25 T = C\b;
26
27 T_b = T(1);
28 T_a = T(2);
29
30 //Display:
31
32
```

```

33 printf('\n\nThe reaction at A    = %1.0 f Nm',T_a);
34 printf('\n\nThe reaction at B    = %1.0 f Nm',T_b);
35
36 //

```

END

Scilab code Exa 5.12 T12

```

1  clear all; clc;
2
3  disp("Scilab Code Ex 5.12 : ")
4
5  //Given:
6  T = 250; //Nm
7  G_st = 80; //GPa
8  G_br = 36; //GPa
9  ri = 10; //mm
10 ro = 20; //mm
11 l_ab = 1.2; //m
12
13 //Equilibrium:
14 // -Tst-Tbr+250Nm = 0
15 coeff1_st = -1;
16 coeff1_br = -1;
17 b1 = -250;
18
19 //Compatibility:
20 //phi = TL/JG
21
22 J1 = (%pi/2)*(ro^4 - ri^4);
23 J2 = (%pi/2)*(ri^4);
24 coeff2_st = 1/(J1*G_st*10^3);

```

```

25  coeff2_br = -1/(J2*G_br*10^3);
26  b2 = 0;
27
28  //Solving the above two equations simultaneously
    using matrices:
29  A = [coeff1_st coeff1_br;coeff2_st coeff2_br ];
30  b = [b1 ; b2];
31  T = A\b;
32
33  T_st = T(1);
34  T_br = T(2);
35
36  shear_br_max = (T_br*10^3*ri)/(J2); //tau = (Tr)/J
37  shear_st_min = (T_st*10^3*ri)/(J1); //tau = (Tr)/J
38  shear_st_max = (T_st*10^3*ro)/(J1); //tau = (Tr)/J
39
40  shear_strain = shear_br_max / G_br;
41  shear_strain = shear_strain;
42
43  //Display:
44
45
46  printf('\n\nThe Torque acting on Steel
           = %1.2 f Nm',T_st);
47  printf('\n\nThe Torque acting on Brass
           = %1.2 f Nm',T_br);
48  printf('\n\nThe maximum shear stress experienced by
           Steel = %1.2 f MPa',shear_st_max);
49  printf('\n\nThe minimum shear stress experienced by
           Steel = %1.2 f MPa',shear_st_min);
50  printf('\n\nThe maximum shear stress experienced by
           Brass = %1.2 f MPa',shear_br_max);
51  printf('\n\nThe shear strain at the interface
           = %1.5 f *10^-3 rad',shear_strain);
52
53
54  //

```

END

Scilab code Exa 5.13 T13

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.13 : ")
4
5 //Given:
6 l = 1.2; //m
7 a = 40; //mm
8 tou_allow = 56; //MPa
9 phi_allow = 0.02; //rad
10 G = 26; //GPa
11 alpha = (60*%pi)/180; //degrees
12
13 //Calculations:
14 T_shear1 = (tou_allow*a^3)/(20*1000); // allowable
    shear stress = (20T)/(a^3)
15 T_twist1 = (phi_allow*a^4*G*10^3)/(46*1*10^6); //
    angle of twist =(46TL)/(a^4*G)
16
17 T1 = min(T_shear1, T_twist1);
18
19 //Circular Cross Section:
20 c_ = (a*a*sin(alpha))/(%pi*2);
21 c = sqrt(c_);
22
23 J = (%pi/2)*(c^4);
24 T_shear2 = (tou_allow*J)/(c*1000);
25 T_twist2 = (phi_allow*J*G*10^3)/(1*10^6);
26
27 T2 = min(T_shear2, T_twist2);
```

```

28
29
30 //Display:
31
32 printf('\n\nThe largest torque that can be applied
      at the end of the triangular shaft      = %1.2f Nm'
      ,T1);
33 printf('\n\nThe largest torque that can be applied at
      the end of the circular shaft          = %1.2f Nm',T2
      );
34
35
36 //

```

END

Scilab code Exa 5.15 T15

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 5.15 : ")
4
5 //Given:
6 l_cd = 0.5; //m
7 l_de = 1.5; //m
8 h =60/1000; //m
9 w = 40/1000; //m
10 t_h = 3/1000; //m
11 t_w = 5/1000; //m
12 T_c = 60; //Nm
13 T_d = 25; //Nm
14 G = 38*10^9; //N/m^2
15 T1 = T_c - T_d;

```

```

16
17 //Average Shear Stress:
18 area = (w-t_w)*(h-t_h);
19
20 shear_a = T1/(2*t_w*area*10^6);
21 shear_b = T1/(2*t_h*area*10^6);
22
23 //Angle of Twist:
24
25 ds_t = 2*((w-t_w)/t_h)+((h-t_h)/t_w));
26 T = [T_c T1];
27 l = [l_cd l_de];
28 phi = 0;
29
30 for i = 1:2
31     phi = phi+ (T(i)*l(i)*ds_t)/(4*area^2*G);
32
33 end
34
35 //Display:
36
37 printf('\n\nThe average shear stress of the tube at
    A     = %1.2 f MPa',shear_a);
38 printf('\n\nThe average shear stress of the tube at B
    = %1.2 f MPa',shear_b);
39 printf('\n\nThe angle of twist of end C
    = %1.6 f rad',phi);
40
41 //

```

END

Scilab code Exa 5.16 T16

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 5.16 : ")
4
5 //Given:
6 T = 85; //Nm
7 G = 26; //GPa
8 t = 10; //mm thickness
9 a = 60; //mm side
10 l = 1.5; //m
11
12 //Average Shear Stress:
13 area_m = (a-t)*(a-t);
14 avg_shear = (T*10^3)/(2*t*area_m); //tau_avg = T/(2
    tarea_m);
15
16
17 //Angle of Twist:
18 ds_t = (4*(a-t))/t;
19 phi = (T*10^3*l*10^3*ds_t)/(4*(area_m^2)*G*10^3);
20
21 //Display:
22
23
24 printf('\n\nThe average shear stress in the tube at
    A = %1.1f N/mm^2', avg_shear);
25 printf('\n\nThe angle of twist due to loading
    = %1.5f rad', phi);
26
27 //

```

END

Scilab code Exa 5.17 T17

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.17 : ")
4
5 //Given:
6 tou_allow = 90; //MPa
7 phi_allow = 2*10^-3; //rad
8 a = 200; //mm side
9 angle = (60*%pi)/180;
10 h = a*sin(angle);
11 l = 3; //m
12 t = 5/1000; //m
13 G = 75*10^9; //N/mm^2
14
15 //Calculations:
16 area_m = 0.5*a*h*10^-6; //m^2 a = (1/2)bh
17 ds_t = (3*a)/(t*1000);
18
19 T_shear = (tou_allow*10^6*2*t*area_m); //tou_avg = T
    /(2tarea_m);
20
21 T_twist = (phi_allow*4*area_m^2*G)/(l*ds_t);
22
23 T = min(T_shear, T_twist);
24
25
26 //Display:
27
28
29 printf('\n\nThe maximum torque that the thin tube
    can be subjected to = %1.1f Nm',T);
30
31 //
```

END

Scilab code Exa 5.18 T18

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 5.18 : ")
4
5 //Given:
6 fillet_r = 6; //mm
7 D = 40/1000; //m
8 d = 20/1000; //m
9 T = 30; //Nm
10 D_d = D/d;
11 r_d = fillet_r/d;
12 k = 1.3;
13
14 //Maximum Shear Stress:
15 c = D/2;
16 J = (%pi/2)*(c^4)
17 max_shear = (k*T*c)/(J*10^6); // tou = K(Tc/J)
18
19 //Display:
20
21 printf('\n\nThe maximum shear stress in the shaft
    due to the applied torques      = %1.2f MPa',
    max_shear);
22
23 //
```

END

Scilab code Exa 5.19 T19

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 5.19 : ")
4
5 //Given:
6 ro = 50/1000; //m
7 ri = 30/1000; //m
8 c = ro;
9 shear = 20*10^6; //N/m^2
10
11 //Maximum Elastic Torque:
12 J = (%pi/2)*((ro^4)-(ri^4));
13 T_y = (shear*J)/c; // tou = Tc/J
14 T_y = T_y/1000; //in kN
15
16 //Plastic Torque:
17 x0 = 0.03;
18 x1 = 0.05;
19 I = integrate('rho^2','rho',x0,x1)
20 Tp = (2*%pi*I*shear);
21 Tp= Tp/1000;
22
23 //Outer Shear Strain:
24 strain = (0.286*10^-3*ro)/(ri);
25
26 //Display:
27
28
29 printf('\n\nThe maximum torque that can be applied
    to the shaft without causing the material to
    yield = %1.2f kNm',T_y);
30 printf('\n\nThe plastic torque that can be applied to
```

```

    the shaft
                                                    = %1.2 f
    kNm',Tp);
31 printf('\nThe minimum shear strain at the outer
    radius of the shaft
                                                    = %1.7 f rad',
    strain);
32
33
34 //

```

END

Scilab code Exa 5.20 T20

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 5.19 : ")
4
5 //Given:
6 r = 20/1000; //m
7 l = 1.5; //m
8 phi = 0.6; //rad
9 shear_y = 75*10^6; //N/m^2
10
11 //Calculations:
12 max_shear_strain = (phi*r)/(l); //phi = (strain*L)/r
13 strain_y = 0.0016;
14
15 r_y = (r*strain_y)/(max_shear_strain); //by ratios
16
17 //T= (%pi*shear_y)*(4c^3 - r_y^3)/6;
18 c = r;

```

```

19
20 T = (%pi*shear_y)*(4*c^3 - r_y^3)/6;
21 T = T/1000;
22
23 //Display:
24
25 printf('\n\nThe torque needed to twist the shaft by
      0.6 rad      = %1.2f kNm',T);
26
27 //

```

END

Scilab code Exa 5.21 T21

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 5.21 : ")
4
5 //Given:
6 l = 1.5; //m
7 G = 42*10^3; //GPa
8 co = 50; //mm
9 ci = 25; //mm
10 shear_y = 84; //N/mm^2
11 strain_y = 0.002; //rad
12
13 //Plastic Torque:
14 T_p = ((2*pi)*(co^3 - ci^3)*shear_y)/3;
15 phi_p = (strain_y*l*10^3)/ci;
16
17 J = (%pi/2)*(co^4 - ci^4);
18 shear_r = (T_p*co)/J;

```

```

19 shear_i = (shear_r*ci)/(co); // shear = Tc/J
20
21 G = shear_y/strain_y;
22
23 phi_dash = (T_p*1*10^3)/(J*G); //phi = TpL/JG;
24
25 phi = phi_p - phi_dash;
26 T_p = T_p/10^6;
27
28 //Display:
29
30
31 printf('\n\nThe plastic torque Tp
           = %1.2f x 10^6
           Nmm',T_p);
32 printf('\nThe permanent twist of the tube if Tp is
           removed = %1.5f rad',phi);
33
34
35 //

```

END

Chapter 6

Bending

Scilab code Exa 6.5 B5

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.5 : ")
4
5 //Shear and Moment Diagrams:
6 p = [-1/9 -2 30]
7 x = roots(p)
8 y = (x(2));
9
10
11     M = (30*y) - (y^2) - (y^3)/27;
12
13
14
15 //Display:
16
17 printf("\n\nThe magnitude of the maximum moment is =
18         %1.0f kNm', M);
19 printf('\nRefer to the shear and moment diagrams in
20         the book.');
```

```
20
21 //-----
    END
-----
```

Scilab code Exa 6.11 B11

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.11 : ")
4
5 //Given:
6 l = 4.5; //m
7 R1 = 1.5; //kN
8 R2 = 3; //kN
9 uv1 = 2; //kN/m
10
11 //Shear diagram:
12 x = sqrt((2*R1*l)/(uv1));
13 M = (R1*x) - (0.5*uv1*x^3)/(3*l);
14
15 //Display:
16
17
18 printf('\n\nV becomes zero at x = %1.1fm',x);
19 printf('\nThe magnitude of the maximum moment = %1
    .1f kNm',M);
20
21 //-----
    END
-----
```


Scilab code Exa 6.13 B13

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 6.13 : ")
4
5 //Given:
6 l_ab = 4; //m
7 l_cd = 4; //m
8 l_bc = 6; //m
9 Rb = 8; //kN
10 uv1 = 2; //kN/m
11
12 //Moment diagram:
13 p = [-1/18 0 -3.6 17.6]
14 x = roots(p)
15 y = x(3);
16
17 //Display:
18
19 printf('\n\nV becomes zero at x = %1.2f m',y);
20
21
22 //
```

END

Scilab code Exa 6.14 B14

```
1 clear all; clc;
```

```

2
3 disp(" Scilab Code Ex 6.14 : ")
4
5 //Given:
6 b = 60; //mm
7 h = 120; //mm
8 sigma_max = 20; //N/mm^2
9 c = b;
10
11 //Part (a):
12 I = (1/12)*b*h^3;
13 M1 = (sigma_max*I)/(c); //sigma_max = Mc/I Flexure
    Formula
14 M1 = M1*10^-6; //in kN/m
15
16 //Part (b):
17 y0=60;
18 y1=-60
19
20 M2 = integrate('-(20*y^2)', 'y', y0, y1);
21 M2 = M2*10^-6;
22
23 F = (0.5*sigma_max*b*b);
24 c = 2*(60 -(0.5*b)); //distance between centroids of
    both the volumes.
25 M = F*c/1000;
26
27 //Display:
28
29 printf("\n\nThe internal moment M calculated using
    : ");
30 printf('\na)The flexure formula = %1.2f kNm',M1);
31 printf('\nb)The resultant of the stress
    distribution using the basic principles = %1.2f
    kNm',M2);
32
33
34 //

```

END

Scilab code Exa 6.15 B15

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.15 : ")
4
5 //Given:
6 udl = 5; //kN/m
7 l1 = 3; //m
8 l2 = 6; //m
9 t = 20/1000; //mm
10 yb = 0.15; //m
11
12 //Section Property:
13 I_bar1 = (1/12)*(0.25)*(0.02^3);
14 Ad2 = (0.25)*(0.02)*(yb+(t/2))^2;
15 I_bar2 = (1/12)*(0.02)*(0.3^3);
16 I = 2*(I_bar1 + Ad2) + I_bar2;
17
18 //Bending stress:
19 c = 0.15 + t;
20 M= 22.5; //kNm
21
22 sigma_max = (M*c)/(I*1000);
23
24 sigma_B = (M*yb)/(I*1000);
25
26 //Display:
27
28 printf('\n\nThe absolute maximum bending stress is
```

```

    = %1.1f MPa',sigma_max);
29
30
31 //

```

END

Scilab code Exa 6.16 B16

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 6.16 : ")
4
5 //Given:
6 t1 = 15/1000; //m
7 t2 = 20/1000; //m
8 l = 250/1000; //m
9 b = 200/1000; //m
10 P = 2.4; //kN
11 l_a = 2; //m
12 l_b = 1; //m
13
14 //Internal Moment:
15 y1 = b/2;
16 y2 = t2/2;
17 A = (2*t1*b)+(t2*l);
18 y_bar = ((2*y1*t1*b)+(y2*t2*l))/A;
19
20 M = (P*l_a)+(1*y_bar);
21
22 //Section Property:
23 I1 = (1/12)*(l*t2^3) + (l*t2*(y_bar - y2)^2);
24 I2 = (1/12)*(t1*b^3) + (t1*b*(y1 - y_bar)^2);

