Scilab Textbook Companion for Mechanics of Materials by R. C. Hibbeler[1]

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

May 26, 2016

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

Book Description

Title: 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

```
clear all; clc;

disp("Scilab Code Ex 1.1 :")

w_varying = 270;
l_crossection = 9;
l_cb = 6;
l_ac = 2;
w_c = (w_varying/l_crossection) * l_cb //By
    proportion, load at C is found.
f_resultant_c = 0.5* w_c *l_cb
// Equations of Equilibrium

//Balancing forces in the x direction:
n_c = 0

//Balncing forces in the y direction:
v_c = f_resultant_c

// Balncing the moments about C:
m_c = - (f_resultant_c*l_ac)
```

```


// Displaying results:

printf('\n\nThe resultant force at C   = %.2f N', f_resultant_c);
printf('\nThe horizontal force at C  = %.2f N',n_c);
printf('\nThe vertical force at C    = %.2f N',v_c);
printf('\nThe moment about C         = %.2f Nm',m_c);


// ————————————————————————————————————————————————END————————————————————————————————————————
```

Scilab code Exa 1.2 S2

```
clear all; clc;

disp("Scilab Code Ex 1.2 : ")

f_d = 225; //N
w_uniform = 800; // N/m
l_ac = 0.200; //m
l_cb = 0.05+0.1; //m
l_bd = 0.100;  //m
l_bearing = 0.05;    //m
f_resultant = w_uniform*l_cb //120N
l_f_resultant_b = (l_cb/2)+ l_bearing; //0.125m
l = l_ac + l_cb + l_bearing + l_bd


// This problem is solved by considering segment AC
```

```
of the shaft.

//Support Reactions:

m_b = 0;                              // Net moment about B is zero for equilibrium . Sum Mb = 0.
a_y = -((f_d*l_bd) - (f_resultant*l_f_resultant_b))/(l - l_bd) // finding the reaction force at A

// Refer to the free body diagram in Fig.1-5c.
f_c = 40                             //N
//Balancing forces in the x direction:
n_c = 0

//Balncing forces in the y direction:
v_c = a_y - f_c              //-18.75N - 40N-Vc = 0

// Balncing the moments about C:
m_c =  ((a_y * (l_ac + 0.05)) - f_c*(0.025) ) // Mc +40N(0.025m)+ 18.75N(0.250m) = 0


// Displaying results:

printf('\n\nThe resultant force       = %.2f N',f_resultant);
printf('\nThe reaction force at A   = %.2f N',a_y);
printf('\nThe horizontal force at C = %.2f N',n_c);
printf('\nThe vertical force at C   = %.2f N',v_c);
printf('\nThe moment about C        = %.2f Nm',m_c);

//------------------------------------------------------------------------END------------------------------------------------------------------------
```

Scilab code Exa 1.3 S3

```
clear all; clc;

disp("Scilab Code Ex 1.3 :")

// Given:
l_ac = 1;       //m.
l_cd = 1.5 ;    //m.
l_bd = 0.5;     //m.
r_a = 0.125;    //m.
r_d = 0.125;    //m.
W = 2000; // N


// Equations of equilibrium:

//Balancing forces in the x direction:
n_c = -W; // N

//Balncing forces in the y direction:
v_c = -W; //N

// Balncing the moments about C:
m_c = - (W*(r_a +l_ac)- W*r_a)


// Displaying results:

printf('\n\nThe horizontal force at C    = %.2f N',
    n_c);
printf('\nThe vertical force at C       = %.2f N',v_c)
    ;
printf('\nThe moment about C            = %.2f Nm',m_c
```

```
);

//
-------------------------------------------------------------------------------
END
-------------------------------------------------------------------------------
```

Scilab code Exa 1.4 S4

```
clear all; clc;

disp("Scilab Code Ex 1.4 :")

// Given:
l_ag = 1; //Length of AG is 1m.
l_gd = 1; //Length of GD is 1m.
l_de = 3; //Length of DE is 1m.
f_a = 1500; //Force at A is 1500N.
l_ec = 1.5;  //Length of EC is 1m.
l = l_ag +l_gd +l_de;
w_uniform_varying = 600; //Nm.

w_resultant = 0.5*l_de*w_uniform_varying;
// calling point of action of resultant as P
l_ep = (2/3)*l_de; //Distance between points P and E.
l_ap = l - l_ep; // Distance between points A and P.


f_ba = 7750; //N
f_bc = 6200; //N
f_bd = 4650; //N

//Free Body Diagram: Using the result for Fba, the
```

```
left section AG of the beam is shown in Fig 1-7d.

// Equations of equilibrium:

//Balancing forces in the x direction:
n_g = -f_ba * (4/5); // N

//Balncing forces in the y direction:
v_g = -f_a + f_ba*(3/5); //N

// Balncing the moments about C:
m_g = (f_ba * (3/5)*l_ag) - (f_a * l_ag); //Nm



// Displaying results:


printf('\n\nThe horizontal force at G = %.2f N',n_g)
    ;
printf('\nThe vertical force at G   = %.2f N',v_g);
printf('\nThe moment about G        = %.2f Nm',m_g);


//
    ----------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------
```

Scilab code Exa 1.5 S5

```
clear all; clc;


```

```
disp("Scilab Code Ex 1.5 :")

// Given:
f_a = 50; //N
m_a = 70; // Moment at A in Nm
l_ad = 1.25; //Length of AD in m.
l_bd = 0.5; //Length of BD in m.
l_cb = 0.75; //Length of BC in m.
w_l  = 2; //Kg/m
g = 9.81; //N/kg- acceleration due to gravity



//Free Body Diagram :

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.
w_ad = w_l*l_ad*g;

// Equations of Equilibrium

//Balancing forces in the x direction:
f_b_x = 0; // N

//Balncing forces in the y direction:
f_b_y = 0; //N

//Balncing forces in the z direction:
f_b_z = g + w_ad + f_a; //N

// Balancing Moments in the x direction:
m_b_x = - m_a + (f_a*l_bd) + (w_ad*l_bd) + (l_bd/2)*
   g; //Nm

// Balancing Moments in the y direction:
m_b_y = - (w_ad*(l_ad/2)) - (f_a*l_ad); //Nm

```

```
// Balancing Moments in the z direction:
m_b_z = 0; //Nm

v_b_shear = sqrt(f_b_z ^2 + 0); //Shear Force in N
t_b = - m_b_y; //Torsional Moment in Nm
m_b = sqrt(m_b_x ^2+ 0); // Bending moment in Nm


//Display

// Displaying results:


printf('\n\n The weight of segment BD
                    =   %.1f N',w_bd);
printf('\n The weight of segment AD                   =
    %.1f N',w_ad);
printf('\n The force at B in the Z direction          =
    %.1f N',f_b_z);
printf('\n The moment about B in the X direction      =
    %.1f Nm',m_b_x);
printf('\n The moment about G in the Y direction      =
    %.1f Nm',m_b_y);
printf('\n The Shear Force at B                       =
    %.1f N',v_b_shear);
printf('\n The Torsional Moment at B                  =
    %.1f Nm',t_b);
printf('\n The Bending Moment at B                    =
    %.1f Nm',m_b);



//
    ----------------------------------------------------------------
    END
    ----------------------------------------------------------------
```

Scilab code Exa 1.6 S6

```
clear all; clc;


disp("Scilab Code Ex 1.6 :")

//Given:
netf_b = 18*(10 ^3); //N Net force at B.
netf_c = 8*(10^3); //N Net force at C.
f_a = 12 *(10^3); //N Force at A.
f_d = 22* (10^3); //N Force at D.
w = 35; //mm Width.
t = 10; //mm Thickness.

//calculations:
p_bc = netf_b + f_a; //N Net force in region BC.
a = w*t; //m^2 The area of the cross section.
avg_normal_stress = p_bc/a; //Average Normal Stress.



// Displaying results:

printf('\n\n Net force in the region BC
                                    = %.2f N',p_bc);
printf('\nThe Area of cross section
                                      = %.2f m^2',a);
printf('\nThe Average Normal Stress in the bar when
    subjected to load = %.2f MPa',avg_normal_stress);

//
    ------------------------------------------------------------
    END
```