```

```

25 I = I1 + 2*I2;
26
27 //Maximum Bending Stress:
28 c = b - y_bar;
29 sigma_max = (M*c)/(I*1000);
30
31 //Display:
32
33 printf('\n\nThe maximum bending stress at section
      a-a = %1.1f MPa', sigma_max);
34
35
36 //

```

END

Scilab code Exa 6.17 B17

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.17 : ")
4
5 //Given:
6 b = 60/1000; //m
7 h = 30/1000; //m
8 M = 40; //Nm
9 c1 = h/2;
10 rib_t = 5/1000; //m
11 rib_w = 10/1000; //m
12
13 //Without Ribs:
14 I1 = (1/12)*(b*h^3);
15 sigma_max1 = (M*c1)/(I1*10^6);

```

```

16
17 //With Ribs:
18 y1 = c1;
19 y2 = h+(rib_t/2);
20 A1 = h*b;
21 A2 = rib_t*rib_w;
22 y_bar = ((y1*A1)+2*(y2*A2))/(A1 + 2*A2);
23
24 c2 = h+rib_t - y_bar;
25 I2 = I1 + (b*h*(y_bar - y1)^2);
26 I3 = (1/12)*rib_w*rib_t^3 + (rib_w*rib_t*(y2 - y_bar
    )^2);
27 I = I2 + 2*I3;
28
29 sigma_max2 = (M*c2)/(I*10^6);
30
31 if(sigma_max2>sigma_max1)
32
33     printf("\n\nThe maximum normal stress in the
        member without ribs = %1.2f MPa',sigma_max1);
34     printf("\n\nThe maximum normal stress in the member
        with ribs = %1.2f MPa',sigma_max2);
35     printf("\n\nThe ribs should be omitted.");
36
37 end
38
39
40 //

```

END

Scilab code Exa 6.18 B18

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.18 : ")
4
5 //Given:
6 M = 12; //kNm
7 l_bc = 0.2; //m
8 l_be = 0.4; //m
9
10 //Internal Moment Components:
11 My = (-4/5)*M;
12 Mz = (3/5)*M;
13
14 Iy = (1/12)*(l_be*l_bc^3);
15 Iz = (1/12)*(l_bc*l_be^3);
16
17 //Bending Stress:
18 sigma_B = (-Mz*1000*(l_be/2))/Iz + (My*1000*(-l_bc
    /2))/Iy;
19 sigma_B = sigma_B/10^6;
20 sigma_C = (-Mz*1000*(l_be/2))/Iz + (My*1000*(l_bc/2)
    )/Iy;
21 sigma_C = sigma_C/10^6;
22 sigma_D = (-Mz*1000*(-l_be/2))/Iz + (My*1000*(l_bc
    /2))/Iy;
23 sigma_D = sigma_D/10^6;
24 sigma_E = (-Mz*1000*(-l_be/2))/Iz + (My*1000*(-l_bc
    /2))/Iy;
25 sigma_E = sigma_E/10^6;
26
27 //Orientation of Nuetral Axis:
28 z = (0.45)/(sigma_E + sigma_B);
29
30 //theta = -atan(4/3);
31 tanA = (Iz/Iy)*(-4/3);
32 alpha = atan(tanA);
33 alpha = alpha*(180/%pi);
34

```

```

35
36 //Display:
37
38
39 printf("\n\nThe normal stress at B = %1.2f MPa',
        sigma_B);
40 printf("\n\nThe normal stress at C = %1.2f MPa',
        sigma_C);
41 printf("\n\nThe normal stress at D = %1.2f MPa',
        sigma_D);
42 printf("\n\nThe normal stress at E = %1.2f MPa',
        sigma_E);
43 printf("\n\nThe orientation of the neutral axis = %1
        .1f degrees ',alpha);
44
45 //-----
        END

```

Scilab code Exa 6.19 B19

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.19 : ")
4
5 //Given:
6 theta = 30*(%pi/180);
7 M = 15; //kNm
8 My = M*cos(theta);
9 Mz = M*sin(theta);
10 b = 0.1; //m
11 t1 = 0.04; //m
12 t2 = 0.03; //m
13

```



```

14
15 //Section Properties:
16 y1 = b/2;
17 y2 = b + t2/2;
18 A1 = (b*t1);
19 A2 = (b*2*t2);
20 z_bar = (y1*A1 + y2*A2)/(A1+A2);
21
22 Iz = (1/12)*(b*t1^3) + (1/12)*(t2*(2*b)^3);
23 Iy = (1/12)*(t1*b^3) + b*t1*(z_bar - y1)^2 + (1/12)
      *(2*b*t2^3) + 2*b*t2*(y2 - z_bar)^2;
24
25 //Maximum Bending Stress:
26 l_b = b+t2 - z_bar;
27 sigma_B = (-Mz*1000*(-b))/Iz + (My*1000*(l_b))/Iy;
28 sigma_B = sigma_B/10^6;
29 sigma_C = (-Mz*1000*(t1/2))/Iz + (My*1000*(-z_bar))/
      Iy;
30 sigma_C = sigma_C/10^6;
31
32 sigma = max(abs(sigma_B),abs(sigma_C));
33
34 //Orientation of the neutral axis:
35 theta1 = 60*(%pi/180);
36 alpha = atan((Iz/Iy)*tan(theta1));
37 alpha = alpha*(180/%pi);
38
39 //Display:
40
41
42 printf("\n\nThe maximum normal stress in the beam =
      %1.2f MPa', sigma);
43 printf("\n The orientation of the neutral axis = %1
      .1f degrees', alpha);
44
45 //

```

END

Scilab code Exa 6.20 B20

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.20 : ")
4
5 //Given:
6 M =20; //kN
7 Iy = 0.96*10^-3; //m^4
8 Iz = 7.54*10^-3; //m^4
9 theta = 57.1*(%pi/180);
10
11
12 //Internal moment Components:
13 My = M*sin(theta);
14 Mz = M*cos(theta);
15
16 //Bending Stress:
17 y_p = -0.2; //y Coordinate of P
18 z_p = 0.35; //z Coordinate of P
19
20 theta1 = (%pi/2)-(theta);
21 yp = -z_p*sin(theta1)+ y_p*cos(theta1);
22 zp = z_p*cos(theta1) + y_p*sin(theta1);
23
24 //Eq 6-17
25
26 sigma_p = ((Mz*-yp)/Iz) + ((My*zp)/Iy) ;
27 sigma_p = sigma_p/10^3;
28
29 //Orientation of the Neutral Axis:
30 alpha = atan((Iz/Iy)*tan(theta));
```

```

31 alpha = alpha*(180/%pi);
32
33 //Display:
34
35
36 printf("\n\nThe maximum normal stress at point P =
      %1.2f MPa', sigma_p);
37 printf("\nThe orientation of the neutral axis = %1
      .1f degrees', alpha);
38
39 //

```

END

Scilab code Exa 6.21 B21

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.21 : ")
4
5 //Given:
6 M = 2; //kNm
7 Ew = 12; //GPa
8 Est = 200; //GPa
9 bw = 150/1000; //m
10 t = 20/1000; //m
11 rib = 9/1000; //m
12
13 //Section Properties:
14 n = (Ew/Est);
15 bst = n*bw;
16
17 y1 = t/2;

```

```

18 A1 = t*bw;
19 y2 = bw/2 + t;
20 A2 = rib*bw;
21
22 y_bar = (y1*A1 + y2*A2)/(A1+A2);
23
24 I1 = (1/12)*(bw)*(t^3) + A1*(y_bar - y1)^2;
25 I2 = (1/12)*(rib)*(bw^3) + A2*(y2-y_bar)^2;
26 Ina = I1+I2;
27
28 //Normal Stress:
29 sigma_B = (M*(bw+t-y_bar))/(Ina*1000);
30 sigma_C = (M*(y_bar))/(Ina*1000);
31
32 //Normal Stress in the wood:
33 sigmaB = n*sigma_B;
34
35 //Display:
36
37
38 printf("\n\nThe normal stress at point B = %1.1f
      MPa', sigma_B);
39 printf("\nThe normal stress at point C = %1.2f MPa
      ', sigma_C);
40 printf("\nThe normal stress at point B in the wood
      = %1.2f MPa', sigmaB);
41
42 //-----
      END

```

Scilab code Exa 6.22 B22

```

1 clear all; clc;

```

```

2
3 disp(" Scilab Code Ex 6.22 : ")
4
5 // Given:
6 sigma_allow_st = 168; //MPa
7 sigma_allow_w = 21; //MPa
8 Est = 200; //GPa
9 Ew = 12; //GPa
10 Iz = 7.93*10^6; //mm^4
11 A1 = 5493.75; //mm^2
12 t = 5; //mm
13 h = 100; //mm
14
15 //Without Board:
16 c = h+t;
17 M1 = (sigma_allow_st*Iz)/(c*10^6);
18
19 //With Board:
20 bw = 300; //mm
21 n = (Ew/Est);
22 bst = n*bw;
23
24 //For the transformed section:
25 y1 = 0;
26 y2 = 55;
27 A2 = bst*h;
28
29 y_bar = (y1*A1 + y2*A2)/(A1+A2);
30
31 I1 = Iz + A1*y_bar^2;
32 I2 = (1/12)*(bst*h^3) + (A2*(y2-y_bar)^2);
33 I = I1+I2;
34
35 c = c+y_bar;
36 M2 = (sigma_allow_st*I)/(c*10^6);
37
38 cw = c - y_bar;
39 Mw = (sigma_allow_w*I)/(n*cw*10^6);

```

```

40
41 M = min(Mw,M2);
42
43 //Display:
44
45 printf("\n\nThe maximum bending moment without re-
      inforcement = %1.3f kNm',M1);
46 printf("\n\nThe maximum bending moment with re-
      inforcement      = %1.2f kNm',M);
47
48 //

```

END

Scilab code Exa 6.23 B23

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.23 : ")
4
5 //Given:
6 M = 60; //kNm
7 Est = 200; //GPa
8 Econc = 25; //GPa
9 d = 25; //mm
10 r = d/2;
11 w = 300; //mm
12 ht =400; //mm
13
14 //Section Properties:
15 n = Est/Econc;
16 Ast = 2*%pi*r^2;
17 A = n*Ast;

```

```

18
19 p = [1 52.37 -20949.33]
20 h = roots(p)
21 h = h(2);
22
23 I = (1/12)*(w*h^3) +w*h*(h/2)^2 + A*(ht - h)^2;
24
25 //Normal Stress:
26 sigma_conc_max = (M*1000*h*1000)/(I);
27 sigma_conc = (M*1000*(ht-h)*1000)/(I);
28 sigma_st = n*sigma_conc;
29
30 //Display:
31
32
33 printf("\n\nThe normal stress in each steel
      reinforcing rod = %1.2f MPa',sigma_st);
34 printf("\nThe maximum normal stress in the concrete
      = %1.2f MPa',sigma_conc_max);
35 //

```

END

Scilab code Exa 6.24 B24

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.24 : ")
4
5 //Given:
6 sigma = 140; //Mpa
7 ri = 90; //mm
8 ro = 110; //mm

```

```

9 a = 20; //mm
10
11 //Section Properties:
12
13 y = integrate('a*(1/r)', 'r', ri, ro)
14 R = (a*a)/y;
15
16 r_avg = (ri+ro)/2;
17 M1 = (-sigma*a*a*ro*(r_avg - R))/(R-ro);
18 M1 = M1*10^-6;
19
20 M2 = (sigma*a*a*ri*(r_avg - R))/(R-ri);
21 M2 = M2*10^-6;
22
23 M = min(M1, M2);
24
25 sigma1 = (M*(R - ro))/(a*a*ro*(r_avg - R));
26
27 //For a straight Bar:
28 I = (1/12)*(a*a^3);
29 c = 10; //mm
30 M_strt= (sigma*I)/c;
31 M_strt = M_strt*10^-6;
32
33 //Display:
34
35 printf("\n\nThe maximum bending moment that can be
    applied to the bar = %1.3f kNm', M);
36 printf("\n\nThe maximum bending moment that can be
    applied to a straight bar = %1.3f kNm', M_strt);
37 //

```

END

Scilab code Exa 6.25 B25

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 6.25 : ")
4
5 //Given:
6 ri = 200/1000; //m
7 r1 = 250/1000; //m
8 ro = 280/1000; //m
9 M = 4; //kNm
10 a = 0.05; //m
11 h = 0.03; //m
12
13 //Section Properties:
14 A1 = a^2 ;
15 A2 = (0.5*a*h);
16 A = A1+A2;
17 r_avg1 = (r1+ri)/2;
18 r_avg2 = r1+(h/3);
19 r_bar =((r_avg1*A1)+(r_avg2*A2))/A;
20
21 int_dA_r1 = a*log(r1/ri);
22 int_dA_r2 = (a*ro*log(ro/r1))/(ro-r1) - a;
23 R = (A)/(int_dA_r1+ int_dA_r2);
24 k= r_bar - R;
25
26 //Normal Stress:
27 sigma_B = (-M*(R-ri))/(A*ri*k*1000);
28 sigma_A = (-M*(R-ro))/(A*ro*k*1000);
29
30 sigma = max(abs(sigma_B),abs(sigma_A))
31
32
```

```

33 //Display :
34
35 printf("\n\nThe maximum normal stress in the bar =
        %1.0f MPa', sigma);
36
37 //-----
        END

```

Scilab code Exa 6.26 B26

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.26 : ")
4
5 //Given :
6 M = 5; //kNm
7 sigma_y = 500; //MPa
8 r = 16; //mm
9 h = 80; //mm
10 w = 120; //mm
11 r_h = r/h;
12 w_h = w/h;
13 k = 1.45;
14 c = h/(2000);
15 t = 20/1000; //m
16
17 //Calculations :
18 I = (1/12)*(t)*(h/1000)^3
19 sigma_max = (k*M*c)/(I*1000);
20
21 //Display :
22
23 printf("\n\nThe maximum normal stress in the steel

```

```

    = %1.0 f MPa', sigma_max);
24
25 //-----
    END
-----

```

Scilab code Exa 6.27 B27

```

1  clear all; clc;
2
3  disp("Scilab Code Ex 6.27 : ")
4
5  //Given:
6  sigma_y = 250; //MPa
7  t = 12.5; //mm
8  w = 200; //mm
9  h = 225; //mm
10
11 //Maximum Elastic Moment:
12 yy = (h+t)/2;
13 I1 = (1/12)*(w*t^3) + (w*t*yy^2);
14 I = (1/12)*(t*h^3) + 2*(I1);
15 c = 125; //mm
16
17 My = (sigma_y*I)/(c); //Flexure Formula
18
19 //Plastic Moment:
20 C1= sigma_y*t*(h/2);
21 C2= sigma_y*t*(w);
22 Mp = (2*56.25*C1) + (2*yy*C2);
23
24 //Shape Factor:
25 k = Mp/My;
26

```

```

27 // Display :
28
29
30 printf("\n\nThe shape factor for the beam = %1.2 f
      ',k);
31
32 //-----
      END
      -----

```

Scilab code Exa 6.28 B28

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.28 : ")
4
5 // Given:
6 sigma_y = 250; //MPa
7 t = 15/1000; //m
8 w = 100/1000; //m
9 h = 120/1000; //m
10 c = 10/1000; //m
11
12 // Calculations:
13 d = ((sigma_y*t*w)+(sigma_y*t*h))/(sigma_y*t*2);
14
15 T = sigma_y*t*d*10^3;
16 C1 = sigma_y*t*c*10^3;
17 C2 = sigma_y*t*w*10^3;
18
19 Mp = (T*d/2)+(C1*c/2)+(C2*(c+t/2));
20
21 // Display :
22