Scilab code Exa 1.7 S7

```
clear all; clc;


disp("Scilab Code Ex 1.7 :")

//Given :
m_lamp = 80; //Mass of lamp in Kg.
d_ab = 10; // Diameter of AB in mm.
d_bc = 8; // Diameter of BC in mm.
ab_h = 60 *(%pi/180); // In degrees - Angle made by
    AB with the horizontal.
w = m_lamp*9.81; //N
a_bc = (%pi/4)*(d_bc^2); //m^2 Area of cross section
    of rod BC
a_ab = (%pi/4)*(d_ab^2); //m^2 Area of cross section
    of rod AB



// Equations of equilibrium: Solving equilibrium
    equations simultaneously ,using matrices ,in the
    x and y directions to obtain force in BC and
    force in BA.


a = [(4/5) -(cos(ab_h)) ; (3/5) (sin(ab_h))];
b = [0 ; w];
f = zeros(1)

f = a\b;
```

```
f_bc = f(1); // Force in BC in N.
f_ba = f(2); //Force in BA in N.
avg_normal_stress_a = f_ba / a_ab; //Mpa Average
    Normal Stress in AB
avg_normal_stress_c = f_bc/ a_bc;// Mpa  Average
    Normal Stress in BC


// Displaying results:


printf('\n\nThe Weight of lamp  = %.2f N',w);
printf('\nThe Net force in BC = %.2f N',f_bc);
printf('\nTheNet force in BA  = %.2f N',f_ba);
printf('\nThe Average Normal Stress in AB when
    subjected to load = %.2f MPa',avg_normal_stress_a
    );
printf('\nThe Average Normal Stress in BC when
    subjected to load = %.2f MPa',avg_normal_stress_c
    );

//
    ________________________________________________________________
    END
    ________________________________________________________________
```

Scilab code Exa 1.8 S8

```
clear all; clc;


disp("Scilab Code Ex 1.8 :")

//Given:
```

```
h_above_ab = 0.8;
h_below_ab = 0.2;
d_a = 0.2;
d_b = 0.1;
sp_w = 80;

// Equation of Equilibrium:


a = %pi* (d_a^2);  // Area of cross section in m^2
p = sp_w * h_above_ab * a;
avg_comp_stress = p/a; // The average compressive
    stress in kN/m^2

//Display:

printf('\nThe internal Axial force P      = %.2f kN',
    p);
printf('\nThe average compressive stress = %.2f kN/m
    ^2',avg_comp_stress);


//
    ______________________________________________________________________
    END
    ______________________________________________________________________
```

Scilab code Exa 1.9 S9

```
clear all; clc;

disp("Scilab Code Ex 1.9 : ")

//Given :
```

```
f = 3000; //N Force acting at distance x from AB.
l_ac = 200; //Length of AC in mm.
a_ab = 400; //Cross sectional area of AB in mm^2.
a_c = 650; // area of C in mm^2.


f_ans = zeros(3)

k = [1 1 0;0 l_ac -f; 1.625 -1 0]
l = [f ; 0 ; 0 ]
f_ans = k\l;

f_ab = f_ans(1)
f_c = f_ans(2)
x = f_ans(3)

//Display:

printf('\n\nThe Net force on AB        = %.2f N',ceil
    (f_ab));
printf('\nNet force on C            = %.2f N',f_c);
printf('\nDistance of force from AB = %.2f mm',ceil(
    x));


//
    ------------------------------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------------------------------
```

Scilab code Exa 1.10 S10

```
clear all; clc;

```

```
disp("Scilab Code Ex 1.10 : ")

//Given:
 af = 800; //N Axial force along centroidal axis
t = 0.040; //m thickness of square cross section
ang_b = 30 *(%pi/180) ;
ang_b_comp = 60 *(%pi/180);
a = t^2; //m^2 Area of cross section
a_new = ((t*1000)^2)/(sin(ang_b_comp)); // mm^2 Area of section at b-b

//Part(a)

//Internal Loading: The bar is sectioned , Fig 1-24b, and the internal resultant loading consists of only axial force.

// Average Stress:
avg_stress = af/(a* 1000);

//Shear Force at the section is zero.
//The average normal stress distribution over the cross section is shown in Fig 1-24c.


//Part(b)


//solve the two equations for two unknowns:

N = af * cos(ang_b);
V = af * sin(ang_b);
avg_normal_stress = (N*1000)/ a_new; // kPa
avg_shear_stress = (V*1000)/a_new; //kPa

//Display

printf('\n\nThe average stress for section a-a
```

```
                  = %.2f kPa',avg_stress);
printf('\nThe Normal Force for section b-b
                      = %.2f N',N);
printf('\nThe Shear Force for section b-b
                       = %.2f N',V);
printf('\nThe Average Normal Stress for section b-b
      = %.2f kPa',avg_normal_stress);
printf('\nThe Average Shear Stress for section b-b
       = %.2f kPa',ceil(avg_shear_stress));

//
   ------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------
```

Scilab code Exa 1.11 S11

```
clear all; clc;

disp("Scilab Code Ex 1.11 : ")

//Given :
f = 5000; //N
d_rod = 10;//Diameter of steel rod in mm.
l_bc = 20; //Length of side bc in mm.
l_bd = 40; //Length of side bd in mm.
a_rod = (%pi/4)* (d_rod^2); //Area of cross section
   of the rod in mm^2.
a_strut = l_bc*l_bd ; //Area of strut in mm^2.


//Average shear stress

avg_shear_rod = f/a_rod; //for rod in Mpa
```

```
avg_shear_strut = (f/2)/a_strut; //for strut

//Display:

printf('\n\nThe average shear stress for the rod
   = %.2f MPa',avg_shear_rod);
printf('\nThe average shear stress for the strut =
   %.2f MPa',avg_shear_strut);



//
   ----------------------------------------------------------------
   END
   ----------------------------------------------------------------
```

Scilab code Exa 1.12 S12

```
clear all; clc;


disp("Scilab Code Ex 1.12 : ")

//Given:
l_bc = 50; //Length of BC in mm.
l_db = 75; // mm.
l_ed = 40; // mm.
l_ab = 25; // mm.
f_diagonal = 3000; //N
a1 = l_ab*l_ed; //Area of face AB in mm^2.
a2 = l_bc*l_ed ; //mm^2.
a3 = l_db*l_ed ; // mm^2.

//Internal loadings - The free body diagram of the
```

```
    inclined member is shown in 1-26b.

//Equilibrium Equations

//Balancing forces along the x- direction.
f_ab = f_diagonal*(3/5); //Force on segment AB in N
V = f_ab; //Shear force acting on the sectioned
    horizontal plane EDB in N

//Balancing forces along the Y direction.
f_bc = f_diagonal*(4/5); //Force on segment BC in N.

//Average compressive stresses along the horizontal
    and vertical planes:

avg_comp_ab = f_ab/a1; // N/mm^2
avg_comp_bc = f_bc/a2; // N/mm^2

//Average shear stress acting on the horizontal
    plane defined by EDB :

avg_shear = f_ab/a3; // N/mm^2

//Display:


printf('\n\nThe Force on segment AB
                           = %.2f N',f_ab);
printf('\nThe Shear Force on sectioned plane EDB    =
    %.2f N',V);
printf('\nThe Force on segment BC                    =
    %.2f N',f_bc);
printf('\nThe average compressive stress along AB   =
    %.2f N/mm^2',avg_comp_ab);
printf('\nThe average compressive stress along BC   =
    %.2f N/mm^2',avg_comp_bc);
printf('\nThe average shear stress along EDB        =
    %.2f N/mm^2',avg_shear);
```

```

//
    ________________________________________________________________________
    END
    ________________________________________________________________________
```

Scilab code Exa 1.13 S13

```

clear all; clc;


disp("Scilab Code Ex 1.13 : ")

//Given:
shear_allow = 90; //MPa
tensile_allow = 115; //MPa

l_AP = 2; //m
l_PB = 1; //m
resultant_A = 5.68; //kN
resultant_B = 6.67; //kN
v_a = 2.84; //kN
v_b = 6.67; //kN


//Diameter of the Pins:
A_A = (v_a*10^3)/(shear_allow*10^6); //Area of pin A
da = (sqrt((4*A_A)/%pi))*10^3 // d = (square root of
    (area*4/pi)) in mm
A_B = (v_b*10^3)/(shear_allow*10^6) ; //Area of pin
    B
db = (sqrt((4*A_B)/%pi))*10^3 // Area = (%pi\4)d^2
    in mm^2
```

```

chosen_da = ceil(da);
chosen_db = ceil(db);

//Diameter of Rod:
A_bc = (resultant_B*10^3)/(tensile_allow*10^6); //Area of BC
dbc = (sqrt((4*A_bc)/%pi)*10^3); // Area = %pi\4)d^2
chosen_dbc = ceil(dbc);

//Displaying Results:

printf ("\n\n The diameter of pin A  = %.3f mm",da);
printf ("\n The diameter of pin B  = %.3f mm",db);
printf ("\n The diameter of rod BC = %.2f mm",dbc);
printf ("\n\n\nThe chosen diameters are: ");
printf ("\n The diameter of pin A  = %.3f mm",chosen_da);
printf ("\n The diameter of pin B  = %.3f mm",chosen_db);
printf ("\n The diameter of rod BC = %.2f mm",chosen_dbc);

//______________________________________________________________________________ END ______________________________________________________________________________
```

Scilab code Exa 1.14 S14

```
clear all; clc;

```

```
disp("Scilab Code Ex 1.14 : ")

//Given:
shear_allow = 55; //MPa
l_ac = 200; //mm
l_cd= 75; //mm
l_de = 50; //mm
l_ce = l_cd + l_de;
load_d =15; //kN
load_e = 25; //kN

//Internal Shear Force:
//summation Mc = 0

f_ab = ((load_d*l_cd +load_e*(3/5)*l_ce)/l_ac);
c_x =-load_d + (load_e*(4/5)); //resolving C in x dir
c_y = load_d + (load_e*(3/5)); //resolving C in y dir

f_c = sqrt(c_x^2 + c_y^2); //kN
V = f_c/2;

//Required Area
A = ((V*10^3)/(shear_allow)); //A = V/Allowable shear in mm^2
d = ((sqrt((4*A)/%pi))) // Area = (%pi\4)d^2 in mm^2

chosen_d = ceil(ceil(d))+1;

//Displaying Results:


printf("\n\nThe force at AB              = %.2f kN",f_ab);
printf("\nThe resultant force at C  = %.2f kN",f_c);
printf("\nThe area of pin              = %.2f mm^2",A);
```

```
printf("\nThe diameter of pin          = %.2f mm", chosen_d);

//  ———————————————————————————— END ————————————————————————————
```

Scilab code Exa 1.15 S15

```
clear all; clc;

disp("Scilab Code Ex 1.15 : ")

//Given:
P= 20; //kN
d_hole = 40; //mm
normal_allow = 60; //MPa
shear_allow = 35; //MPa


//Diameter of Rod:
area1 = (P*10^3)/(normal_allow*10^6); //Area in m^2
d = ((sqrt((4*area1)/%pi))*1000); // Area = (%pi\4)d^2


//Thickness of disc:
V = P;
area2 = (V*10^3)/(shear_allow*10^6); //Area in m^2
thickness = (area2*10^6)/(d_hole*%pi);// A = pi*d*t


printf("\n\nThe cross sectional area of disc    = %.8
```

```
    f m^2",area1);
printf("\nThe diameter of rode                    = %.2f
    mm",d);
printf("\nThe thickness of disc                   = %.2f
    mm",thickness);

//
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 1.16 S16

```
clear all; clc;

disp("Scilab Code Ex 1.16 : ")

//Given:
bearing_allow = 75; //MPa
tensile_allow = 55; //MPa
d_shaft = 60; //mm
r_shaft = d_shaft/2; //mm
area_shaft = %pi*(r_shaft^2); //Area = pi*r^2
d_collar = 80; //mm
r_collar = d_collar/2; //mm
area_collar = %pi*(r_collar^2); //Area = pi*r^2
thick_collar = 20; //mm

//Normal Stress:
P1 = (tensile_allow* area_shaft)/3; //Tensile stress
     = 3P/A.
P1_kN = P1/1000;


```

```
//Bearing Stress:
bearing_area = area_collar-area_shaft; //mm^2
P2 = (bearing_allow*bearing_area)/3; //Bearing
    stress = 3P/A.
P2_kN= P2/1000;

if(P2_kN<P1_kN)
    big = P2_kN;
else big = P1_kN;
    end

//Displaying Results:

printf("\n\nThe load calculated by Normal Stress
                = %.1f kN",P1_kN);
printf("\nThe load calculated by Bearing Stress
                = %.1f kN",P2_kN);
printf("\nThe largest load that can be applied to
    the shaft  = %.1f kN",big);

//
    ------------------------------------------------------------------------
    END-----------------------------------------------
```

Scilab code Exa 1.17 S17

```
clear all; clc;

disp("Scilab Code Ex 1.17 : ")

//Given:
d_ac= 20; //mm
area_ac = %pi*(d_ac/2)^2; //Area = (%pi\4)d^2
area_al = 1800; //mm^2
d_pins = 18; //mm
```

```
area_pins = %pi*(d_pins/2)^2;
st_fail_stress = 680; //MPa
al_fail_stress = 70; //MPa
shear_fail_pin = 900; //MPa
fos = 2; //Factor of safety
l_ab = 2; //m
l_ap = 0.75; //m


st_allow= st_fail_stress /fos; //MPa
al_allow = al_fail_stress/fos; //MPa
pin_allow_shear = shear_fail_pin/fos; //MPa

//Rod AC
f_ac = (st_allow*area_ac)/1000;
P1 = ((f_ac*l_ab)/(l_ab-l_ap));

//Block B
f_b =(al_allow*area_al)/1000;
P2 = ((f_b*l_ab)/l_ap);

//Pin A or C:
V = (pin_allow_shear*area_pins)/1000;
P3 = (V*l_ab)/(l_ab-l_ap);

if(P1<P2 & P1<P3)
    big = P1;
else if(P2<P1 & P2<P3)
    big = P2;
else big = P3;
end

//Displaying Results:

printf("\n\nThe load allowed on rod AC
                        = %.1f kN",round(P1));
printf("\nThe load allowed on block B
                        = %.1f kN",P2);
```

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

//
   ------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------
```

Chapter 2

Strain

Scilab code Exa 2.1 Strain1

```
clear all; clc;

disp("Scilab Code Ex 2.1 : ")

//Given:
e_z= 4;
ab = 0.200; //m


//Calculations:

//Part a)

z=integrate('1+(40*10^-3)*(sqrt(z))','z',0,ab); //
   Strain formula for short line segment = delta(
   sdash) =(1+e_z)delta(s)
deltaB= z-ab;
deltaB_mm= deltaB*1000;

//Part b)

```

```
e_avg = deltaB/ab;//  Normal strain formula : e = (
    delta(sdash) -delta(s))/delta(s)

//Display:


printf("\n\nThe value of integration is
                 =%10.5f m",z);
printf("\nThe displacement at the end of the rod is
    =   %0.2f mm",deltaB_mm);
printf("\nThe average normal strain in the rod is
    =%10.4f mm/mm",e_avg);

//
    ______________________________________________________________________
    END
    ______________________________________________________________________
```

Scilab code Exa 2.2 Strain2

```
clear all; clc;

disp("Scilab Code Ex 2.2 : ")

//Given:
theta = 0.002; //radians
bc=1; //m
ba = 0.5;//m

//Calculations:

bb_dash = theta*ba;
avg_normal_strain = bb_dash/bc;//m/m

```

```
//Display:


printf("\n\nThe average normal strain =%10.3f m/m",
    avg_normal_strain);

//-----------------------------------------------END
    --------------------------------------------------------------------------------
```

Scilab code Exa 2.3 Strain3

```
clear all; clc;

disp("Scilab Code Ex 2.3 : ")

//Given:

ab= 250; //mm
bbdash_x = 3; //mm
bbdash_y = 2; //mm
ac = 300; //mm

//calculations:

//Part(a)
abdash = sqrt((ab - bbdash_y)^2 + (bbdash_x)^2);  //
    Pythagoras theorem
avg_normal_strain = (abdash-ab)/ab;

//Part(b)
gamma_xy = atan(bbdash_x/(ab - bbdash_y)); //shear
    strain formula

//Display:
```

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

//-------------------------------------------------------------------------- END --------------------------------------------------------------------------
```

Scilab code Exa 2.4 Strain4

```
clear all; clc;

disp("Scilab Code Ex 2.4 : ")

//Given:
ab = 150; //mm
bc = 150; //mm
disp_cd= 2; //mm
ab_half = ab/2;
addash_half = (bc+disp_cd)/2 ;

//Calculations:

//Part(a)

ac = sqrt((ab)^2 + (bc)^2);  //Pythagoras theorem in mm
ac_m = ac/1000; //in m
acdash = sqrt((ab)^2 + (bc+disp_cd)^2);  //Pythagoras theorem in mm
acdash_m = acdash/1000; //in m

```

```
avg_strain_ac = (acdash_m - ac_m)/ac_m; //Normal
    strain formula

//Part(b)

theta_dash = 2* atan((addash_half)/(bc/2)); //theta
    found in radians
gamma_xy = (%pi / 2)- theta_dash; //shear strain
    formula

//Display:


printf("\n\nThe average normal strain along the
    diagonal AC is    =%10.5f mm/mm",avg_strain_ac);
printf("\nThe shear strain at E relative to the x,y
    axes        = %10.5f rad",gamma_xy);

//
    ____________________________________________________________
    END
    ____________________________________________________________
```

Chapter 3

Mechanical Properties of Materials

Scilab code Exa 3.1 MPM1

```
clear all; clc;

disp("Scilab Code Ex 3.1 : ")

//Given:
offset = 0.2; //%
a_x = 0.0016; //mm/mm
a_y = 345; //Mpa

//Refer to the given graph.

//Calculations:

//Modulus of Elasticity
E = a_y/(a_x*10^3); //E is the slope in GPa.

//Yield Strength:
sigma_ys = 469; //Graphically, for a strain of 0.002
   mm/mm
```

```

//Ultimate Stress:
sigma_u = 745.2; //Mpa B is the peak of stress
    strain graph.