```

```

23
24 printf("\n\nThe plastic moment that can be resisted
      by the beam = %1.1f kNm',Mp);
25
26 //-----
      END
      -----

```

Scilab code Exa 6.29 B29

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 6.29 : ")
4
5 //Given:
6 ep1 = 0.01;
7 ep2 = 0.05;
8 sig1 = 1050; //N/mm^2
9 sig2 = 1330; //N/mm^2
10 sig3 = 280; //N/mm^2
11 y = 0.3; //cm
12 h = 3; //cm
13 w = 2; //cm
14
15 //Calculations:
16 yy = (h/2)-y
17 T1 = (1/2)*(sig3*yy*w);
18 y1 = y +(2/3)*(yy);
19 T2 = yy*sig1*w;
20 y2 = y+(0.5*yy);
21 T3 = (0.5*y*sig1*w);
22 y3 = (2/3)*(y);
23
24 M = 2*(T1*y1 + T2*y2 + T3*y3);

```

```

25 M = M/1000;
26
27 //Display:
28
29
30 printf("\n\nThe bending moment applied that will
      cause a strain of 0.05mm/mm = %1.2 f kNm',M);
31
32 //-----
      END
      -----

```

Scilab code Exa 6.30 B30

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 6.30 : ")
4
5 //Given:
6 sigma_y = 250; //MPa
7 t = 12.5; //mm
8 w = 200; //mm
9 h = 225; //mm
10 c = (h/2)+t;
11 I = 82.44*10^6; //mm^4
12 Mp = 188; //kN
13
14 //Calculations:
15 sigma_allow = (Mp*10^6*c)/(I);
16 y = (sigma_y*c)/(sigma_allow);
17
18 //Display:
19
20 printf("\n\nThe point of zero normal stress = %1.2 f

```

```
    mm',y);
21  printf("\nThe Residual Stress distribution is shown
      in the text book.");
22
23  //-----
      END
      -----
-----
```

Chapter 7

Transverse Shear

Scilab code Exa 7.1 TS1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 7.1 : ")
4
5 //Given:
6 V = 3; //kN
7 h = 125; //mm
8 b = 100; //mm
9 y_top = 50; //mm
10 x_right = 37.5; //mm
11
12 //Part (a):
13
14 //Section Properties:
15 I = (b*h^3)/12;
16 y_dash_1 = ((h-y_top)-(h/2));
17 A = y_top*b;
18 Q = (y_dash_1+(y_top/2))*A;
19
20 //Shear Stress:
21 tou_p = (V*Q)/(I*b); //tou = VQ/It
```



```

22  tou_p = tou_p*10^3;
23
24  //Part (b):
25
26  //Section Properties:
27  y_dash_2 = (y_dash_1+(y_top));
28  a_dash= b*y_dash_2;
29  Q_dash =(y_dash_2*a_dash)/2;
30
31  //Shear Stress:
32  tou_max = (V*Q_dash)/(I*b);
33  tou_max = tou_max*10^3;
34
35  //Display:
36
37  printf("\n\nThe shear stress in the beam at point P
          = %1.3 f MPa', tou_p);
38  printf('\n\nThe maximum shear stress in the beam
          = %1.3 f MPa', tou_max);
39
40  //-----
    END
    -----

```

Scilab code Exa 7.2 TS2

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 7.2 : ")
4
5  //Given:
6  V = 80; //kN
7  thick_1 = 20/1000; //m
8  thick_2 = 15/1000; //m

```

```

 9 l = 300/1000; //m
10 y = 100/1000; //m
11 h = 2*y;
12 y_dash = y +thick_1/2;
13
14 //Part(a):
15
16 I1 = (thick_2*(h^3))/12;
17 I2 = (l*(thick_1^3))/12;
18 I3 = (l*thick_1*(y_dash)^2);
19 I = I1+2*(I2+I3); //Moment of inertia
20
21 Q_b = y_dash*l*thick_1;
22 //At B'
23 tou_b_dash = (V*Q_b)/(I*l*1000);
24 //At B
25 tou_b = (V*Q_b)/(I*thick_2*1000);
26
27 //At C:
28 Q_c = (y_dash*l*thick_1)+(y*thick_2*y/2);
29 tou_c = (V*Q_c)/(I*thick_2*1000);
30
31 //Part(b)
32
33
34 y0 = -0.1;
35 y1 = 0.1;
36
37 function Q =f(y),Q = ((0.735 - (7.5*y*y))*10^-3),
38 endfunction
39 Int =intg(y0,y1,f)
40
41 V_w = (V*Int*thick_2)/(I*thick_2);
42
43 //Display:
44
45 printf("\n\nThe shear stress at B dash      =
      %1.2 f MPa', tou_b_dash);

```

```

46 printf("\nThe shear stress at B           =
      %1.1f MPa', tou_b);
47 printf("\nThe shear stress at C           =
      %1.1f MPa', tou_c);
48 printf("\nThe shear force resisted by the web =
      %1.1f kN', V_w);
49
50 //

```

END

Scilab code Exa 7.3 TS3

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 7.3 : ")
4
5 //Given:
6 udl = 6.5; //kN
7 l_bc = 8; //m
8 l = 150/1000; //m
9 t = 30/1000; //m
10
11 //Internal Shear:
12 w = udl*l_bc/2;
13 l_wc = l_bc/4;
14 l_bw = l_bc - l_wc;
15 V = (w*l_bw)/l_bc;
16 R_b = w - V;
17
18 //Section Properties:
19 y1 = l/2;
20 A = (l*t);

```

```

21 y2= l+(t/2);
22 y_dash = (y1*A + y2*A)/(2*A);
23 I1 = (t*l^3)/12;
24 I2 = (A*(y_dash-y1)^2);
25 I3 = (l*t^3)/12;
26 I4 = (A*(y2 - y_dash)^2);
27 I = I1+I2+I3+I4;
28
29 Q = ((l+t)-(t/2)-y_dash)*A;
30
31 //Shear Stress:
32 tou_max = (V*Q)/(I*t*1000);
33
34 //Display:
35
36 printf("\n\nThe maximum shear stress in the glue
        necessary to hold the boards together    = %1.2 f
        MPa', tou_max);
37
38
39 //-----
        END
        -----

```

Scilab code Exa 7.4 TS4

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 7.4 : ")
4
5 //Given:
6
7 V = 850; //kN
8 l1 =250/1000; //m

```

```

9  l2 = 300/1000; //m
10 l3 = 125/1000; //m
11 t = 10/1000; //m
12 h = 200/1000; //m
13
14 A1 = l1*t;
15 A2 = l2*t;
16 A3 = l3*t;
17
18 y1 = l2+(t/2);
19 y2 = l2/2;
20 y3 = h+(t/2);
21
22 y_dash = (2*y2*A2 + A1*y1 + A3*y3)/(2*A2 + A1 + A3);
23
24 I1 = ((l1*t^3)/12) +(A1 * (l2+(t/2)-y_dash)^2);
25 I2 = ((t*l2^3)/12) +(A2 * (y_dash - (l2/2))^2);
26 I3 = ((l3*t^3)/12) +(A1 * (h+(t/2)-y_dash)^2);
27 I = 2*I2 + I1 + I3;
28
29 Q_b = (l2+(t/2) - y_dash)*A1; //Q = y'A'
30 Q_c = (h+(t/2) - y_dash)*A3; //Q = y'A'
31
32 //Shear Flow:
33
34 q_b = (V*Q_b)/I;
35 q_c = (V*Q_c)/I;
36
37 q_b = q_b/(2*1000);
38 q_c = q_c/(2*1000);
39
40 //Display:
41
42 printf("\n\nThe shear flow at B, resisted by the
      glue is      = %1.2f MN/m',q_b);
43 printf("\n\nThe shear flow at C, resisted by the glue
      is      = %1.4f MN/m',q_c);
44

```

45
46
47
48 //

END

Scilab code Exa 7.5 TS5

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 7.5 : ")
4
5 //Given:
6 V = 80; //N
7 t = 1.5; //cm
8 a = 7.5; //cm
9 b = a-2*t; //cm
10 F_nail= 30; //N
11
12 //Section Properties:
13 I = (a*a^3 - b*b^3 )/12;
14 Q_b = (((a-2*t)/2)+(t/2))*a*t; //Q = y'A'
15 Q_c = (((a-2*t)/2)+(t/2))*(a-2*t)*t; //Q = y'A'
16
17 //Shear Flow:
18 q_b = (V*Q_b)/I;
19 q_c = (V*Q_c)/I;
20
21 s_b = F_nail/(q_b/2);
22 s_c = F_nail/(q_c/2);
23
24 //Display :
```

```

25
26
27 printf("\n\nThe maximum spacing of nails required at
      B is      = %1.0f cm',s_b);
28 printf("\n\nThe maximum spacing of nails required at C
      is      = %1.1f cm',s_c);
29
30
31
32 //

```

END

Scilab code Exa 7.6 TS6

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 7.6 : ")
4
5 //Given:
6 F = 40; //N
7 s = 9; //cm
8 h = 5; //cm
9 t = 0.5; //cm
10 w = 3; //cm
11 w_3 = w/3; //cm
12
13 // Calculations:
14
15 I = (w*h^3)/12 - (2*w_3*(h - 2*t)^3)/12;
16
17 //Case 1:
18

```

```

19 Q1 = ((h-t)/2)*(w*t);
20 V1 =((F/s)*I)/Q1 ; //q = VQ/I
21
22 //Case2:
23
24 Q2 = ((h-t)/2)*(w_3*t);
25 V2 =((F/s)*I)/Q2 ; //q = VQ/I
26
27 //Display:
28
29
30 printf("\n\nThe largest vertical shear that can be
    supported in Case 1    = %1.1f N',V1);
31 printf("\n\nThe largest vertical shear that can be
    supported in Case 2    = %1.1f N',V2);
32
33 //

```

END

Scilab code Exa 7.7 TS7

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 7.7 : ")
4
5 //Given:
6 V = 10; //kN
7 b1 = 6; //cm
8 h1 = 8; //cm
9 t = 1; //cm
10 b2 = b1-2*t;
11 h2 = h1-2*t; //cm

```



```

12 b3 = 4; //cm
13
14 // Calculations:
15 I = ((b1*h1^3)/12) - ((b2*h2^3)/12);
16
17 q_b = 0;
18
19 Q_c = ((b1/2)+(t/2))*(b3+(t))*t;
20 q_c = (V*Q_c*100)/(I); //Q = VQ/I
21
22 Q_d = (2*h1/4*t*b3) + ((b1/2)+(t/2))*b3*t;
23 q_d = (V*Q_d*100)/(I); //Q = VQ/I
24
25 //Display:
26
27
28 printf("\n\nVariation of shear flow at B = %1.1f N/
      mm', q_b);
29 printf("\nVariation of shear flow at C = %1.1f N/mm
      ', q_c);
30 printf('\nVariation of shear flow at D = %1.1f N/mm
      ', q_d);
31
32 //

```

END

Chapter 8

Combined Loadings

Scilab code Exa 8.1 CL1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 8.1 : ")
4
5 //Given:
6 di = 1.2*1000; //m
7 ri = di/2;
8 t = 12; //mm
9 sigma = 140; //MPa
10
11 //Cylindrical Pressure Vessel:
12
13 p1 = (t*sigma)/ri; //sigma = pr/t
14
15 //Spherical Vessel:
16
17 p2 = (2*t*sigma)/(ri); //sigma = pr/2t
18
19 //Display:
20
21 printf("\n\nThe maximum internal pressure the
```

```

        cylindrical pressure vessel can sustain      = %1.1
        f N/mm2',p1);
22 printf('\nThe maximum internal pressure a spherical
        pressure vessel can sustain      = %1.1 f N/mm
        ^2',p2);
23
24 //-----
        END
-----

```

Scilab code Exa 8.2 CL2

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 8.2 : ")
4
5  // Given:
6  P = 15000; //N
7  a = 40; //mm
8  b = 100; //mm
9
10 // Stress Components:
11
12 // Normal Force:
13 A = a*b;
14 sigma = P/A;
15
16 // Bending Moment:
17 I = (a*b3)/12; // I = (1/12)*bh3
18 M = P*(b/2);(b/2);
19 c = b/2;
20 sigma_max =(M*c)/I;
21
22 // Superposition:

```

```

23 x = ((sigma_max-sigma)*b)/((sigma_max+sigma)+(
      sigma_max-sigma));
24 sigma_b = (sigma_max-sigma);
25 sigma_c = (sigma_max + sigma);
26
27 //Display:
28
29 printf("\n\nThe state of stress at B    = %1.1f MPa
      (tensile)',sigma_b);
30 printf('\n\nThe state of stress at C    = %1.1f MPa (
      compressive)',sigma_c);
31
32 //-----
      END
      -----

```

Scilab code Exa 8.3 CL3

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 8.3 : ")
4
5  //Given:
6  ri = 600/1000; //m
7  t = 12/1000; //m
8  ro = ri+t;
9  sp_wt_water = 10; //kN/m^3
10 sp_wt_steel = 78; //kN/m^3
11 l_a = 1; //m depth of point A from the top
12
13 //Internal Loadings:
14 v = (%pi*l_a)*(ro^2 - ri^2);
15 W_st = sp_wt_steel*v;
16

```

```

17 p = sp_wt_water*l_a; //Pascal's Law
18
19 //Stress Components:
20
21 //Circumferential Stress:
22 sigma1 = (p*ri)/t;
23
24 //Longitudinal Stress:
25 A_st = (%pi)*(ro^2 - ri^2);
26 sigma2 = W_st/A_st;
27
28 //Display:
29
30
31 printf("\n\nThe state of stress at A (
      Circumferential)      = %1.1f kPa',sigma1);
32 printf('\n\nThe state of stress at A (Longitudinal)
      = %1.1f kPa',sigma2);
33
34 //-----
      END
      -----

```

Scilab code Exa 8.4 CL4

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 8.4 : ")
4
5 //Given:
6 y_c = 125/1000; //m
7 x_c = 1.5; //m
8 y_b = 1.5; //m
9 x_b = 6; //m

```

```

10 udl = 50; //kN/m
11 l_udl = 2.5; //m
12 l = 250/1000; //m
13 width = 50/1000; //m
14
15
16 //Internal Loadings:
17 N = 16.45; //kN
18 V = 21.93; //kN
19 M = 32.89; //kNm
20
21 //Stress Components:
22
23 //Normal Force:
24 A = l*width;
25 sigma1 = N/(A*1000);
26
27 //Shear Force:
28 tou_c = 0;
29
30 //Bending Moment:
31 c = y_c;
32 I = (1/12)*(width*l^3);
33 sigma2 = (M*c)/(I*1000);
34
35 //Superposition:
36 sigmaC = sigma1+sigma2;
37
38 //Display:
39
40
41 printf('\n\nThe stress due to normal force at C
    = %1.2f MPa',sigma1);
42 printf('\n\nThe stress due to shear force at C      =
    %1.2f MPa',tou_c);
43 printf('\n\nThe stress due to bending moment at C   =
    %1.2f MPa',sigma2);
44 printf('\n\nThe resultant stress at C              =

```