//Fracture Stress:
ep_f = 0.23; //mm/mm
sigma_f = 621; //Mpa from the graph.

//Display:

printf("\n\nThe Modulus of Elasticity is            =
    %10.1f GPa",E);
printf("\nThe Yield Strength from the graph      =
       %0.2f MPa",sigma_ys);
printf("\nThe Ultimate Stress from the graph is =%10
    .1f MPa",sigma_u);
printf("\nThe Fracture Stress from the graph is =%10
    .1f MPa",sigma_f);

//
    ------------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------------
```

Scilab code Exa 3.2 MPM2

```
clear all; clc;

disp("Scilab Code Ex 3.2 : ")

//Given:
stress_b = 600; //MPa
strain_b = 0.023; //mm/mm
```

```
stress_a = 450; //Mpa
strain_a = 0.006; //mm/mm

//Calculations:

//Permanent Strain:
E = stress_a/strain_a;
strain_cd = stress_b/E; //The recovered elastic
    strain
perm_strain = strain_b - strain_cd; //mm/mm

//Modulus of Resilience:
ur_initial = (0.5*stress_a*strain_a);//MJ/m^3
ur_final = (0.5*stress_b*strain_cd); //MJ/m^3

//Display:

printf("\n\nThe Permanent Strain is                =
    %10.5f mm/mm",perm_strain);
printf("\nThe Initial Modulus of Resilience is = %0
    .2f MJ/mm^3",ur_initial);
printf("\nThe Final Modulus of Resilience is   = %0
    .2f MJ/mm^3",ur_final);



//
    ____________________________________________________________________________
    END
    ____________________________________________________________________________
```

Scilab code Exa 3.3 MPM3

```
clear all; clc;

```

```
disp("Scilab Code Ex 3.3 : ")

//Given:
p = 10000; //N
E_al = 70*(10^3); //MPa
l_ab = 600; //mm
d_ab = 20; //mm
l_bc = 400; //mm
d_bc = 15; //mm

//Calculations:

a_ab = (%pi/4)*(d_ab^2);// Area of AB
a_bc = (%pi/4)*(d_bc^2);
stress_ab = p/a_ab;// Stress = load/area
stress_bc = p/a_bc;

e_ab = stress_ab/E_al; //Hookes's Law. Elastic
    strain.
e_bc = 0.045; //mm/mm . From the graph for stress_bc

elongation = (l_ab*e_ab)+ (l_bc*e_bc);
strain_rec = stress_bc/E_al; //Strain Recovery

e_og = e_bc-strain_rec;// mm/mm
rod_elong = e_og*l_bc;

//Display:

printf("\n\nThe elongation of the rod when load is
    applied              =%10.1f mm",elongation);
printf("\nThe permanent elongation of the rod when
    load is removed = %0.1f mm",rod_elong);

//
    ________________________________________________________________________________
    END
    ________________________________________________________________________________
```

Scilab code Exa 3.4 MPM4

```
clear all; clc;

disp("Scilab Code Ex 3.4 : ")

//Given:
P = 80; //kN
l_z = 1.5; //m
l_y = 0.05;//m
l_x = 0.1; //m

//Calculations:
A= l_x*l_y;
normal_stress_z = (P*(10^3))/A; //Pa

Est = 200; //GPa - from the tables.
strain_z = (normal_stress_z)/(Est*(10^9)); // Strain = stress/modulus of elasticity

axial_elong = strain_z*l_z; //elongation in the y direction

nu_st = 0.32; //Poisson's Ratio - from the tables.
strain_x = -(nu_st)*(strain_z); //strain in the x direction.
strain_y = strain_x;

//Elongations:
delta_x = strain_x*l_x;
delta_y = strain_y*l_y;

//Display:
```

```
printf("\n\nThe change in the length (z direction) = %10.8f m",axial_elong);
printf("\nThe change in the cross section (x direction)= %10.8f m',delta_x);
printf("\nThe change in the cross section (y direction)= %10.8f m',delta_y);

printf("\n\nIn the standard form:")
printf("\nThe change in the length (z direction) =  %10.2f x10^6m",(axial_elong*10^6));
printf("\nThe change in the cross section (x direction)= %10.2f x10^6m',(delta_x*10^6));
printf("\nThe change in the cross section (y direction)= %10.2f x10^6m',(delta_y*10^6));

//--------------------------------------------------------------------------------END--------------------------------------------------------------------------------
```

Scilab code Exa 3.5 MPM5

```
clear all; clc;

disp("Scilab Code Ex 3.5 : ")

//Given:
//Refer to the graph of shear stress-strain of titanium alloy.
x_A = 0.008; //rad - x co-ordinate of A
y_A = 360; //MPa - y co-ordinate of A
height = 50;//mm
l = 75; //mm
```

```
b = 100; //mm


//Calculations:

//Shear Modulus:
G = y_A/x_A;

//Proportional Limit:
tou_pl = 360; //Mpa Point A

//Ultimate Stresss:
tou_u = 504; //MPa - Max shear stress at B

//Maximum Elastic Displacement:
tanA= x_A;// tan theta is approximated as theta.
d = tanA*height;

//Shear Force:
A = l*b;
V = tou_pl*A;

//Display:


printf("\n\nThe Shear Modulus                  = %10.2
   f MPa",G);
printf("\nThe Proportional Limit           = %10.2f
   Mpa",tou_pl);
printf("\nThe Ultimate Shear Stress        = %10.2f
   MPa ",tou_u);
printf("\nThe Maximum Elastic Displacement = %10.2f
   mm",d);
printf("\nThe Shear Force                  = %10.2f
   kN ",(V/1000));

//
   ____________________________________________________________________
```

END

Scilab code Exa 3.6 MPM6

```
clear all; clc;

disp("Scilab Code Ex 3.6 : ")

//Given:
d_o = 0.025; //m
l_o =0.25; //m
F =165; //kN
delta = 1.2; //mm
G_al = 26; //GPa
sigma_y = 440; //MPa

//Calculations:

//Modulus of Elasticity:
A = (%pi/4)*(d_o^2);
avg_normal_stress = (F*10^3)/A;
avg_normal_strain = delta/l_o;
E_al = avg_normal_stress/ avg_normal_strain;

E_al = E_al/10^6;

//Contraction of Diameter:
nu = (E_al/(2*G_al))-1;
strain_lat = nu*(avg_normal_strain) ;
d_contraction = strain_lat* d_o ;


//Display:
```

```

printf("\n\nThe Modulus of Elasticity
                          = %10.1f GPa",E_al);
printf("\nThe contraction in diameter due to the
   force = %10.4f mm",d_contraction);

//
  ------------------------------------------------------------------
  END
  ------------------------------------------------------------------
```

Chapter 4

Axial Load

Scilab code Exa 4.1 AL1

```
clear all; clc;

disp("Scilab Code Ex 4.1 : ")

//Given:
a_ab = 600; //mm^2
a_bd = 1200; //mm^2
a_bc = a_bd;
p = 75; //kN
l_ab = 1; //m
l_bc = 0.75; //m
l_cd = 0.5; //m

//Calculations:

//Internal Forces: By method of Sections
P_bc = 35; //kN
P_cd = 45;//kN

//Displacement:
E_st = 210*(10^3); //From the tables
```

```
P = [p P_bc -P_cd];
A =[a_ab a_bc a_bd];
L= [l_ab l_bc l_cd];
E = []
n = length(P)

delta_sum =0;

for i = 1:n;
    delta_sum = delta_sum + (P(i)*L(i)*(10^6))/(A(i)
       *E_st);
end

delta_bc = (P_bc*l_bc*10^6)/(a_bc*E_st);



//Display:

printf("\n\nThe vertical displacement of end A      =
    +%1.2f mm",delta_sum);
printf("\nThe displacement of B relative to C is = +
   %1.3f mm",delta_bc);

//
   ____________________________________________________________
   END
   ____________________________________________________________
```

Scilab code Exa 4.2 AL2

```
clear all; clc;

```

```
disp("Scilab Code Ex 4.2 : ")

//Given:
a_ab = 400; //mm^2
d_rod = 10; //mm
r_rod = d_rod/(2*1000); //radius in m
P = 80; //kN
E_st = 200*(10^9); //Pa
E_al = 70*(10^9); //Pa
l_ab = 400; //mm
l_bc = 600; //mm

//Calculations:

//Internal forces: tension = compression = 80kN.