```
    %1.1 f MPa',sigmaC);
45
46 //

```

END

Scilab code Exa 8.5 CL5

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 8.5 : ")
4
5 // Given:
6 r = 0.75*10; //mm
7 f_x =500; //N
8 f_y =800; //N
9 l1 = 8*10; //mm
10 l2 = 10*10; //mm
11 l3 = 14*10; //mm
12
13 // Stress Components:
14
15 // Normal Force:
16 A1 = (%pi*r^2);
17 sigma1 = f_x/A1; //stress = P/A
18
19 // Shear Force:
20 y_bar = (4*r)/(3*%pi);
21 A2 = A1/2;
22 Q = y_bar*A2; //Q = yA
23 V = f_y;
24 I = (1/4)*(%pi*r^4);
25 t = 2*r;
```

```

26  tou_a = (V*Q)/(I*t); //Shear = VQ/It
27
28  //Bending Moment:
29  M_y = f_x*l3;
30  c = r;
31  sigma_A = (M_y*c)/I;
32
33  //Torsional Moment:
34  T = f_y*l3;
35  J = (0.5*pi*r^4);
36  tou_A = (T*c)/J;
37
38  //Resultant:
39  res_normal= sigma1+sigma_A;
40  res_shear = tou_a+tou_A;
41
42  //Display:
43
44  printf('\n\nThe stress due to normal force at A
          = %1.2f MPa',sigma1);
45  printf('\n\nThe stress due to shear force at A
          = %1.2f MPa',tou_a);
46  printf('\n\nThe stress due to bending moment at A
          = %1.2f MPa',sigma_A);
47  printf('\n\nThe stress due to torsional moment at A
          = %1.2f MPa',tou_A);
48  printf('\n\nThe resultant normal stress component at A
          = %1.2f MPa',res_normal);
49  printf('\n\nThe resultant shear stress component at A
          = %1.2f MPa',res_shear);
50
51  //

```

END

Scilab code Exa 8.6 CL6

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 8.6 : ")
4
5 //Given:
6 P = 40; //kN
7 l_ab = 0.4; //m
8 l_bc = 0.8; //m
9
10 //Stress Components:
11
12 //Normal Force:
13 A = l_ab*l_bc;
14 sigma = P/A;
15
16 //Bendng Moments:
17 M_x = P*l_ab/2;
18 cy = l_ab/2;
19 Ix = (1/12)*(l_bc*l_ab^3); //I = (1/12)*(bh^3)
20 sigma_max_1 = (M_x*cy)/Ix; //sigma = My/I
21
22 M_y = P*l_bc/2;
23 cx = l_bc/2;
24 Iy = (1/12)*(l_ab*l_bc^3); //I = (1/12)*(bh^3)
25 sigma_max_2 = (M_y*cx)/Iy; //sigma = My/I
26
27 //Superposition:
28 stress_A = -sigma + sigma_max_1 + sigma_max_2;
29 stress_B = -sigma - sigma_max_1 + sigma_max_2;
30 stress_C = -sigma - sigma_max_1 - sigma_max_2;
31 stress_D = -sigma + sigma_max_1 - sigma_max_2;
32
```

```

33 e = abs((stress_B*l_ab)/(stress_A-stress_B));
34 h = abs((stress_B*l_bc)/(stress_A-stress_B));
35
36 //Display:
37
38
39 printf('\n\nThe normal stress at corner A
      = %1.0f kPa',stress_A);
40 printf('\nThe normal stress at corner B           =
      %1.0f kPa',stress_B);
41 printf('\nThe normal stress at corner C           =
      %1.0f kPa',stress_C);
42 printf('\nThe normal stress at corner D           =
      %1.0f kPa',stress_D);
43 printf('\nThe line of zero stress along AB         =
      %1.4f m',e);
44 printf('\nThe line of zero stress along AD         =
      %1.3f m',h);
45
46 //

```

END

Chapter 9

Stress Transformation

Scilab code Exa 9.1 ST1

```
1
2 clear all; clc;
3
4 disp(" Scilab Code Ex 9.1 : ")
5
6 // Given:
7 tau = 25; //MPa
8 sigma1 = 50; //MPa
9 sigma2 = 80; //MPa
10 phi = 30*(%pi/180);
11
12 // Calculations:
13 sigma_x1 = (sigma1*cos(phi)*cos(phi))- (tau*cos(phi)
    *sin(phi)) - (sigma2*sin(phi)*sin(phi))- (tau*sin
    (phi)*cos(phi));
14 tau1 = (sigma1*cos(phi)*sin(phi))+ (tau*cos(phi)*cos
    (phi)) + (sigma2*sin(phi)*cos(phi))- (tau*sin(phi)
    )*sin(phi));
15 sigma_x2 = (tau*cos(phi)*sin(phi))- (sigma2*cos(phi)
    *cos(phi)) + (tau*sin(phi)*cos(phi))+ (sigma1*sin
    (phi)*sin(phi));
```

```

16 tou2 = (tou*cos(phi)*cos(phi))+ (sigma2*cos(phi)*sin
    (phi)) - (tou*sin(phi)*sin(phi))+ (sigma1*sin(phi)
    )*cos(phi));
17
18 //Display:
19
20 printf("\n\nThe normal stress component in the x
    diection is      = %1.2f MPa',sigma_x1);
21 printf("\n  The shear stress component in the x
    diection is      = %1.1f MPa',tou1);
22 printf("\n  The normal stress component in the y
    diection is      = %1.1f MPa',sigma_x2);
23 printf("\n  The shear stress component in the y
    diection is      = %1.1f MPa',tou2);
24
25 //

```

END

Scilab code Exa 9.2 ST2

```

1 clear all; clc;
2
3
4 disp(" Scilab Code Ex 9.2 : ")
5
6 //Given:
7 phi = -30*(%pi/180);
8 theta = 60*(%pi/180);
9 sigma_x = -80; //MPa
10 sigma_y = 50; //MPa
11 tou_xy = -25; //MPa
12

```

```

13 //Plane CD:
14 sigma_x1 = (sigma_x+sigma_y)/2 + ((sigma_x-sigma_y)*
    cos(2*phi))/2 + (tou_xy*sin(2*phi)); //Eqn 9.1
15 tou_xy1 = ((-(sigma_x - sigma_y)*sin(2*phi))/2) + (
    tou_xy*cos(2*phi)); //Eqn 9.2
16
17 //Plane BC:
18 sigma_x2 = (sigma_x+sigma_y)/2 + ((sigma_x-sigma_y)*
    cos(2*theta))/2 + (tou_xy*sin(2*theta)); //Eqn
    9.1
19 tou_xy2 = (-(sigma_x - sigma_y)*sin(2*theta))/2 +
    tou_xy*cos(2*theta); //Eqn 9.2
20
21 //Display:
22
23 printf('\n\nThe normal stress of plane CD inclined
    at 30 degrees = %1.1f MPa',sigma_x1);
24 printf('\nThe shear stress of plane CD inclined at
    30 degrees = %1.1f MPa',tou_xy1);
25 printf('\nThe normal stress of plane BC inclined at
    60 degrees = %1.2f MPa',sigma_x2);
26 printf('\nThe shear stress of plane BC inclined at
    60 degrees = %1.1f MPa',tou_xy2);
27
28 //

```

END

Scilab code Exa 9.5 ST5

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 9.5 : ")

```

```

4
5 //Given:
6 sigma_x = -20; //MPa
7 sigma_y = 90; //MPa
8 tou_xy = 60; //MPa
9
10 //Orientation of Element:
11 theta_p2 = atan((2*tou_xy)/(sigma_x - sigma_y));
12 theta_p2 = theta_p2/2;
13 theta_p1 = (180+2*theta_p2)/2;
14
15 //Principal Stresses:
16
17 sigma1 = ((sigma_x+sigma_y)/2)+(sqrt(((sigma_x -
18     sigma_y)/2)^2 + tou_xy^2));
19 sigma2 = ((sigma_x+sigma_y)/2)- sqrt(((sigma_x-
20     sigma_y)/2)^2 + tou_xy^2);
21 sigma_x2 = ((sigma_x+sigma_y)/2)+ (((sigma_x-sigma_y
22     )/2)*cos(2*theta_p2)) + (tou_xy*sin(2*theta_p2));
23
24 //Display:
25
26 printf("\n\nThe first principal stress is
27     = %1.0f MPa',sigma1);
28 printf("\n\nThe second principal stress is
29     = %1.1f MPa',sigma2);
30 printf('\n\nThe normal stress acting on the 23.7
31     degrees plane = %1.1f MPa',sigma_x2);
32
33 //

```

END

Scilab code Exa 9.6 ST6

```
1
2 clear all; clc;
3
4 disp(" Scilab Code Ex 9.6 : ")
5
6 //Given:
7 sigma_x = -20; //MPa
8 sigma_y = 90; //MPa
9 tau_xy =60; //Mpa
10
11 //Orientation of Element:
12 theta_s2 = atan(-(sigma_x - sigma_y)/(2*tau_xy));
13 theta_s2 = theta_s2/2;
14 theta_s1 = %pi + 2*theta_s2;
15 theta_s1 = theta_s1/2;
16
17 //Maximum in plane Shear Stress:
18 tau_max = (sqrt(((sigma_x - sigma_y)/2)^2 + tau_xy
19             ^2));
20
21 //Average Normal Stress:
22 sigma_avg = (sigma_x+sigma_y)/2;
23
24 //Display:
25
26 printf("\n\nThe maximum in-plane shear stress is
27         = %1.1f MPa',tau_xy1);
28
29 printf("\n\nThe average normal stress is
30         = %1.0f MPa',sigma_avg);
31
32 //
```

END

Scilab code Exa 9.9 ST9

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 9.9 : ")
4
5 //Given:
6 sigma_x = -12; //MPa
7 sigma_y = 0;
8 tou_xy = -6; //MPa
9
10 //Construction of the circle:
11 sigma_avg = (sigma_x+sigma_y)/2;
12 R = sqrt((-sigma_x+sigma_avg)^2 + (tou_xy)^2);
13
14 //Principal Stresses:
15 sigma2 = -R+sigma_avg;//From the Mohr's circle
16 sigma1 = R+sigma_avg;
17
18 theta_p2 = atan((-tou_xy)/(-sigma_x+sigma_avg));
19 theta_p2 = theta_p2/2*(180/%pi);
20
21 //Display:
22
23 printf('\n\nThe first principal stress is
           = %1.2f MPa',sigma1);
24 printf('\n\nThe second principal stress is
           = %1.1f MPa',sigma2);
25 printf('\n\nThe direction of the principal plane is
           = %1.1f degrees ',theta_p2);
26
27
28 //
```

END

Scilab code Exa 9.10 ST10

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 9.10 : ")
4
5 //Given:
6 sigma_x = -20; //MPa
7 sigma_y = 90; //MPa
8 tou_xy = 60; //MPa
9
10 //Construction of the circle:
11 sigma_avg = (sigma_x+sigma_y)/2;
12 R = sqrt(((sigma_x-sigma_avg))^2 + (tou_xy)^2);
13
14 //Maximum In plane Shear Stress:
15 tou_max = R;
16
17 theta_s1 = atan(-(sigma_x - sigma_avg)/(tou_xy));
18 theta_s1 = theta_s1/2*(180/%pi);
19
20 //Display:
21
22 printf('\n\nThe maximum in-plane shear stresses are
        = %1.1f MPa', tou_max);
23 printf('\n
        = %1.1f MPa', sigma_avg);
24 printf('\n\nThe orientation of the element is
        = %1.1f degrees', theta_s1);
25
```

26 //

END

Scilab code Exa 9.11 ST11

```
1
2 clear all; clc;
3
4 disp("Scilab Code Ex 9.11 : ")
5
6 //Given:
7 sigma_x = -8; //MPa
8 sigma_y = 12; //MPa
9 tou_xy = -6; //Mpa
10
11 //Construction of the circle:
12 sigma_avg = (sigma_x+sigma_y)/2;
13
14 R = sqrt( 10^2 + tou_xy^2);
15
16 //Stresses on 30 degree element:
17 phi = atan(6/10);
18 psi = (%pi/3) - phi;
19
20 //On face BD:
21 sigma_x1 = 2 - (R*cos(psi));
22 tou_xy1 = (R*sin(psi));
23
24 //On face DE:
25 sigma_x2 = 2 + (R*cos(psi));
26 tou_xy2 = -(R*sin(psi));
27
```

```

28 //Display:
29
30 printf('\n\nThe normal stress on plane BD inclined
      at 30 degrees is = %1.2f MPa',sigma_x1);
31 printf('\n\nThe normal stress on plane DE inclined at
      60 degrees is = %1.1f MPa',sigma_x2);
32 printf('\n\nThe shear stress is
      = %1.2f
      MPa',tou_xy1);
33
34
35 //

```

END

Scilab code Exa 9.12 ST12

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 9.12 : ")
4
5 //Given:
6 P = 900; //N
7 T = 2.5; //Nm
8 d = 40/1000; //m
9 r = d/2;
10 c = r;
11
12 //Stress Components:
13 J = (%pi/2)*(r^4);
14 tou = (T*c)/(J*1000);
15
16 A = (%pi*r^2);

```

```

17 sigma = P/(A*1000);
18
19 //Principal Stresses:
20 sigma_avg = (0 + sigma)/2;
21
22 R = sqrt( sigma_avg^2 + tou^2);
23 sigma1 = sigma_avg + R;
24 sigma2 = sigma_avg - R;
25
26 //Display:
27
28 printf('\n\nThe prinicipal stresses at point P are:')
    ;
29 printf('\n %1.1f  kPa',sigma1);
30 printf('\n %1.1f  kPa',sigma2);
31
32 //

```

END

Scilab code Exa 9.13 ST13

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 9.13 : ")
4
5 //Given:
6 w = 120; //kN/m
7 I = 67.4*(10^-6);
8 V= 84; //kN
9 M = 30.6; //kNm
10 t = 10/1000; //m
11

```

```

12 //Stress Components:
13 y = 0.200/2;
14 sigma = -(M*10^3*y)/(I*10^6);
15
16 Q = (0.100 + 0.015/2)*(0.175)*(0.015);
17 tou = (V*Q*10^3)/(I*t*10^6);
18
19 //Principal Stresses:
20
21 k = sigma/2;
22 R = sqrt((-sigma+k)^2 + tou^2);
23 sigma1 = R + k;
24 sigma2 = k -R ;
25
26 theta_p2 = atan(-tou/(sigma-k));
27 theta_p2 =theta_p2/2*(180/%pi);
28
29 //Display:
30
31
32 printf('\n\nThe prinicpal stresses at point P are:')
    ;
33 printf('\n %1.1f MPa',sigma1);
34 printf('\n %1.1f MPa',sigma2);
35 printf('\nThe angle of rotation of the plane %1.1f
    degrees ',theta_p2);
36
37 //

```

END

Scilab code Exa 9.14 ST14

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 9.14 : ")
4
5 //Given:
6 tau = 40; //kPa
7 sigma = -20; //kPa
8
9 //Principal Stresses:
10 sigma_avg = sigma/2;
11 R = sqrt((-sigma + sigma_avg)^2 + tau^2);
12 sigma_max = sigma_avg + R ;
13 sigma_min = sigma_avg - R ;
14
15 theta = atan(tau/(-sigma+sigma_avg));
16 theta = theta/2;
17
18 //Absolute Maximum Shear Stress:
19 tau_max = (sigma_max - sigma_min)/2;
20 sigma_avg = (sigma_max + sigma_min)/2;
21
22 //Display:
23
24 printf('\n\nThe principal stresses at the point are:
25         ');
26 printf('\n %1.1f kPa',sigma_max);
27 printf('\n %1.1f kPa',sigma_min);
28 printf('\nThe absolute maximum shear stress at the
29         point %1.1f kPa',tau_max);
30
31 //

```

END

Scilab code Exa 9.15 ST15

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 9.15 : ")
4
5 //Given:
6 sigma_max = 32; //MPa
7 sigma_min = 0; //MPa
8 sigma_int = 16; //MPa
9
10 tau_max = (sigma_max - sigma_min)/2 ; //MPa
11 sigma_avg = (sigma_max + sigma_min)/2 ; //MPa
12
13 tau_in_plane = (sigma_max - sigma_int)/2;
14 sigma_avg2 = sigma_avg + (tau_in_plane);
15
16 //Display:
17
18 printf('\n\nThe maximum absolute shear stress = %1
19         .2f MPa',tau_max);
20 //
```

END

Chapter 10

Strain Transformation

Scilab code Exa 10.1 StnT1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.1 : ")
4
5 //Given:
6 ep_x = 500; //Normal Strain
7 ep_y = -300; //Normal Strain
8 gamma_xy = 200; //Shear Strain
9 theta = 30*(%pi/180);
10 theta = theta*-1;
11
12 ep_x_new = ((ep_x+ep_y)/2) + ((ep_x - ep_y)/2)*cos
    (2*theta) + (gamma_xy/2)*sin(2*theta);
13
14 gamma_xy_new = -((ep_x - ep_y)/2)*sin(2*theta) + (
    gamma_xy/2)*cos(2*theta);
15 gamma_xy_new = 2*gamma_xy_new;
16
17 phi = 60*(%pi/180);
18 ep_y_new = (ep_x+ep_y)/2 + ((ep_x - ep_y)/2)*cos(2*
    phi) + (gamma_xy/2)*sin(2*phi);
```



```

19
20 //Display:
21
22
23 printf('\n\nThe equivalent strain acting on the
      element in the x plain oriented at 30 degrees
      clockwise      = %1.1f *10-6', ep_x_new);
24 printf('\n\nThe equivalent strain acting on the
      element in the y plain oriented at 30 degrees
      clockwise      = %1.1f *10-6', ep_y_new);
25 printf('\n\nThe equivalent shear strain acting on the
      element
                                                                 = %1
      .0f *10-6', gamma_xy_new);
26
27 //

```

END

Scilab code Exa 10.2 StnT2

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 10.2 : ")
4
5 //Given:
6 ep_x = -350; //(*10-6) Normal Strain
7 ep_y = 200; //*(10-6) Normal Strain
8 gamma_xy = 80; //*(10-6) Shear Strain
9
10 //Orientation of the element:
11 tan_thetap = (gamma_xy)/(ep_x - ep_y);
12 thetap1 = (0.5)*(atan(tan_thetap));

```

```

13
14 //Principal Strains:
15
16 k = (ep_x + ep_y)/2;
17 l = (ep_x - ep_y)/2;
18 tou = gamma_xy/2;
19 R = sqrt( (l)^2 + tou^2);
20 ep1 = R + k;
21 ep2 = k -R ;
22 ep = [ep1 ep2];
23
24 ep_x1 = k + l*cos(2*thetap1)+ tou*sin(2*thetap1);
25 thetap1 = thetap1*(180/%pi);
26 thetap2 = (90 + thetap1);
27 thetap =[thetap1 thetap2];
28
29
30 //Display:
31
32 printf('\n\nThe orientation of the element in the
    positive counterclockwise direction = %1.2f
    degrees, %1.2f degrees ',thetap);
33 printf('\nThe principal strains are
    = %1.0f *10^-6 , %1.0f *10^-6 ',ep);
34 printf('\nThe principal strain in the new x
    direction is = %1.0
    f *10^-6 ',ep_x1);
35
36 //

```

END

Scilab code Exa 10.3 StnT3

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 10.3 : ")
4
5 //Given:
6 ep_x = -350; //(*10^-6) Normal Strain
7 ep_y = 200; //*(10^-6) Normal Strain
8 gamma_xy = 80; //*(10^-6) Shear Strain
9
10 //Orientation of the element:
11 tan_thetap = -(ep_x - ep_y)/(gamma_xy);
12 thetap1 = (0.5)*(atan(tan_thetap));
13
14 //Maximum in-plane shear strain:
15
16 l = (ep_x - ep_y)/2;
17 tou = gamma_xy/2;
18 R = sqrt(l^2 + tou^2);
19 max_inplane_strain = 2*R;
20
21 gamma_xy_1 = (-1*sin(2*thetap1)+ tou*cos(2*thetap1))
    *2;
22 strain_avg = (ep_x + ep_y)/2;
23
24 thetap1 = thetap1*(180/%pi);
25 thetap2 = (90 + thetap1);
26 thetap = [thetap1 thetap2];
27
28 //Display:
29
30 printf('\n\nThe orientation of the element
    = %1.1f degrees , %1.1f degrees ',
    thetap);
31 printf('\n\nThe maximum in-plane shear strain
    = %1.0f *10^-6 ',max_inplane_strain);
32 printf('\n\nThe average strain
```

```

    = %1.0f *10^-6 ', strain_avg);
33
34 //

```

END

Scilab code Exa 10.4 StnT4

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.4 : ")
4
5 //Given:
6 ep_x = 250; //(*10^-6) Normal Strain
7 ep_y = -150; //*(10^-6) Normal Strain
8 gamma_xy = 120; //*(10^-6) Shear Strain
9
10 //Construction of the circle:
11 strain_avg = (ep_x + ep_y)/2;
12 tou = gamma_xy/2;
13 R = sqrt((ep_x - strain_avg)^2 + (tou^2));
14
15 //Principal Strains:
16 ep1 = (strain_avg + R);
17 ep2 = (strain_avg - R);
18 strain = [ep1 ep2];
19
20 tan_thetap = (tou)/(ep_x - strain_avg);
21 thetap1 = (atan(tan_thetap))/2;
22 thetap1 = thetap1*(180/%pi);
23
24 //Display:
25

```

```

26 printf('\n\nThe principal strains are           =
    %1.0f *10^-6, %1.0f*10^-6 ',strain);
27 printf('\nThe orientation of the element       =
    %1.2f degrees ',thetap1);
28 printf('\nThe average strain                   =
    %1.0f *10^-6 ',strain_avg);
29
30 //

```

END

Scilab code Exa 10.5 StnT5

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.5 : ")
4
5 //Given:
6 ep_x = 250; //(*10^-6) Normal Strain
7 ep_y = -150; //*(10^-6) Normal Strain
8 gamma_xy = 120; //*(10^-6) Shear Strain
9
10 //Orientation of the element:
11 thetas = 90 - 2*8.35;
12 thetas1 = thetas/2;
13
14 //Maximum in-plane shear strain:
15
16 l = (ep_x - ep_y)/2;
17 tou = gamma_xy/2;
18 R = sqrt( l^2 + tou^2);
19 max_inplane_strain = 2*R;
20

```

```

21
22 strain_avg = (ep_x + ep_y)/2;
23
24
25 //Display:
26
27 printf('\n\nThe orientation of the element
           = %1.1f degrees ',thetas1);
28 printf('\n\nThe maximum in-plane shear strain
           = %1.0f *10^-6 ',max_inplane_strain);
29 printf('\n\nThe average strain
           = %1.0f *10^-6 ',strain_avg);
30
31 //

```

END

Scilab code Exa 10.6 StnT6

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 10.6 : ")
4
5 //Given:
6 ep_x = -300; //(*10^-6) Normal Strain
7 ep_y = -100; //*(10^-6) Normal Strain
8 gamma_xy = 100; //*(10^-6) Shear Strain
9 theta = 20; //degrees
10
11
12 //Construction of the circle:
13 strain_avg = (ep_x+ ep_y)/2;
14 tou = gamma_xy/2;

```

```

15 R = sqrt((-ep_x + strain_avg)^2 + tou^2);
16
17 //Strains on Inclined Element:
18 theta1 = 2*theta;
19
20 phi = atan((tou)/(-ep_x +strain_avg));
21 phi = phi*(180/%pi);
22 psi = theta1 - phi;
23 psi = psi*(%pi/180);
24
25 ep_x1 = -(-strain_avg+ R*cos(psi));
26 gamma_xy1 = -(R*sin(psi))*2;
27
28 ep_y1 = -(-strain_avg - R*cos(psi));
29
30 //Display:
31
32 printf('\n\nThe normal strain in the new x direction
          = %1.0f *10^-6 ',ep_x1);
33 printf('\n\nThe normal strain in the new y direction
          = %1.1f *10^-6 ',ep_y1);
34 printf('\n\nThe shear strain in the new xy direction
          = %1.0f *10^-6 ',gamma_xy1);
35 printf('\n\nThe average strain
          = %1.0f *10^-6 ',
          strain_avg);
36
37 //

```

END

Scilab code Exa 10.7 StnT7

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 10.7 : ")
4
5 //Given:
6 ep_x = -400; //(*10^-6) Normal Strain
7 ep_y = 200; //*(10^-6) Normal Strain
8 gamma_xy = 150; //*(10^-6) Shear Strain
9
10 //Maximum in-plane Shear Strain:
11 strain_avg = (ep_x+ ep_y)/2;
12 tou = gamma_xy/2;
13
14 R = sqrt((-ep_x + strain_avg)^2 + tou^2);
15 strain_max = strain_avg + R;
16 strain_min = strain_avg - R;
17
18 max_shear_strain = strain_max - strain_min;
19
20 //Absolute Maximum Shear Strain:
21 abs_max_shear = max_shear_strain;
22
23 //Display:
24
25 printf('\n\nThe maximum in-plane principal strain
        = %1.0f *10^-6 ', strain_max);
26 printf('\n\nThe minimum in-plane principal strain
        = %1.0f *10^-6 ', strain_min);
27 printf('\n\nThe maximum in-plane shear strain
        = %1.0f *10^-6 ', max_shear_strain);
28 printf('\n\nThe absolute maximum shear strain
        = %1.0f *10^-6 ', abs_max_shear);
29 printf('\n\nThe average strain
        = %1.0f *10^-6 ', strain_avg);
30 //

```

END

Scilab code Exa 10.8 StnT8

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 10.8 : ")
4
5 //Given:
6 ep_a = 60; //>(*10^-6) Normal Strain
7 ep_b = 135; //*(10^-6) Normal Strain
8 ep_c = 264; //*(10^-6) Normal Strain
9
10 theta_a = 0;
11 theta_b = 60*(%pi/180);
12 theta_c = 120*(%pi/180);
13
14 //Using matrices to solve the equations:
15 a1 = (cos(theta_a))^2;
16 b1 = (sin(theta_a))^2;
17 c1 = cos(theta_a)*sin(theta_a);
18
19 a2 = (cos(theta_b))^2;
20 b2 = (sin(theta_b))^2;
21 c2 = cos(theta_b)*sin(theta_b);
22
23 a3 = (cos(theta_c))^2;
24 b3 = (sin(theta_c))^2;
25 c3 = cos(theta_c)*sin(theta_c);
26
27 A = [a1 b1 c1 ; a2 b2 c2; a3 b3 c3 ]
28 b = [ep_a ; ep_b ; ep_c];
29 strain = A\b;
30
31 ep_x = strain(1);
```

```

32 ep_y = strain(2);
33 gamma_xy = strain(3);
34
35 strain_avg = (ep_x + ep_y )/2;
36 tou = gamma_xy/2;
37
38 R = sqrt((-ep_x + strain_avg)^2 + tou^2);
39
40 ep1 = strain_avg + R;
41 ep2 = strain_avg - R;
42 ep = [ep1 ep2];
43
44 tan_thetap = atan(-tou/(-ep_x + strain_avg));
45 thetap = tan_thetap/2;
46 thetap2 = thetap*(180/%pi);
47
48 //Display:
49
50
51 printf('\n\nThe maximum in-plane principal strains
      are      = %1.0f *10^-6 , %1.1f *10^-6 ',ep);
52 printf('\nThe angle of orientation
      = %1.1f degrees ',thetap2);
53
54 //

```

END

Scilab code Exa 10.9 StnT9

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.9 : ")

```

```

4
5 //Given:
6 E_st = 200*10^9; //GPa
7 nu_st = 0.3; //Poisson's ratio
8 ep1 = 272 *10^-6;
9 ep2 = 33.8 *10^-6;
10
11 //Solving for sigma using matrices:
12 A = [1 -nu_st; -nu_st 1];
13 b = [(ep1*E_st) ; (ep2*E_st)];
14 sigma = A\b;
15
16 sigma1= sigma(1)/(10^6);
17 sigma2= sigma(2)/(10^6);
18
19 //Display:
20
21 printf('\n\nThe principal stresses at point A are
           = %1.0f MPa , %1.1f MPa',sigma1, sigma2);
22
23
24 //

```

END

Scilab code Exa 10.10 StnT10

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.10 : ")
4
5 //Given:
6 a = 300; //mm

```

```

7 b = 50; //mm
8 t = 20; //mm
9 E_cu = 120*10^3; //MPa
10 nu_cu = 0.34; // Poisson 's ratio
11
12 //By inspection:
13 sigma_x = 800; //MPa
14 sigma_y = -500; //MPa
15 tau_xy = 0;
16 sigma_z = 0;
17
18 //By Hooke 's Law:
19 ep_x = (sigma_x/E_cu) - (nu_cu/E_cu)*(sigma_y +
    sigma_z);
20 ep_y = (sigma_y/E_cu) - (nu_cu/E_cu)*(sigma_x +
    sigma_z);
21 ep_z = (sigma_z/E_cu) - (nu_cu/E_cu)*(sigma_y +
    sigma_x);
22
23 //New lengths:
24
25 a_dash = a + ep_x*a;
26 b_dash = b + ep_y*b;
27 t_dash = t + ep_z*t;
28
29 //Display:
30
31 printf('\n\nThe new length      = %1.2fmm ',a_dash);
32 printf('\n\nThe new width      = %1.2f mm ',b_dash);
33 printf('\n\nThe new thickness  = %1.2f mm ',t_dash);
34 //

```

END

Scilab code Exa 10.11 StnT11

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 10.11 : ")
4
5 //Given:
6 p = 20; //kPa
7 E = 600; //kPa
8 nu = 0.45
9 a = 4; //cm
10 b = 2; //cm
11 c = 3; //cm
12
13 //Dilatation:
14 sigma_x = -p;
15 sigma_y = -p;
16 sigma_z = -p;
17
18 e = ((1-2*nu)/E)*(sigma_x + sigma_y + sigma_z);
19
20 //Change in Length:
21 ep = (sigma_x - nu*(sigma_y + sigma_z))/E;
22
23 del_a = ep*a;
24 del_b = ep*b;
25 del_c = ep*c;
26
27 //Display:
28
29 printf('\n\nThe change in length a      = %1.4fcm ',
        del_a);
30 printf('\n\nThe change in length b      = %1.5fcm ',
        del_b);
```

```

31 printf('\nThe change in length c      = %1.4fcm ',
    del_c);
32 //

```

END

Scilab code Exa 10.12 StnT12

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.12 : ")
4
5 // Given:
6 di = 60/1000; //m
7 ri = di/2;
8 d0 = 80/1000; //m
9 ro = d0/2;
10 T = 8000; //Nm
11 M = 3500; //Nm
12 sigma_y_sqr = 250^2; //MPa
13
14 // Calculations:
15 c = ro;
16 J = (%pi/2)*(ro^4 - ri^4)*(10^6);
17 I = (%pi/4)*(ro^4 - ri^4)*(10^6);
18 tou_a = (T*c)/J;
19 sigma_a = (M*c)/I;
20
21 sigma_avg = (0-sigma_a)/2;
22
23 R = sqrt(116.4^2 + sigma_avg^2);
24 sigma1 = sigma_avg + R;
25 sigma2 = sigma_avg - R;