//Displacement:

//delta =PL/AE
numerator1 = P*(10^3)*(l_bc/1000);
denominator1 = (%pi*r_rod^2*E_st);
delta_cb = numerator1/denominator1; //to the right

numerator2 = -P*(10^3)*(l_ab/1000);
denominator2 = (a_ab* 10^-6 *E_al);
delta_a = -numerator2/denominator2; //to the right

delta_c = delta_a+delta_cb;

//Display:



printf("\n\nThe displacement of C with respect to B
        = +%1.6f m',delta_cb);
printf("\nThe displacement of B with respect to A
        = +%1.6f m",delta_a);
printf('\nThe displacement of C relative to A
```

```
            = +%1.5 f  m' , delta_c ) ;

//──────────────────────────────────────────────
    END
    ──────────────────────────────────────────────
```

Scilab code Exa 4.3 AL3

```
clear all; clc;

disp("Scilab Code Ex 4.3 : ")

//Given:
d_ac = 20; //mm
r_ac = d_ac/(2*1000); //radius in m
d_bd =40; //mm
r_bd = d_bd/(2*1000); //radius in m
P = 90; //kN
E_st = 200*(10^9); //Pa
E_al = 70*(10^9); //Pa
l_af = 200; //mm
l_fb = 400; //mm
l_bd = 300; //mm
l_ac = l_bd;

//Calculations:

//Internal Force:
P_ac = 60; //kN
P_bd = 30; //kN

//Displacement:

//Post AC: delta = PL/AE
```

```
num1 = -(P_ac*10^3*(l_ac/1000));
denom1 = %pi* r_ac^2*E_st;
delta_a = -num1/denom1; //downwards
delta_a = delta_a*1000; //in m

//Post BD: delta = PL/AE
num2 = -(P_bd*10^3*(l_bd/1000));
denom2 = %pi* r_bd^2*E_al;
delta_b = -num2/denom2; //downwards
delta_b = delta_b*1000; //in m


delta_f = delta_b + (0.184)*(l_fb/(l_af+l_fb)); //By similar triangles from the figure.

//Display:

printf('\n\nThe displacement of Post AC        = +%1.3f mm downwards',delta_a);
printf('\nThe displacement of Post BD        = +%1.3f mm downwards',delta_b);
printf('\nnThe displacement of point F       = +%1.3f mm downwards',delta_f);

//------------------------------------------------------------------------------END------------------------------------------------------------------------------
```

Scilab code Exa 4.5 AL5

```
clear all; clc;

disp("Scilab Code Ex 4.5 : ")
```

```
//Given:
d_ab = 5; //mm
A = (%pi/4)*(d_ab/1000)^2;
gap = 1; //mm
P = 20; //kN
E_st = 200; //GPa
l_ac = 0.4; //m
l_cb = 0.8; //m
l_ab = l_ac+l_cb;

//Calculations:

//Equilibrium:
//  Eqn1: -Fa - Fb +P*10^3 = 0;

//Compatibility:
delta_ba = gap/1000; //in m

delta = delta_ba*(A*E_st*10^9); //delta_ba* Lac/AE


//Eqn2: (L/AE)*Fa -(Lb/AE)*Fb = delta_ba

//Solving Equations 1 and 2 by matrices:
coeff_F = [1 1; l_ac  -l_cb];
b =[P*10^3 ; delta];
F = coeff_F\b;

F_a = F(1)/1000;
F_b = F(2)/1000;

//Display:


printf("\n\nThe reaction force at A = %1.1f kN",F_a)
    ;
printf("\nThe reaction force at B = %1.2f kN",F_b);
```

```
//
  ________________________________________________________________
  END
  ________________________________________________________________
```

Scilab code Exa 4.6 AL6

```
clear all; clc;

disp("Scilab Code Ex 4.6 : ")

//Given:
P = 45; //kN
E_al = 70*10^3;
E_br = 105*10^3;
h = 0.5; //m
ri = 25/1000; //m
ro = 50/1000; //m
A = (%pi*(ro^2 -ri^2));
Ai = %pi*ri^2;

//Calculations:

//Equilibrium: Eqn1:F_al +F_br = P

//Compatibility:
coeff_F_br = (A*E_al)/(Ai*E_br); // delta_al =
    delta_brass

//Eqn2 : F_al- (coeff_F_br*F_br) = 0

//Solving equations 1 and 2 using matrices:

```

```
coeff_F = [1 1; 1 -coeff_F_br];
b = [P; 0];
F = coeff_F\b;

F_al =F(1);
F_br =F(2);

avg_stress_al = F_al/A;
avg_stress_br = F_br/Ai;

avg_stress_al = avg_stress_al/1000;
avg_stress_br = avg_stress_br/1000;

//Display:


printf("\n\nThe axial force experienced by Al      =
   %1.1f kN",F_al);
printf("\nThe axial force experienced by Brass  = %1
   .2f kN",F_br);
printf('\nThe average normal stress in Al       = %1
   .2f MPa',avg_stress_al);
printf('\nThe average normal stress in Al Brass = %1
   .2f MPa',avg_stress_br);

//
   ----------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------
```

Scilab code Exa 4.7 AL7

```
clear all; clc;

```

```
disp("Scilab Code Ex 4.7 : ")

//Given:
P = 15; //kN
a_ab = 25; //mm^2
a_ef =a_ab;
a_cd = 15; //mm^2
l_ef = 0.5; //m
l_ce = 0.4; //m
l_ac = 0.4; //m

//Calculations:

//Equilibrium:
//In the y direction ;    F_a +F_c +F_e = P
//of moments: -F_a(l_ac)+ P(l_ac/2) +F_e(l_ce) = 0

//Compatibility equation for displacemnts:
coeff_Fc = (1/a_cd); //coefficient of Fc
coeff_Fa = (0.5/a_ab); //coefficient of Fc
coeff_Fe = (0.5/a_ef); //coefficient of Fc

//Using matrices to solve the 3 Equations:
A = [1 1 1; -l_ac 0 l_ce; coeff_Fa -coeff_Fc coeff_Fe];
b = [P ; -P*(l_ac/2); 0];
F = A\b;


F_a = F(1);
F_b = F(2);
F_c = F(3);

//Display:


printf("\n\nThe force in rod AB        = %1.2f kN', F_a);
```

```
printf('\nThe force in rod CD          = %1.2f kN',F_b);
printf('\nThe force in rod EF          = %1.2f kN',F_c);

//---------------------------------------------------------------------------------- END ----------------------------------------------------------------------------------
```

Scilab code Exa 4.8 AL8

```
clear all; clc;

disp("Scilab Code Ex 4.8 : ")

//Given:
r_o = 10; //mm
r_i = 5; //mm
l = 60; //mm
a_t = (%pi)*(r_o^2 - r_i^2); //Area of thread
a_b = (%pi*(r_i^2));// Area of bolt
one_turn =20/20;
E_am = 45; //GPa
E_al = 75; //GPa

//calculations:

//Equilibrium:
// In Y direction: F_b - F_t = 0

//Compatibility:
half_turn = one_turn/2;
coeff_Ft = l/(a_t*E_am*10^3); // delta = PL/AE
coeff_Fb = l/(a_b*E_al*10^3);
```

```

//Solving the two simultaneous equations for F_b and
    F_t:
A = [1 -1; coeff_Fb coeff_Ft];
b = [0 ; half_turn];
F = A\b;

F_b =F(1);
F_t = F(2);

stress_b = F_b/a_b;
stress_t = F_t/a_t;

F_b = F_b/1000; //in kN
F_t = F_t/1000; //in kN

//Display:


printf('\n\nThe force experienced by threads        =
    %1.2f kN',F_t);
printf('\nThe force experienced by the bolt       =
   %1.2f kN',F_b);
printf('\nThe stress in the screw                 =
   %1.1f MPa',stress_t);
printf('\nThe stress in the bolt                  =
   %1.1f MPa',stress_b);

//
   ________________________________________________________________________
   END
   ________________________________________________________________________
```

Scilab code Exa 4.9 AL9

```
clear all; clc;

disp("Scilab Code Ex 4.9 : ")

//Given:
l_ab = 800 + 400;//mm
P = 20; //kN
d = 5/1000; //m
area = (%pi/4)*d^2; //Cross sectional area
l_bbdash = 1/1000;//m
E = 200; //GPa

//Calculations:

//Compatibility
delta_p = (P*10^3*0.4)/(area*E*10^9); //delta = PL/
    AE
delta_b = delta_p-l_bbdash;
F_b = (delta_b*area*E*10^9)/(l_ab/1000);
F_b = F_b/1000;

//Equilibrium:
F_a = P - F_b;

//Display:

printf("\n\nThe reaction at A       = %1.1f kN',F_a);
 printf('\nThe reaction at B       = %1.1f kN',F_b);

//--------------------------------------------------------------
    END
    ----------------------------------------------------------------------
```

Scilab code Exa 4.10 AL10

```
clear all; clc;

disp("Scilab Code Ex 4.10 : ")

//Given:
T1 = 30; //degree celcius
T2 = 60;//degress celcius
l_ab = 1;//m
area = 10*10*10^-6; //m^2
alpha = 12*10^-6;// per degree celcius
E = 200*10^6; //kPa