```

```

26
27 test = (sigma1^2 - (sigma1*sigma2) + sigma2^2);
28
29
30 if(test<sigma_y_sqr)
31     printf("\n\nThe material within the pipe will
           not yield.");
32 end
33
34 //

```

END

Scilab code Exa 10.13 StnT13

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.13 : ")
4
5 //Given:
6 T = 400; //Nm
7 sigma_ult = 150*10^6; //N/m^2
8
9 //Calculations:
10
11 x = T/(%pi/2);
12 r_3 = [x/sigma_ult];
13 r = nthroot(r_3, 3);
14 r= r*1000; //in mm
15
16 //Display:
17
18 printf('\n\nThe smallest radius of the solid cast

```

```

    iron shaft      = %1.2fmm ',r);
19
20 //

```

END

Scilab code Exa 10.14 StnT14

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 10.14 : ")
4
5 // Given:
6 r = 0.5; //cm
7 sigma_yield = 360; //MPa
8 T = 3.25; //kN/cm
9 A= (%pi*r^2);
10 P = 15; //kN
11 J = (%pi/2)*(r^4);
12 sigma_y_sqr = sigma_yield^2;
13
14 // Calculations:
15 sigma_x = -(P/A)*10;
16 sigma_y = 0;
17 tou_xy = (T*r*10)/J;
18
19 k = (sigma_x + sigma_y)/2;
20 R = sqrt(k^2 + (tou_xy^2));
21
22 sigma1 = k+R;
23 sigma2 = k-R;
24 l = sigma1 - sigma2;
25

```



```
26 //Maximum Shear Stress Theory:
27 test1 = abs(1);
28
29 if(test1 >= sigma_yield)
30
31     printf("\n\nFailure occurs by Maximum Shear
        Stress Theory. ');
32 end
33
34
35 //Maximum Distortion-Energy Theory:
36 test2 = (sigma1^2 - (sigma1*sigma2) + sigma2^2);
37
38
39 if(test2 < sigma_y_sqr)
40
41     printf("\n\nFailure will not occur by Maximum
        Distortion-Energy Theory. ');
42 end
```

Chapter 11

Design of Beams and Shafts

Scilab code Exa 11.1 DBS1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 11.1 : ")
4
5 //Given:
6 sigma_allow = 170; //MPa
7 tou_allow = 100; //MPa
8
9 //Shear and Moment Diagrams:
10 V_max = 90; //kN
11 M_max = 120; //kNm
12
13 //Bending Stress:
14 S_reqd = (M_max*(10^3))/sigma_allow;
15
16 W = [60 67 64 74 80 100];
17 S = [1120 1200 1030 1060 984 987];
18
19 i = find(min(W));
20 S_chosen = S(i);
21 flag1 = 0;
```

```

22 flag2 = 0;
23
24 if (S_reqd<S_chosen)
25     flag1 =1;
26 end
27
28 //Shear Stress:
29 d = 455; //mm
30 tw = 8; //mm
31 tou_avg = (V_max*10^3)/(d*tw);
32
33 if(tou_avg<tou_allow)
34     flag2 =1;
35 end
36
37 if(flag1==1 & flag2==1)
38
39
40     printf("\n\nUse a W460X60 standard shape.');
```

```

41 end
42
43 //-----
    END
    -----

```

Scilab code Exa 11.2 DBS2

```

1 clear all; clc;
2
3 disp("Scilab Code Ex 11.2 : ")
4
5 //Given:
6 l = 200/1000; //m
7 t = 30/1000; //m

```

```

8 sigma_allow = 12; //MPa
9 tou_allow = 0.8; //MPa
10 V_nail = 1.50; //kN
11 l_bc = 2; //m
12 l_cd = 2; //m
13
14 //Shear and Moment Diagrams:
15 V_max = 1.5; //kN
16 M_max = 2; //kNm
17
18 //Bending Stress:
19 y1 = l/2;
20 A1 = l*t;
21 y2 = l+(t/2);
22 A2 = t*l;
23 y_dash = (y1*A1 + y2*A2)/(A1 + A2);
24
25 I1 = (t*l^3)/12 + (t*l*(y_dash - y1)^2);
26 I2 = (l*t^3)/12 + (t*l*(y2 - y_dash)^2);
27 I = I1 + I2;
28
29 c = y_dash;
30 sigma = (M_max*c)/(I);
31 flag1 = 0;
32 sigma_allow = sigma_allow*1000; //kPa
33
34 if(sigma<sigma_allow)
35     flag1 = 1;
36 end
37
38 //Shear Stress:
39 y3 = y_dash/2;
40 A3 = y_dash*t;
41 Q = y3*A3;
42
43 tou = (V_max*Q)/(I*t);
44 tou_allow = tou_allow*1000; //kPa
45 flag2 = 0;

```

```

46
47 if(tou<tou_allow)
48     flag2 = 1;
49 end
50
51 //Nail Spacing:
52 y4a = (l+t-y_dash);
53 y4 = y4a - (t/2);
54 A4 = l*t;
55 Q4 = y4*A4;
56 V_bc = 1.5; //kN
57 V_cd = 1; //kN
58
59 q_bc = (V_bc*Q4)/I;
60 q_cd = (V_cd*Q4)/I;
61
62 s_bc = (V_nail)/(q_bc);
63 s_cd = (V_nail)/(q_cd);
64
65 chosen_bc = 150; //mm
66 chosen_cd = 250; //mm
67
68 if(flag1==1 & flag2==1)
69
70     printf('\n\nThe design is safe in bending and
71           shear. ');
72     printf('\nThe calculated nail spacing BC = %1.3 f
73           m',s_bc);
74     printf('\nThe calculated nail spacing CD = %1.3 f
75           m',s_cd);
76     printf('\nThe chosen nail spacing BC      = %1.0 f
77           mm',chosen_bc);
78     printf('\nThe chosen nail spacing CD      = %1.0 f
79           mm',chosen_cd);
80
81 end
82
83 //

```

END

Scilab code Exa 11.3 DBS3

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 11.3 : ")
4
5 //Given:
6 udl = 12; //kN/m
7 h_by_a = 1.5;
8 sigma_allow = 9; //MPa
9 tou_allow = 0.6; //MPa
10
11 //Shear and Moment Diagrams:
12 V_max = 20; //kN
13 M_max = 10.67; //kNm
14
15 //Bending Stress:
16 S_reqd = (M_max)/(sigma_allow*1000);
17 c = h_by_a/2;
18 a_cube = (S_reqd*c*12)/(1.5^3); //S_reqd = I/c
19 a = a_cube^(1/3);
20
21
22 A = a*h_by_a*a;
23 tou_max = (1.5*V_max)/(A*1000);
24
25
26 if(tou_max>tou_allow)
27     a_sqr = (3/2)*(V_max)/(h_by_a*tou_allow*1000);
28     a = sqrt(a_sqr);
29 end
```

```

30
31 //Display :
32
33     printf("\n\nThe smallest width for the
           laminated wooden beam = %1.3 f m', a);
34
35 //-----
   END
-----

```

Scilab code Exa 11.6 DBS6

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 11.6 : ")
4
5  //Given:
6  tou_allow = 50*10^6; //MPa
7  T = 7.5; //Nm
8  R_ah = 150; //N
9  R_av = 475; //N
10 l_ac = 0.25; //m
11
12 mc = R_ah*l_ac;
13 m = R_av*l_ac;
14
15 M_c = sqrt(m^2 + mc^2);
16
17 k = sqrt(M_c^2 + T^2);
18 c1 = (2*k)/(pi*tou_allow);
19 c = c1^(1/3);
20
21 d = 2*c*1000;
22

```

```
23 //Display :
24
25     printf("\n\nThe smallest allowable diameter of
           the shaft = %1.1f mm', d);
26
27 //-----
   END
-----
-----
```


Chapter 12

Deflection of Beams and Shafts

Scilab code Exa 12.6 DefBS6

```
1
2 clear all; clc;
3
4 disp(" Scilab Code Ex 12.6 : ")
5
6 //Display:
7     printf("\n\nRefer to the relation derived in the
8           book. ');
9 //-----
   END
-----
```

Scilab code Exa 12.10 DefBS10

```
1 clear all; clc;
2
```

```

3 disp(" Scilab Code Ex 12.10 : ")
4
5 //Given:
6 E = 200*10^6; //kN/m^2
7 I = 17*10^-6; //mm^4
8 l_ac = 2; //m
9 l_cF = 4; //m
10 l_Fb = 2; //m
11 l_cb = 6; //m
12 l_aF = 6; //m
13 l_ab = 8; //m
14 F = 16; //kN
15 R_b = (F*l_cb)/l_ab;
16 R_a = F - R_b;
17
18 mc = R_a*l_ac;
19 mf = R_b*l_Fb;
20 theta_ca = (0.5*l_ac*mc)/(E*I);
21
22 A1 = 0.5*l_aF*mf;
23 t1_ba = (l_Fb + l_aF/3)*(A1);
24
25 A2 = 0.5*l_Fb*mf;
26 t2_ba = (l_Fb*2*A2)/3;
27
28 t_ba = (t1_ba+t2_ba)/(E*I);
29
30 theta_c = (t_ba/l_ab)-(theta_ca);
31
32 //Display:
33
34 printf("\n\nThe slope at point C of the steel beam
        = %1.5f rad ', theta_c);
35
36
37 //-----
        END

```

Scilab code Exa 12.12 DefBS12

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 12.12 : ")
4
5 //Given:
6 E = 200; //kN/m^2
7 I = 50*10^6; //mm^4
8 l_ab = 4; //m
9 l_bc = 4; //m
10 l_ac = l_ab+l_bc;
11 R_a = -25; //kN
12 R_b = 50; //kN
13 R_c = 25; //kN
14
15 mb = R_a*l_ab;
16
17 //Moment–Area Theorem:
18
19 t_ca = (l_ab*0.5*l_ac*mb*(10^3)^3)/(E*I);
20 t_ba = (l_ab*0.5*l_ab*mb*(10^3)^3)/(E*I*3);
21
22 del_c = -t_ca + 2*t_ba;
23
24 //Display:
25
26 printf("\n\nThe displacement at point C for the
       steel overhanging beam = %1.1f mm', del_c);
27
28
29 //-----
      END
```

Scilab code Exa 12.13 DefBS13

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 12.13 : ")
4
5 // Given:
6 w = 2; //kN/m
7 L = 8; //m
8 P = 8; //kN
9
10 // Calculations:
11 EI_theta_A1 = (3*w*L^3)/(128); //ThetaA1 = (3wL^3)
    /(128EI)
12 EI_nu_C1 = (5*w*L^4)/(768); //NuC1 = (5wL^4)/(768EI)
13
14 EI_theta_A2 = (P*L^2)/(16); //theta_A2 = (PL^2)/(16
    EI)
15 EI_nu_C2 = (P*L^3)/(48); //nu_C2 = (PL^3)/(48EI)
16
17 theta_A = EI_theta_A1 + EI_theta_A2;
18 nu_C = EI_nu_C1 + EI_nu_C2;
19
20 // Display:
21
22 printf('\n\nThe slope at A in terms of EI
    = %1.0f/EI kNm^2',theta_A);
23 printf('\n\nThe displacement at point C in terms of EI
    = %1.0f/EI kNm^3',nu_C);
24
25 //
```

END

Scilab code Exa 12.14 DefBS14

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 12.14 : ")
4
5 //Given:
6 w = 5; //kN/m
7 l_ab = 4; //m
8 l_bc = 2; //m
9 P = 10; //kN
10 M = w*l_ab; //kNm
11
12 //Calculations:
13 EI_theta_B1 = (w*l_ab^3)/(24); //ThetaB1 = (wL^3)
    /(24EI)
14 EI_nu_C1 = l_bc*EI_theta_B1;
15
16 EI_theta_B2 = (M*l_ab)/(3); //
17 EI_nu_C2 = l_bc*EI_theta_B2;
18
19 EI_nu_C3 = (P*l_bc^3)/(3); //nuC3 = (PL^3)/(24EI)
20
21 nu_C = -EI_nu_C1 + EI_nu_C2 + EI_nu_C3;
22
23 //Display:
24
25 printf('\n\nThe displacement at end C of the
    overhanging beam, in terms of EI = %1.1f/EI kNm
    ^3',nu_C);
26
```

27 //

END

Scilab code Exa 12.15 DefBS15

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 12.15 : ")
4
5 //Given:
6 w = 4; //kN/m
7 l = 10; //m
8 l_bc =3; //m
9
10 //Calculations:
11 EI_theta_B = (w*l^3)/(24); //ThetaB1 = (wL^3)/(24EI)
12 EI_nu_B = (w*l^4)/(30); //nuB = (wL^4)/(30EI)
13
14 nu_C = EI_nu_B + (EI_theta_B*l_bc);
15
16 //Display:
17
18 printf('\n\nThe displacement at end C of the
        cantilever beam, in terms of EI = %1.0f/EI kNm
        ^3',nu_C);
19
20 //
```

END

Scilab code Exa 12.16 DefBS16

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 12.16 : ")
4
5 //Given:
6 k = 45; //kN/m
7 F = 3; //kN
8 E = 200*10^6; //kPa
9 l_ab = 3; //m
10 l_ac = 1; //m
11 l_cb = 2; //m
12 I = 4.687*10^-6; //m^4
13 R_a = (F*l_cb)/(l_ab);
14 R_b = F-R_a;
15
16 //Calculations:
17 nu_a = (R_a)/k;
18 nu_b = (R_b)/k;
19
20 nu_c1 = nu_b + (l_cb/l_ab)*(nu_a - nu_b);
21 nu_c2 = ((F*l_ac*l_cb)*(l_ab^2 - l_ac^2 - l_cb^2))
        /(6*E*I*l_ab);
22
23 nu_c = nu_c1 + nu_c2;
24 nu_C = nu_c*1000;
25
26 //Display:
27
28 printf('\n\nThe vertical displacement of the force
        at C = %1.3f mm',nu_C);
29
30 //
```

END

Scilab code Exa 12.21 DefBS21

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 12.21 : ")
4
5 //Given:
6 l = 3; //m
7 l_af = l/2; //m
8 P = 8; //kN
9 w = 6; //kN/m
10
11 //Compatibility Equation:
12 EI_nu_b1 = (w*l^4)/8 + (5*P*l^3)/48; //nu_b = (wl^4)
    /8EI + (5Pl^3)/48EI
13 EI_nu_b2 = (l^3)/3;
14
15 B_y = EI_nu_b1 / EI_nu_b2;
16
17 //Display:
18
19 printf("\n\nThe reactions at roller support B =
    %1.2f kN', B_y);
20
21
22 //-----
    END
```

Scilab code Exa 12.22 DefBS22

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 12.22 : ")
4
5 //Given:
6 l = 8; //m
7 l_ab = l/2; //m
8 l_bc = l/2; //m
9 l_af = l_ab/2; //m
10 b = 12/1000; //m
11 w = 24; //kN/m
12 E = 200*10^6; //Kn/m^2
13 I = 80*10^-6; // m^4
14
15 //Compatibility Equation:
16 nu_b = (5*w*l^4)/(768*E*I); //nu_b = (5wl^4)/768EI
17 nu_b_byBy = (l^3)/(48*E*I); //nu_b' = (Pl^3)/48EI
18
19 B_y = (nu_b-b)/nu_b_byBy;
20
21 C_y = ((w*l_ab*l_af) - (B_y*l_ab))/l;
22
23 A_y = (w*l_ab - B_y - C_y);
24
25 //Display:
26
27 printf('\n\nThe reaction at A = %1.0f kN',A_y);
28 printf('\n\nThe reaction at B = %1.0f kN',B_y);
29 printf('\n\nThe reaction at C = %1.0f kN',C_y);
30
31
32 //
```

END

Scilab code Exa 12.23 DefBS23

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 12.23 : ")
4
5 //Given:
6 d = 12; //mm
7 E = 210; //GPa
8 I = 186*10^6; //mm^4
9 P = 40; //kN
10 l_bc = 3; //m
11 l_ab = 4; //m
12 l = 5; //m
13
14 //Compatibility Equation: nuB'' = nuB - nuB'
15 A = (%pi/4)*(d^2);
16
17 nuB1_by_Fbc = (l_bc*1000)/(A*E*1000); //nuB'' = PL/
    AE
18 nuB2 = (5*P*1000*(l_ab*1000)^3)/(48*E*1000*I); //nuB
    = (5PL^3)/(48EI)
19 nuB2_by_Fbc = ((l*1000)^3)/(3*E*1000*I); //nuB' = (
    PL^3)/(3EI)
20
21 F_bc = (nuB2)/(nuB1_by_Fbc + nuB2_by_Fbc );
22 F_bc = F_bc/1000; //in kN
23
24 //Display:
25
```

```

26 printf('\n\nThe force in the rod due to loading    =
      %1.3 f kN',F_bc);
27
28
29 //

```

END

Scilab code Exa 12.24 DefBS24

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 12.24 : ")
4
5 //Given:
6 l_ab = 4; //m
7 l = l_ab/2;
8 w = 9; //kN/m
9
10 //Compatibility Equations:
11
12 EI_thetaB = (w*l_ab^3)/(48); //thetaB = (wL^3)/(48EI
    )
13 EI_nuB = (7*w*l_ab^4)/(384); //nuB = (7wl^4)/(384EI)
14
15 //Only redundant By applied:
16 EI_thetaB_by_By = (l_ab^2)/(2); //thetaB' = (PL^2)
    /(2EI)
17 EI_nuB_by_By = (l_ab^3)/(3); //nuB' = (PL^3)/(3EI)
18
19 //Only redundant Mb is applied:
20 EI_thetaB_by_Mb = l_ab; //thetaB'' = (ML)/(EI)
21 EI_nuB_by_Mb = (l_ab^2)/(2); //nuB'' = (ML^2)/(2EI)

```

```

22
23 //Solving for By and Mb using matrices:
24
25 A = [EI_thetaB_by_By EI_thetaB_by_Mb; EI_nuB_by_By
      EI_nuB_by_Mb ];
26 b = [-EI_thetaB; -EI_nuB ] ;
27 moments = A\b;
28
29 By = moments(1);
30 Mb = moments(2);
31
32 //Display:
33
34 printf('\n\nThe vertical force at B for the beam
      = %1.3 f kN',By);
35 printf('\n\nThe moment at B for the beam          =
      %1.2 f kNm',Mb);
36
37 //

```

END

Chapter 13

Buckling of Columns

Scilab code Exa 13.1 BoC1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 13.1 : ")
4
5 // Given:
6 l = 7.2*1000; //mm
7 E = 200; //GPa
8 ro = 75; //mm
9 ri = 70; //mm
10 sigma_y = 250; //MPa
11
12 // Calculations:
13 I = (%pi/4)*(ro^4 - ri^4)
14 A = (%pi)*(ro^2 -ri^2);
15
16 Pcr = (%pi^2*(E*10^6)*I*(1000)^-2)/(l^2); //Pcr = (
    %pi^2*EI)/(l^2)
17
18 sigma_cr = (Pcr*1000)/A;
19
20 if(sigma_cr<sigma_y)
```

```

21
22     printf("\n\nThe maximum allowable axial load
           that the column can support    = %1.1f kN',
           Pcr);
23 end
24
25 //-----
    END
-----

```

Scilab code Exa 13.2 BoC2

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 13.2 : ")
4
5  // Given:
6  E = 200; //GPa
7  I = 15.3*10^6; //mm^4
8  l= 4*1000; //mm
9  A = 5890; //mm^2
10 sigma_y = 250; //MPa
11
12 // Calculations:
13
14 Pcr = ((%pi^2)*E*10^6*I*1000^-2)/(l^2); //Pcr = (%pi
      ^2*EI)/(l^2)
15
16 sigma_cr = (Pcr*1000)/A;
17
18 if(sigma_cr>sigma_y)
19     Pcr = (sigma_y*A);
20     Pcr = Pcr/1000; //in kN
21 end

```

```

22
23 //Display:
24
25 printf("\n\nThe maximum allowable axial load that
    the column can support = %1.1f kN',Pcr);
26
27 //-----
    END
    -----

```

Scilab code Exa 13.3 BoC3

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 13.3 : ")
4
5 //Given:
6 E = 200; //GPa
7 Ix = 13.4*10^-6;
8 Iy = 1.83*10^-6;
9 l = 8;
10 KLx = 0.5*l; //m
11 KLy = 0.7*(l/2); //m
12 rx = 66.2; //mm
13 ry = 24.5; //mm
14
15 Pcrx = (%pi^2*E*10^6*Ix)/(KLx^2); //Pcr = (%pi^2*EI)
    /(l^2)
16 Pcry = (%pi^2*E*10^6*Iy)/(KLy^2); //Pcr = (%pi^2*EI)
    /(l^2)
17
18 Pcr = min(Pcrx,Pcry);
19 A = 3060; //mm^2
20 sigma_cr = Pcr/A;

```

```

21
22 sl_ratio_x = (KLx*1000)/(rx);
23 sl_ratio_y = (KLy*1000)/(ry);
24 s_ratio = max(sl_ratio_x, sl_ratio_y);
25
26 //Display:
27
28 printf("\n\nThe maximum load that the column can
      support without buckling      = %1.0f kN',Pcr);
29 printf("\nThe largest slenderness ratio
      = %1.1f N/mm
      ^2',s_ratio);
30 //

```

END

Scilab code Exa 13.4 BoC4

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 13.4 : ")
4
5 //Given:
6 E = 70; //GPa
7 Ix = 61.3*10^-6;
8 Iy = 23.2*10^-6;
9 l = 5;
10 KLx = 2*l; //m
11 KLy = 0.7*(l); //m
12 FS = 3; //Factor of safety
13 sigma_y = 215; //MPa
14
15

```



```

16 Pcrx = (%pi^2*E*10^6*Ix)/(KLx^2); //Pcr = (%pi^2*EI)
      /(l^2)
17 Pcry = (%pi^2*E*10^6*Iy)/(KLy^2); //Pcr = (%pi^2*EI)
      /(l^2)
18
19 Pcr = min(Pcrx,Pcry);
20 A = 7.5*10^-3; //mm^2
21 P_allow = Pcr/FS;
22 sigma_cr = (Pcr*10^-3)/A;
23
24
25 if(sigma_cr<sigma_y)
26
27     printf("\n\nThe largest allowable load that the
            column can support      = %1.0 f kN', P_allow);
28 end
29
30 //-----
      END

```

Scilab code Exa 13.5 BoC5

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 13.5 : ")
4
5 //Given:
6 E = 200*10^3; //MPa
7 sigma_y = 250; //MPa
8 x1 = 50; //mm
9 y1 = 75; //mm
10 z1 = 4.5; //m
11 e = 25; //mm

```

```

12
13 Ix = (1/12)*x1*(y1*2)^3;
14 A = x1*2*y1;
15 rx = sqrt(Ix/A);
16 L = z1*1000;
17 KL = 1*L;
18
19 sl_ratio = KL/rx;
20 c = y1;
21 ec_r = e*c/(rx^2);
22 P_a = 83; //MPa
23 A = 7500; //mm^2
24 P = P_a*A;
25 P = P/1000; //in kN
26
27 k = (L/(2*rx))*(sqrt(P/(E*A)));
28 sigma_max = (P*1000/A)*(1+ec_r*sec(k)); //Secant
    Formula
29
30 l = sqrt((P*1000)/(E*Ix));
31 nu_max = e*(sec(l*L/2)-1);
32
33 //Display:
34
35 printf('\n\nThe allowable eccentric load that can be
    applied on the column = %1.1fkN',P);
36 printf('\n\nThe maximum deflection of the column due
    to the loading = %1.0f mm',nu_max);
37
38 //

```

END

Scilab code Exa 13.6 BoC6

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 13.6 : ")
4
5 //Given:
6 z1 = 4*1000; //mm
7 e = 200; //mm
8 KLy = 0.7*z1;
9 Iy = 20.4*10^6;
10 E = 200*10^3; //N/mm^2
11 sigma_y =250; //MPa
12
13 //y-y Axis Buckling:
14 Pcry = (%pi^2*E*10^6*Iy)/(KLy^2); //Pcr = (%pi^2*EI)
    /(l^2)
15 Pcry = Pcry/1000;
16
17 //x-x Axis Yielding:
18 Kx= 2;
19 KLx = Kx*z1;
20 c = (z1-KLy)/2;
21 rx = 89.9;
22
23 //Solved by applying the Secant Formula and then
    finding Px by trial and error:
24
25 trial_Px = 419.4; //kN
26
27 A = 7850; //mm^2
28 sigma = (trial_Px*1000)/(A);
29
30 if(Pcry>trial_Px & sigma<sigma_y)
31 printf('\n\nThe maximum eccentric load that the
    column can support = %1.1fkN',trial_Px);
32 printf('\nFailure will occur about the x-x axis.');
```

```
34 end
35
36 //
```

END

Scilab code Exa 13.7 BoC7

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 13.7 : ")
4
5 // Given:
6 d = 30; //mm
7 r = d/2;
8 L = 600; //mm
9 sigma_pl = 150; //MPa
10
11 // Calculations:
12 I = (%pi/4)*(r^4);
13 A = %pi*r^2;
14 r_gyr = sqrt(I/A);
15 K = 1;
16 sl_ratio = (K*L)/(r_gyr);
17 flag1 = 0;
18
19 // Assuming the critical stress is elastic:
20 E = 150/0.001;
21 sigma_cr1 = (%pi^2*E)/(sl_ratio^2); //Pcr = (%pi^2*
    EI)/(l^2)
22
23
24 if(sigma_cr1 > sigma_pl)
```

```

25     Et = (270 - 150)/(0.002 - 0.001);
26     sigma_cr2 = (%pi^2*Et)/(sl_ratio^2); //Pcr = (
        %pi^2*EI)/(l^2)
27
28     if(sigma_cr2>150 & sigma_cr2<270)
29         Pcr = sigma_cr2*A;
30         Pcr = Pcr/1000; //in kN
31         printf('\n\nThe critical load when used as a
        pin supported column = %1.0fkN ',Pcr);
32
33     end
34
35
36 end

```

Scilab code Exa 13.8 BoC8

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 13.8 : ")
4
5  //Given:
6  E = 200*10^3; //MPa
7  sigma_y = 250; //MPa
8  L = 5*1000; //mm
9  K = 1;
10 A = 19000; //mm^2
11 rx = 117; //mm
12 ry = 67.4; //mm
13
14 //Calculations:
15 sl_ratio = (K*L)/(ry);
16 sl_ratio_c = sqrt((2*pi^2*E)/(sigma_y));
17
18 if(sl_ratio>0 & sl_ratio<sl_ratio_c)

```

```

19     num = (1 - (sl_ratio^2/(2*sl_ratio_c^2)))*
        sigma_y;
20     denom1 = (5/3) + ((3/8)*sl_ratio/sl_ratio_c);
21     denom2 = (sl_ratio^3)/(8*sl_ratio_c^3);
22     sigma_allow = num/(denom1 - denom2);
23
24     P = sigma_allow*A;
25     P = P/1000;
26     printf('\n\nThe largest load the pin
        supported column can safely bear = %1.0 f
        kN',P);
27
28     end
29
30 //

```

END

Scilab code Exa 13.9 BoC9

```

1  clear all; clc;
2
3  disp(" Scilab Code Ex 13.9 : ")
4
5  // Given:
6  P = 80; //kN
7  E = 210*10^3; //MPa
8  sigma_y = 360; //MPa
9  L = 5000; //mm
10 K = 0.5;
11
12 // Calculations:
13 I_by_d = (1/4)*(%pi)*(0.5^4);

```

```

14 A_by_d = (1/4)*(%pi);
15 r_by_d = sqrt(I_by_d/A_by_d);
16
17 sl_ratio_c = sqrt((2*%pi^2*E)/(sigma_y));
18 sigma_allow = (P*1000)/A_by_d;
19
20 d4 = (sigma_allow*23*(K*L)^2*16)/(12*%pi^2*E);
21 d = d4^(1/4);
22
23 //Check:
24 d = ceil(d);
25 r = d/4;
26 KL_r = (K*L)/r;
27
28
29 if(KL_r>sl_ratio_c & KL_r<200)
30     printf('\n\nThe smallest diameter of the rod
           as allowed by AISC specification = %1.0
           fmm',d);
31
32 end
33
34 //

```

END

Scilab code Exa 13.10 BoC10

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 13.10 : ")
4
5 //Given:

```

```

6 L = 750; //mm
7 P = 60; //kN
8 sigma = 195; //N/mm^2
9 K = 1;
10
11 // Calculations:
12 b2 = (P*1000)/(2*sigma);
13 b = sqrt(b2);
14
15 A = 2*b*b;
16 Iy = (1/12)*(2*b*b^3);
17 ry = sqrt(Iy/A);
18
19 sl_ratio = (K*L)/(ry);
20
21
22
23 if(sl_ratio>12)
24     b4 = (P*1000*2598.1^2)/(2*378125); //Eqn 13.26
25     b = b4^(1/4);
26
27     sl_ratio = (2598.1)/(b);
28     w = 2*b;
29
30     if(sl_ratio>55)
31         printf('\n\nThe thickness of the bar = %1.0
32             fmm',b);
33         printf('\n\nThe width of the bar = %1.0fmm
34             ',w);
35     end
36 end
37 //

```

END

Scilab code Exa 13.11 BoC11

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 13.11 : ")
4
5 //Given:
6 P = 20*10^3; //N
7 y1 = 150; //mm
8 x1 = 40; //mm
9 A = (x1*y1);
10 d = 40;
11 K = 1;
12
13 //Eqn 13.29
14
15 L2 = (3718*A*d^2)/(P);
16 L = sqrt(L2);
17 KL_d = (K*L)/(d);
18
19 if(KL_d>26 & KL_d<=50)
20     printf('\n\nThe greatest allowable length L as
21           specified by the NFPA = %1.0f mm',L);
22 end
23
24 //
```