//Equilibrium:
//F_a = F_b = F

del_T = T2-T1;
F = alpha*del_T*area*E; //Thermal Stress Formula

avg_normal_comp_stress = (F*10^-3)/area; // sigma =
    F/A

//Display:

printf("\n\nThe force at A and B                    =
    %1.1f kN",F);
printf('\nThe average normal compressive stress  =
    %1.1f MPa',avg_normal_comp_stress);


//
    ____________________________________________________________
    END
    ____________________________________________________________
```

Scilab code Exa 4.11 AL11

```
clear all; clc;

disp("Scilab Code Ex 4.11 : ")

//Given:
area_sleeve = 600*10^-6; //m^2
area_bolt = 400*10^-6; //m^2
T1 = 15; //degree celcius
T2 = 80; //degree celcius
alpha_bolt = 12*10^-6; //per degree celcius
alpha_sleeve = 23*10^-6; //per degree celcius
l = 0.15; //m
E_bolt = 200*10^9; //N/m^2
E_sleeve = 73.1*10^9; //N/m^2

//Equilibrium:
//F_s = F_b

//Compatibility:
del_T = T2 - T1; // temperature difference
delb_T = alpha_bolt*del_T*l;
delb_F = l/(area_bolt*E_bolt);
dels_T = alpha_sleeve*del_T*l;
dels_F = l/(area_sleeve*E_sleeve);

//delb_T + F_b*delb_F = dels_T + F_s*dels_F

F_b = (dels_T-delb_T)/(delb_F+dels_F);
F_b = F_b/1000; //in kN
F_s= F_b;

sigma_b = F_b/(area_bolt*10^3); //Average Normal Stress
sigma_s = F_s/(area_sleeve*10^3); //Average Normal Stress

```

```
//Display:


printf("\n\nThe force experienced by sleeve and bolt
          = %1.2f kN",F_s);
printf('\nThe average normal stress on bolt
                = %1.1f MPa',sigma_b);
printf('\nThe average normal stress on sleeve
               = %1.1f MPa',sigma_s);



//
  ------------------------------------------------------------
  END
  ------------------------------------------------------------------
```

Scilab code Exa 4.12 AL12

```

clear all; clc;

disp("Scilab Code Ex 4.12 : ")

//Given:
h = 0.250; //m
T1 = 20; //degree celcius
udl = 150; //kN/m
T2 = 80; //degree celcius
len = 0.3; //m
dia_steel = 0.04;//m
r_steel = 0.02;
dia_aluminium = 0.06; //m
r_al = dia_aluminium/2;
area_st = %pi*(r_steel^2);
```

```
area_al = %pi*(r_al^2);
F = 90*10^3;//N
alpha_st = 12*10^-6; //per degree celcius
alpha_al = 23*10^-6; //per degree celcius
E_st = 200*10^9; // N/m^2
E_al = 73.1*10^9; // N/m^2

//Equilibrium:
//From the free body diagram: Eqn1 : 2F_st + F_al-


// -delst_T + F_st*delst_F = -delal_T + F_al*delal_F

//Eqn2 : 165.9*10^3  =1.216F_al - F_st F = 0

//Compatibility:
delst_T = alpha_st*(T1+T2)*h;
delst_F = h/(area_st*E_st);
delal_T = alpha_al*(T1+T2)*h;
delst_F = h/(area_al*E_al);

coeffMat = [2 1; -1 1.216]
b= [90*10^3 ; 165.9*10^3]
F = coeffMat\b;
F_st = F(1)/1000;
F_al =F(2)/1000;
F_al =ceil(F_al);

//Display:


printf("\n\nThe force on the steel post              =
    %1.1f kN",F_st);
printf('\nThe force on the aluminium post          =
   %1.1f kN',F_al);

//
   --------------------------------------------------------------------------------
```

END

Scilab code Exa 4.13 AL13

```
clear all; clc;

disp("Scilab Code Ex 4.13 : ")

//Given:
sigma_allow = 115; //MPa

//Determinng the stress concentration factor:

r_n =10/20;
w_h = 40/20;
k = 1.4; //from graph
sigma_avg = sigma_allow/k;
P =sigma_avg*20*10;
P = P/1000;

//Display:

printf("\n\nThe largest axial force that the bar can
    carry       = %1.2f kN",P);

//
---------------------------------------------------------------------------
```

END

Scilab code Exa 4.14 AL14

```
clear all; clc;

disp("Scilab Code Ex 4.14 : ")

//Given:
P = 80*10^3; //N
yield_stress = 700; //MPa;
E = 200*10^9; //N/mm^2
l1 = 0.3; //m
l2 = 0.8; //m

//Maximum Normal Stress:
r_h = 6/20;
w_h = 40/20;
K = 1.6;

area2 = 0.02*0.01; //m^2 note its not 0.001.
max_stress = (K*P)/area2;
max_stress = (max_stress/10^6); // converting to MPa

//Displacement:
area1 = 0.04*0.01;
del_ad_1 = (P*l1)/(area1*E);
del_ad_2 = (P*l2)/(area2*E);
del_ad = (2*del_ad_1)+ del_ad_2;
del_ad = del_ad*1000; //converting m to mm

//Display:


printf("\n\nThe maximum normal stress                              = %1.1f MPa",
    max_stress);
printf('\nThe displacement of one end with respect
    to the other  = %1.2f mm',del_ad);
```

Scilab code Exa 4.15 AL15

```
clear all; clc;

disp("Scilab Code Ex 4.15 : ")

//Given
weight = 15; //kN
l_ab = 5; //m
l_ac= 5.0075; //m
area = 30; //mm^2

//calculations:
strain_ab = (l_ac-l_ab)/l_ab;
max_strain = 0.0017;

stress_ab = (350*strain_ab)/max_strain;
F_ab = stress_ab*area; // F= stress*area
E_st = 350/max_strain; //Modulus ofelasticity

del1 = l_ab/(area*10^-6*E_st*10^3); //del = PL/AE
del2 = l_ac/(area*10^-6*E_st*10^3); //del = PL/AE

//Eqn1 = T_ab + T_ac = weight
//Eqn2 = del1*T_ab - del2*T_ac = (l_ac-l_ab)

//Solving using matrices:
A = [1 1;del1 -del2];
b = [weight; (l_ac-l_ab)];
T = A\b;

T_ab = T(1);
T_ac = T(2);

```

```
stress_in_ab = (T_ab*10^3)/area;

if(stress_in_ab>350)
    T_ab = (350*area)/1000;
end

T_ac = 15-T_ab;
stress = (T_ac*10^3)/area;
strain_ac = (stress*max_strain)/350;

elong_ac = strain_ac*l_ac; //m
elong_ab = (l_ac-l_ab)+elong_ac; //m



//Display:

printf('\n\nThe force experienced by wire AB  = %1.1f kN',T_ab);
printf('\nThe force experienced by wire AC  = %1.1f kN',T_ac);
printf('\nThe elongation in wire AB         = %1.5f m',elong_ab);
printf('\nThe elongation in wire AC         = %1.5f m',elong_ac);

//------------------------------------------------------------------------END------------------------------------------------------------------------
```

Scilab code Exa 4.16 AL16

```
clear all; clc;
```

```
disp("Scilab Code Ex 4.16 : ")

//Given:
yield = 250; //MPa
r = 4; //mm
width = 40; //mm
thick = 2; //mm

//a)
r_h = r/(width - (2*r));
w_h = width/(width - (2*r));
K = 1.75;
area = (thick*(width - (2*r))*10^-6);
P_y = (yield*10^6*area)/K;
P_y = P_y/1000;

//b)
P_p = (yield*10^6*area);
P_p = P_p/1000;

//Display:

printf("\n\nThe maximum load P that does not cause
    the steel to yield     = %1.2f kN",P_y);
printf('\nThe maximum load that the bar can support
                            = %1.2f kN',P_p);

//
    ____________________________________________________________
    END
    ____________________________________________________________
```

Scilab code Exa 4.17 AL17

```
clear all; clc;

disp("Scilab Code Ex 4.17 : ")

//Given:
r = 5/1000; //m
yield = 420; //MPa
E = 70; //GPa
P = 60; //kN
l_ac = 100/1000; //m
l_cb = 300/1000; //m
F_a = 45; //kN by elastic analysis
F_b = 15; //kN by elastic analysis

//Calculations:
area = %pi*(r^2)
sigma_ac = F_a/(area*1000);
sigma_ac1 = sigma_ac;
sigma_cb = F_b/(area*1000);
sigma_cb1 = sigma_cb;

if(sigma_ac>yield)
    F_a_y = yield*10^3*area;
    F_b = P - F_a_y;

    sigma_ac = yield;
    sigma_cb = F_b/(area*1000);
end

//Residual Stress:
defl_c = (F_b*l_cb)/(area*E*10^6);
strain_cb = defl_c/l_cb;
strain_ac = -defl_c/l_ac;

sigma_ac_r = -sigma_ac+ sigma_ac1;
sigma_cb_r = sigma_cb - sigma_cb1;

    sigma = sigma_cb_r;
```

```
//Permanent Displacement:
res_strain_cb = (sigma*10^6)/(E*10^9);
perm_defl_c = res_strain_cb*l_cb*1000;


//Display:

printf("\n\nThe residual stress in AC                    = %1.1f MPa",sigma_ac_r);
printf("\nThe residual stress in CB                    = %1.1f MPa",sigma_cb_r);
printf("\nThe permanent displacement of the collar at C       = %1.3f mm",perm_defl_c);

//_______________________________________________________________END_______________________________________________________________
```

Chapter 5

Torsion

Scilab code Exa 5.1 T1

```
clear all; clc;

disp("Scilab Code Ex 5.1 : ")

//Given:
r = 50; //mm
J = (%pi/2)*(r^4); //polar moment of inertia
tou_max = 56; //MPa
T = (tou_max*J)/(r*10^6); //toumax = Tc/J

//Display:


printf("\n\nThe resultant internal torque    = %1.0f
    kNm',T);

//---------------------------------------------------------------------------------
    END
    ---------------------------------------------------------------------------------
```

Scilab code Exa 5.3 T3

```
clear all; clc;

disp("Scilab Code Ex 5.3 : ")

//Given:
T1 = 4250; //kNmm
T2 = -3000; //kNm
T3 = T1+T2; //kNm
r = 75; //mm

//Section Property:
J = (%pi/2)*(r^4); //polar moment of inertia

//Shear Stress:
c_a = 75; //mm
tou_a = (T3*c_a*1000)/J; //tou = Tc/J

c_b = 15; //mm
tou_b = (T3*c_b*1000)/J; //tou = Tc/J

//Display:

printf('\n\nThe shear stress developed at A   = %1.2f MPa',tou_a);
printf('\nThe shear stress developed at B   = %1.3f MPa',tou_b);

//
    ________________________________________________________________
    END
    ________________________________________________________________
```

Scilab code Exa 5.4 T4

```
clear all; clc;

disp("Scilab Code Ex 5.4 : ")

//Given:
di = 80; //mm
ri = 40/1000; //m
d0 = 100; //mm
ro = d0/2000; //m
F = 80; //N
l1 = 0.2; //m
l2 = 0.3; //m

//Internal Torque:
T = F*(l1+l2);

//Section Property:
J = (%pi/2)*((ro^4)-(ri^4));

//Shear Stress:
c_o = 0.05;//m
tou_o = (T*c_o)/(J*10^6);

c_i = 0.04; //m
tou_i = (T*c_i)/(J*10^6);

//Display:


printf('\n\nThe shear stress in the inner wall     =
    %1.3f MPa',tou_i);
printf('\nThe shear stress in the outer wall     = %1
```

```
.3f MPa',tou_o);


//
______________________________________________________________________________
END
______________________________________________________________________________
```

Scilab code Exa 5.5 T5

```
clear all; clc;

disp("Scilab Code Ex 5.5 : ")

//Given:
P = 3750; //W
N = 175; //rpm
allow_shear = 100; //MPa

//Calculations:
ang_vel = (2*%pi*N)/60; // rad/s
T = P/ang_vel; //P = T*angular velocity

c = ((2*T*1000)/(%pi*allow_shear))^(1/3);
d = round(2*c);

//Display:


printf('\n\nThe required diameter of the shaft   =
    %1.0f mm',d);

//
______________________________________________________________________________
```

END

Scilab code Exa 5.6 T6

```
clear all; clc;

disp("Scilab Code Ex 5.6 : ")

//Given:
di = 30; //mm
ri= (di/2000); //m
d0 = 42; //mm
ro = (d0/2000); //m
P = 90; //kW
max_shear = 50; //MPa

//Calculations:
c = ro; //m
J = (%pi/2)*((ro^4)-(ri^4)); //Polar moment of
    inertia of hollow shaft
T = (max_shear*J)/c; //tou max = Tc/J

//P = 2(%pi)fT
f = (P)/(2*%pi*T*10^3);

//Display:


printf('\n\nThe required frequency of rotation of
    the shaft    = %1.1f Hz',f);

//
```

END

Scilab code Exa 5.7 T7

```
clear all; clc;

disp("Scilab Code Ex 5.7 : ")

//Given:
E = 80*10^3; //MPa
d = 14/1000; //m
r = d/2; //m
R = 100; //mm
l_ac = 0.4; //m
l_cd = 0.3; //m
l_de = 0.5; //m
T_c = 280; //Nm
T_a = 150; //Nm
T_d = 40; //Nm
T_ac = T_a; //Nm
T_cd = T_ac - T_c;
T_de = T_cd - T_d;

//Angle of Twist:
J = (%pi/2)*(r^4);

T = [T_ac T_cd T_de];
l = [l_ac l_cd l_de];

sumTwist = 0;

for i= 1:3
    sumTwist = sumTwist+ ((T(i)*l(i))/(J*E*10^6));
```

```
end

displacement = - sumTwist*R;

//Display:


printf('\n\nThe angle of twist of the shaft
            = %1.3f rad',sumTwist);
printf('\nThe displacement of tooth P on gear A     =
    %1.1f mm',displacement);

//
    ------------------------------------------------------------------
    END
    ------------------------------------------------------------------
```

Scilab code Exa 5.8 T8

```
clear all; clc;

disp("Scilab Code Ex 5.8 : ")

//Given:
T = 45; //N
G = 80; //GPa
d = 20/1000; //m
r = d/2; //m
l_dc = 1.5; //m
l_ab = 2; //m
r1 = 75/1000; //m
r2 = 150/1000; //m

//Internal Torque:
```

```
F = T/r2;
T_d_x = F*r1;

//Angle of twist:
J = (%pi/2)*(r^4);
phi_c = (T*l_dc)/(2*J*G*10^9);
phi_b = (phi_c*r1)/r2;

phi_ab = (T*l_ab)/(J*G*10^9);

phi_a = phi_b + phi_ab;

//Display:


printf('\n\nThe angle of twist of end A of shaft AB
     = + %1.4f rad',phi_a);

//
    ____________________________________________________________________________
    END
    ____________________________________________________________________________
```

Scilab code Exa 5.9 T9

```
clear all; clc;

disp("Scilab Code Ex 5.9 : ")

//Given:
d = 50; //mm
r = d/2;
c = d/2;
l_buried = 600; //mm
```

```
G = 40*10^3; //MPa
F = 100; //N
l_handle= 150; //mm
l_ab = 900; //mm

//Internal Torque:
T_ab = F*2*l_handle;
t = T_ab/l_buried;

//Maximum Shear Stress:
J = (%pi/2)*(r^4);
tou_max = (T_ab*c)/(J);

//Angle of Twist:

x0=0;
x1=l_buried;
X=integrate('x','x',x0,x1);

phi_a = ((T_ab*l_ab)+(50*X))/(J*G);

//Display:



printf('\n\nThe maximum shear stress in the post
            = %1.2f N/mm^2',tou_max);
printf('\nThe angle of twist at the top of the post
      = %1.5f rad',phi_a);

//
    ________________________________________________________________________
    END
    ________________________________________________________________________
```

Scilab code Exa 5.11 T11

```
clear all; clc;

disp("Scilab Code Ex 5.11 : ")

//Given:
d = 20/1000; //m
r = d/2;
l_bc = 0.2;
l_cd = 1.5;
l_da = 0.3;
T_c = 800; //Nm
T_d = -500; //Nm

//Equilibrium:
//Eqn 1 :      300 = T_a + T_b

//Compatibility:
//Eqn 2:
coeff_Tb = -l_bc;
coeff_Ta = l_cd + l_da;

//Solving Equations simultaneously using matrices:
C = [1 1; coeff_Tb coeff_Ta];
b = [300 ; -750];
T = C\b;

T_b = T(1);
T_a = T(2);

//Display:


```

```
printf('\n\nThe reaction at A    = %1.0f Nm',T_a);
printf('\nThe reaction at B    = %1.0f Nm',T_b);