END

Scilab code Exa 13.12 BoC12

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 13.12 : ")
4
5 //Given:
6 L = 1600; //mm
7 K = 2;
8 l = 80; //mm
9 b = 40; //mm
10 e = 20; //mm
11 c = 40; //mm
12
13 //Calculations:
14 I1 = (1/12)*(l*b^3);
15 A = l*b;
16 r = sqrt(I1/A);
17 sl_ratio = (K*L)/(r);
18
19 //Eqn 13.26:
20 sigma_allow = (378125)/(sl_ratio^2);
21
22 I2 = (1/12)*(b*l^3);
23 coefficient = (1/A) + (e*c)/I2;
24 sigma_max = sigma_allow;
25 P = sigma_max/coefficient;
26 P = P/1000;
27
28 //Display:
29
30 printf('\n\nThe load that can be supported if the
        column is fixed at its base = %1.2f kN',P);
31
32 //
```

END

Scilab code Exa 13.13 BoC13

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 13.13 : ")
4
5 //Given:
6 sigmaB_allow = 160; //MPa
7 E = 200; //GPa
8 sigma_y = 250; //MPa
9 K= 1;
10 A = 3790; //mm^2
11 Ix = 17.1*10^6; //mm^4
12 ry = 38.2; //mm
13 d = 157; //mm
14 c= d/2;
15 e = 750; //mm
16 L = 4000; //mm
17
18 sl_ratio = (K*L)/(ry);
19 sl_ratio_c = sqrt((2*pi^2*E*1000)/(sigma_y));
20
21
22
23 if(sl_ratio<sl_ratio_c)
24     num = (1 - (sl_ratio^2/(2*sl_ratio_c^2)))*
25           sigma_y;
26     denom1 = (5/3) + ((3/8)*sl_ratio/sl_ratio_c);
27     denom2 = (sl_ratio^3)/(8*sl_ratio_c^3);
28     sigmaA_allow = num/(denom1 - denom2);
29
30     coeffP = 1/(sigmaA_allow*A) + (e*c)/(Ix*
31           sigmaB_allow);
```

```

30     P = 1/coeffP;
31
32     sigA = (P/A)/(sigmaA_allow);
33     P = P/1000; //in kN
34
35
36     if(sigA < 0.15)
37         printf('\n\nThe maximum allowable value of
38             eccentric load = %1.2f kN',P);
39     end
40
41 //

```

END

Scilab code Exa 13.14 BoC14

```

1
2 clear all; clc;
3
4 disp(" Scilab Code Ex 13.14 : ")
5
6 //Given:
7 K = 2;
8 d= 60; //mm
9 L = 1200; //mm
10 e = 80; //mm
11 c = d;
12 A = 60*120; //mm^2
13 l = 60; //mm
14 b = 120; //mm
15

```

```

16
17 // Calculations:
18 sl_ratio = (K*L)/(d);
19
20 if(sl_ratio>26 & sl_ratio<50)
21     sigma_allow = (3718)/(sl_ratio^2);
22     sigma_max = sigma_allow;
23
24     I = (1/12)*(1*b^3);
25     coeffP = (1/A) + (e*c)/(I);
26     P = sigma_max/coeffP;
27     P = P/1000; //kN
28
29     printf('\n\nThe eccentric load that can be
        supported = %1.2f kN',P);
30 end
31
32 //Answer given in the textbook varies.
33
34 //

```

END

Chapter 14

Energy Methods

Scilab code Exa 14.1 EM1

```
1 clear all; clc;
2
3 disp(" Scilab Code Ex 14.1 : ")
4
5 //Given:
6 sigma_y = 310; //N/mm^2
7 db =18; //mm
8 rb = db/2;
9 Ab = %pi*(rb^2);
10 E = 210*10^3; //N/mm^2
11 da1 = 20; //mm
12 ra1 = da1/2;
13 Aa1 = %pi*(ra1^2);
14 La1 = 50; //mm
15 La2= 6; //mm
16 da2 =18; //mm
17 ra2 = da2/2;
18 Aa2 = %pi*(ra2^2);
19 Lb = 56; //mm
20
21
```

```

22 //Bolt A:
23 P_max = sigma_y*Ab;
24 Uia = (P_max^2/(2*E))*(La1/Aa1 + La2/Aa2); //Ui = (N
      ^2L)/(2AE)
25 Uia = Uia/1000;
26
27 //Bolt B:
28 Uib = (P_max^2/(2*E))*(Lb/Ab);
29 Uib = Uib/1000;
30
31 //Display:
32     printf('\n\nThe greatest amount of strain energy
      absorbed by bolt A    = %1.3f J',Uia);
33     printf('\n\nThe greatest amount of strain energy
      absorbed by bolt B    = %1.3f J',Uib);
34
35 //

```

END

Scilab code Exa 14.5 EM5

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 14.5 : ")
4
5 //Given:
6 G = 75*10^9; //N/m^2
7 ro = 80/1000; //m
8 t = 15/1000; //m
9 ri = ro - t;
10 l1 = 750/1000; //m
11 l2 = 300/1000; //m

```

```

12 T1 = 40; //Nm
13 T2 =15; //Nm
14
15 // Calculations:
16
17 J = (%pi/2)*(ro^4 - ri^4);
18
19 //Eqn 14-22
20 U1 = (T1^2*l1)/(2*G*J);
21 U2 = (T2^2*l2)/(2*G*J);
22 Ui = U1 + U2;
23 Ui = Ui*10^6; //in micro Joule
24
25 //Display:
26
27     printf('\n\nThe strain energy stored in the shaft
           = %1.0fX10^-6 J ',Ui);
28
29 //

```

END

Scilab code Exa 14.6 EM6

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 14.6 : ")
4
5 //Given:
6 l_ab = 1; //m
7 l_bc = 2; //m
8 N_ab = 11.547*1000; //N
9 Nb = 20*1000; //N

```



```

10 Nc = -23.094*1000; //N
11 N_ac = -20*1000; //N
12 A = 100/(1000^2); //mm^2
13 E = 200*10^9; //N/m^2
14 P = 20*10^3; //N
15
16 //Eqn 14-26
17 P_by_2 = P/2;
18 l_ac = sqrt(l_bc^2 - l_ab^2);
19 del = 0;
20
21 N2= [N_ab^2 Nc^2 N_ac^2];
22 L = [l_ab l_bc l_ac];
23
24 for i = 1:3
25     del = del + (N2(i)*L(i))/(2*A*E);
26 end
27
28 del_bh = del/P_by_2;
29 del_bh = del_bh*1000;
30
31 //Display:
32
33 printf('\n\nThe horizontal displacement at point B
        = %1.2fmm', del_bh);
34
35 //

```

END

Scilab code Exa 14.8 EM8

```

1 clear all; clc;

```

```

2
3 disp(" Scilab Code Ex 14.8 : ")
4
5 //Given:
6 ro = 60; //mm
7 ri = 50; //mm
8 E = 70; //kN/mm^2
9 W = 600; //kN
10 L = 240; //mm
11 h = 0;
12
13 //Part a:
14
15 A = (%pi)*(ro^2 - ri^2);
16 del_st= (W*L)/(A*E);
17
18 //Part b:
19
20 del_max = del_st*(1 + sqrt(1 + 2*(h/del_st)));
21
22 //Display:
23
24 printf('\n\nThe maximum displacement at the top of
      the pipe for gradually applied load = %1.4f
      mm',del_st);
25 printf('\n\nThe maximum displacement at the top of
      the pipe for suddenly applied load = %1.4
      f mm',del_max);
26
27 //

```

END

Scilab code Exa 14.9 EM9

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 14.9 : ")
4
5 //Given:
6 W = 6000; //N
7 h = 50; //mm
8 E = 210*1000; //N/mm^2
9 L = 5000; //mm
10 I = 87.3*10^6; //mm^2
11
12 //Calculations:
13
14 del_st = (W*L^3)/(48*E*I);
15 del_max = del_st*(1 + sqrt(1 + 2*(h/del_st)));
16
17 c = 252/2;
18 sigma_max = (12*E*del_max*c)/(L^2);
19
20 //Display:
21
22 printf('\n\nThe maximum bending stress in the
        steel beam = %1.2f N/mm^2',sigma_max);
23 printf('\n\nThe maximum deflection in the beam
        = %1.3f mm',del_max);
24
25 //
```

END

Scilab code Exa 14.10 EM10

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 14.10 : ")
4
5 //Given:
6 m = 80*1000; //kg
7 v = 0.2; //m/s
8 l_ac = 1.5; //m
9 E = 200*10^9; //N/m^2
10 w = 0.2; //m
11 I = (1/12)*(w^4);
12 l_ab = 1000; //mm
13
14 //Calculations:
15 del_Amax = sqrt((m*v^2*l_ac^3)/(3*E*I));
16
17 P_max = (3*E*I*del_Amax)/(l_ac^3);
18 theta_A = (P_max*l_ac^2)/(2*E*I);
19 del_Amax = del_Amax*1000;
20 del_Bmax = del_Amax + (theta_A*l_ab);
21
22
23 //Display:
24
25     printf('\n\nThe maximum horizontal displacement
           of the post at B due to impact = %1.1f mm',
           del_Bmax);
26
27
28 //

```

END

Scilab code Exa 14.11 EM11

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 14.11 : ")
4
5 //Given:
6 A = 400*10^-6; //m^2
7 E = 200*10^6; //kN/m^2
8 P = 100; //kN
9
10 //Virtual Work Equation:
11
12 n = [0 0 -1.414 1];
13 N = [-100 141.4 -141.4 200];
14 L = [4 2.828 2.828 2];
15 del_cv = 0;
16
17 for i=1:4
18     del_cv = del_cv + (n(i)*N(i)*L(i))/(A*E);
19 end
20
21 del_cv = del_cv*1000;
22
23 //Display:
24
25     printf('\n\nThe vertical displacement of joint C
26           of the steel truss = %1.1f mm',del_cv);
27
28 //
```

END

Scilab code Exa 14.12 EM12

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 14.12 : ")
4
5 //Given:
6 A = 300*10^-6; //m^2
7 E = 210*10^6; //kN/m^2
8 P = 60; //kN
9 F_ac = 1.25; //kN
10
11 //Part a:
12
13 //Virtual Work Equation:
14
15 n = [0 1.25 0 -0.75];
16 N = [0 75 -60 -45];
17 L = [1.5 2.5 2 1.5];
18 del_ch = 0;
19
20 for i=1:4
21     del_ch = del_ch + (n(i)*N(i)*L(i))/(A*E);
22 end
23
24 del_chA = del_ch*1000;
25
26 //Part b:
27
28 del_L = -6; //mm
29 del_chB = F_ac*del_L;
30
31 if(del_chB<0)
32
```

```

33
34 //Display:
35
36     printf('\n\nThe horizontal displacement of joint
           C if a force is applied to B   = %1.3f mm',
           del_chA);
37     printf('\n\nThe horizontal displacement of joint C
           if AC is fabricated short   = %1.1f mm',
           del_chB);
38 end
39
40
41
42 //

```

END

Scilab code Exa 14.13 EM13

```

1  clear all; clc;
2
3  disp("Scilab Code Ex 14.13 : ")
4
5  //Given:
6  del_T = 60; //degree celcius
7  alpha = 12*10^-6; //per degree celcius
8  E = 200*10^6; //kN/m^2
9  A = 250*10^-6; //m^2
10 L = 4; //m
11
12 //Virtual Work Equation:
13 n = 1.155; //kN
14 N = -12; //kN

```

```

15
16 del_bh = (n*N*L)/(A*E) + (n*alpha*del_T*L);
17 del_bh = del_bh*1000;
18
19 //Display:
20
21 printf('\n\nThe horizontal displacement of joint B
      of the truss = %1.2f mm',del_bh);
22
23 //

```

END

Scilab code Exa 14.16 EM16

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 14.16 : ")
4
5 //Given:
6 I = 175.8*10^-6; //m^4
7 E = 200*10^6; //kN/m^2
8 Ra = 1; //kN
9 l_ab = 3; //m
10 l_bc = 6; //m
11
12
13 //Virtual Work Equation:
14 m1 = -1; //x1
15 M1 = -2.5; //x1^3
16 m2 = -0.5; //x2
17
18 x10 = 0;

```



```

19 x11 = l_ab;
20 I1 = integrate('m1*M1*(x1^4)', 'x1', x10, x11);
21
22 x20 = 0;
23 x21 = l_bc;
24 I2 = integrate('m2*123.75*(x2^2)', 'x2', x20, x21);
25
26 x20 = 0;
27 x21 = l_bc;
28 I3 = integrate(' -m2*22.5*(x2^3)', 'x2', x20, x21);
29
30 In = I1 + I2 + I3;
31 del_A = (In)/(E*I);
32 del_A = del_A*1000;
33
34
35 //Display:
36
37     printf('\n\nThe displacement of point A of the
           steel beam    = %1.1f mm', del_A);
38
39 //

```

END

Scilab code Exa 14.17 EM17

```

1 clear all; clc;
2
3 disp(" Scilab Code Ex 14.17 : ")
4
5 //Given:
6 E = 210*10^3; //N/mm^2

```

```

7 P = 40*10^3; //N
8 A_ab = 1250; //mm^2
9 A_ac = 625; //mm^2
10 A_cd = 1250; //mm^2
11 A_bc = 625; //mm^2
12
13 N_by_P = [0 0 1.67 -1.33];
14 L = [4000 3000 5000 4000];
15 A = [A_ab A_bc A_ac A_cd];
16 N = zeros(4);
17 sum = 0;
18
19
20 for i =1:4
21     N(i) = N_by_P(i)*P;
22     num(i) = N(i)*N_by_P(i)*L(i);
23
24 end
25
26 for i = 1:4
27     sum = sum + (num(i)/(A(i)*E)); //By Castigliano '
        s Second theorem.
28 end
29
30 del_ch = sum;
31
32 //Display:
33     printf('\n\nThe horizontal displacement of joint
        C of the steel truss = %1.2f mm',sum);
34
35 //

```

END

Scilab code Exa 14.18 EM18

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 14.18 : ")
4
5 //Given:
6 E = 200*10^6; //kN/m^2
7 P = 0; //N
8 A = 400*10^-6; //m^2
9
10 N_by_P = [0 0 -1.414 1];
11 L = [4 2.828 2.828 2];
12 N = [-100 141.4 -141.4 200];
13 sum = 0;
14
15
16 for i =1:4
17     num(i) = N(i)*N_by_P(i)*L(i);
18 end
19
20 for i = 1:4
21     sum = sum + (num(i)/(A*E)); //By Castigliano's
22     Second theorem.
23 end
24 del_ch = sum*1000;
25
26 //Display:
27 printf('\n\nThe vertical displacement of joint C
28     of the steel truss = %1.1f mm',del_ch);
29 //
```

END

Scilab code Exa 14.21 EM21

```
1 clear all; clc;
2
3 disp("Scilab Code Ex 14.21 : ")
4
5 //Given:
6 I = 125*10^-6; //m^4
7 E = 200*10^6; //kN/m^2
8 Rc = 5; //kN
9 l_ac = 6; //m
10 l_cb = 4; //m
11
12
13 //Castigliano's Second Theorem:
14 m = 0.4/9;
15
16 x10 = 0;
17 x11 = l_ac;
18 I11 = integrate('4.4*(x1^2)', 'x1', x10, x11);
19 I12 = integrate('-m*(x1^4)', 'x1', x10, x11);
20 I1 = I11 + I12;
21
22 x20 = 0;
23 x21 = l_cb;
24 I21 = integrate('6*0.6*(x2^2)', 'x2', x20, x21);
25 I22 = integrate('18*0.6*(x2)', 'x2', x20, x21);
26 I2 = I21+I22;
27
28 In = I1 + I2 ;
29 del_cv = (In)/(E*I);
```

```
30 del_cv = del_cv*1000;
31
32
33 //Display:
34
35     printf('\n\nThe vertical displacement of point C
           of the steel beam = %1.1f mm',del_cv);
36
37 //
```

END