//
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 5.12 T12

```
clear all; clc;

disp("Scilab Code Ex 5.12 : ")

//Given:
 T = 250; //Nm
 G_st = 80; //GPa
 G_br = 36; //GPa
 ri = 10; //mm
 ro = 20; //mm
 l_ab = 1.2; //m

 //Equilibrium:
 // -Tst-Tbr+250Nm = 0
 coeff1_st = -1;
 coeff1_br = -1;
 b1 = -250;

 //Compatibility:
 //phi = TL/JG

 J1 = (%pi/2)*(ro^4 - ri^4);
 J2 = (%pi/2)*(ri^4);
 coeff2_st = 1/(J1*G_st*10^3);
```

```
coeff2_br = -1/(J2*G_br*10^3);
b2 = 0;

//Solving the above two equations simultaneously
    using matrices:
A = [coeff1_st coeff1_br;coeff2_st coeff2_br ];
b = [b1 ; b2];
T = A\b;

T_st = T(1);
T_br = T(2);

shear_br_max = (T_br*10^3*ri)/(J2); //tou = (Tr)/J
shear_st_min = (T_st*10^3*ri)/(J1); //tou = (Tr)/J
shear_st_max = (T_st*10^3*ro)/(J1); //tou = (Tr)/J

shear_strain = shear_br_max / G_br;
shear_strain = shear_strain;

//Display:


printf('\n\nThe Torque acting on Steel
                        = %1.2f Nm',T_st);
printf('\nThe Torque acting on Brass
                        = %1.2f Nm',T_br);
printf('\nThe maximum shear stress experienced by
    Steel     = %1.2f MPa',shear_st_max);
printf('\nThe minimum shear stress experienced by
    Steel     = %1.2f MPa',shear_st_min);
printf('\nThe maximum shear stress experienced by
    Brass     = %1.2f MPa',shear_br_max);
printf('\nThe shear strain at the interface
                    = %1.5f *10^-3 rad',shear_strain);


//
    ----------------------------------------------------------------
```

END

Scilab code Exa 5.13 T13

```
clear all; clc;

disp("Scilab Code Ex 5.13 : ")

//Given:
 l = 1.2; //m
 a = 40; //mm
 tou_allow = 56; //MPa
 phi_allow = 0.02; //rad
 G = 26; //GPa
 alpha = (60*%pi)/180; //degrees

 //Calculations:
 T_shear1 = (tou_allow*a^3)/(20*1000); // allowable shear stress = (20T)/(a^3)
 T_twist1 = (phi_allow*a^4*G*10^3)/(46*l*10^6); // angle of twist =(46TL)/(a^4*G)

 T1 = min(T_shear1, T_twist1);

//Circular Cross Section:
c_ = (a*a*sin(alpha))/(%pi*2);
c = sqrt(c_);

J = (%pi/2)*(c^4);
T_shear2 = (tou_allow*J)/(c*1000);
T_twist2 = (phi_allow*J*G*10^3)/(l*10^6);

 T2 = min(T_shear2, T_twist2);
```

```


//Display:

printf('\n\nThe largest torque that can be applied
    at the end of the triangular shaft    = %1.2f Nm'
    ,T1);
printf('\nThe largest torque that can be applied at
    the end of the circular shaft      = %1.2f Nm',T2
    );


//
    ____________________________________________________________________________
    END
    ____________________________________________________________________________
```

Scilab code Exa 5.15 T15

```
clear all; clc;

disp("Scilab Code Ex 5.15 : ")

//Given:
l_cd = 0.5; //m
l_de = 1.5; //m
h =60/1000; //m
w = 40/1000; //m
t_h = 3/1000; //m
t_w = 5/1000; //m
T_c = 60; //Nm
T_d = 25; //Nm
G = 38*10^9; //N/m^2
T1 = T_c - T_d;
```

```

//Average Shear Stress:
area = (w-t_w)*(h-t_h);

shear_a = T1/(2*t_w*area*10^6);
shear_b = T1/(2*t_h*area*10^6);

//Angle of Twist:

ds_t = 2*(((w-t_w)/t_h)+((h-t_h)/t_w));
T = [T_c T1];
l = [l_cd l_de];
phi = 0;

for i = 1:2
    phi = phi+ (T(i)*l(i)*ds_t)/(4*area^2*G);

end

//Display:

printf('\n\nThe average shear stress of the tube at
    A     = %1.2f MPa',shear_a);
printf('\nThe average shear stress of the tube at B
       = %1.2f MPa',shear_b);
printf('\nThe angle of twist of end C
                              = %1.6f rad',phi);

//
    ________________________________________________________________________________
    END
    ________________________________________________________________________________
```

Scilab code Exa 5.16 T16

```
clear all; clc;

disp("Scilab Code Ex 5.16 : ")

//Given:
T = 85; //Nm
G = 26; //GPa
t = 10; //mm thickness
a = 60; //mm side
l = 1.5; //m

//Average Shear Stress:
area_m = (a-t)*(a-t);
avg_shear = (T*10^3)/(2*t*area_m); //tou_avg = T/(2 tarea_m);


//Angle of Twist:
ds_t = (4*(a-t))/t;
phi = (T*10^3*l*10^3*ds_t)/(4*(area_m^2)*G*10^3);

//Display:


printf('\n\nThe average shear stress in the tube at A     = %1.1f N/mm^2',avg_shear);
printf('\nThe angle of twist due to loading                = %1.5f rad',phi);

//
________________________________________________________________________________ END ________________________________________________________________________________
```

Scilab code Exa 5.17 T17

```
clear all; clc;

disp("Scilab Code Ex 5.17 : ")

//Given:
tou_allow = 90; //MPa
phi_allow = 2*10^-3; //rad
a = 200; //mm side
angle = (60*%pi)/180;
h = a*sin(angle);
l = 3; //m
t = 5/1000; //m
G = 75*10^9; //N/mm^2

//Calculations:
area_m = 0.5*a*h*10^-6;//m^2 a = (1/2)bh
ds_t = (3*a)/(t*1000);

T_shear = (tou_allow*10^6*2*t*area_m); //tou_avg = T
    /(2tarea_m);

T_twist = (phi_allow*4*area_m^2*G)/(l*ds_t);

 T = min(T_shear, T_twist);


//Display:


printf('\n\nThe maximum torque that the thin tube
    can be subjected to     = %1.1f Nm',T);

//
    ----------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------
```

Scilab code Exa 5.18 T18

```
clear all; clc;

disp("Scilab Code Ex 5.18 : ")

//Given:
fillet_r = 6; //mm
D = 40/1000; //m
d = 20/1000; //m
T = 30; //Nm
D_d = D/d;
r_d = fillet_r/d;
k = 1.3;

//Maximum Shear Stress:
c = D/2;
J = (%pi/2)*(c^4)
max_shear = (k*T*c)/(J*10^6); // tou = K(Tc/J)

//Display:

printf('\n\nThe maximum shear stress in the shaft
   due to the applied torques     = %1.2f MPa',
   max_shear);

//
   ________________________________________________________________
   END
   ________________________________________________________________
```

Scilab code Exa 5.19 T19

```
clear all; clc;

disp("Scilab Code Ex 5.19 : ")

//Given:
ro = 50/1000; //m
ri = 30/1000; //m
c = ro;
shear = 20*10^6; //N/m^2

//Maximum Elastic Torque:
J = (%pi/2)*((ro^4)-(ri^4));
T_y = (shear*J)/c; // tou = Tc/J
T_y = T_y/1000; //in kN

//Plastic Torque:
x0 = 0.03;
x1 = 0.05;
I = integrate('rho^2','rho',x0,x1)
Tp =  (2*%pi*I*shear);
Tp= Tp/1000;

//Outer Shear Strain:
strain = (0.286*10^-3*ro)/(ri);

//Display:


printf('\n\nThe maximum torque that can be applied
    to the shaft without causing the material to
    yield   = %1.2f kNm',T_y);
printf('\nThe plastic torque that can be applied to
```

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


//------------------------------------------------------------ END ------------------------------------------------------------
```

Scilab code Exa 5.20 T20

```
clear all; clc;

disp("Scilab Code Ex 5.19 : ")

//Given:
r = 20/1000; //m
l = 1.5; //m
phi = 0.6; //rad
shear_y = 75*10^6; //N/m^2

//Calculations:
max_shear_strain = (phi*r)/(l); //phi = (strain*L)/r
strain_y = 0.0016;

r_y = (r*strain_y)/(max_shear_strain); //by ratios

//T= (%pi*shear_y)*(4c^3 - r_y^3)/6;
c = r;
```

```
T = (%pi*shear_y)*(4*c^3 - r_y^3)/6;
T = T/1000;

//Display:

printf('\n\nThe torque needed to twist the shaft by
    0.6 rad      = %1.2f kNm',T);

//
    ----------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------
```

Scilab code Exa 5.21 T21

```
clear all; clc;

disp("Scilab Code Ex 5.21 : ")

//Given:
l = 1.5; //m
G = 42*10^3; //GPa
co = 50; //mm
ci = 25;//mm
shear_y = 84; //N/mm^2
strain_y = 0.002; //rad

//Plastic Torque:
T_p = ((2*%pi)*(co^3 - ci^3)*shear_y)/3;
phi_p = (strain_y*l*10^3)/ci;

J = (%pi/2)*(co^4 - ci^4);
shear_r = (T_p*co)/J;
```

```
shear_i = (shear_r*ci)/(co);// shear = Tc/J

G = shear_y/strain_y;

phi_dash = (T_p*l*10^3)/(J*G); //phi = TpL/JG;

phi = phi_p - phi_dash;
T_p = T_p/10^6;

//Display:


printf('\n\nThe plastic torque Tp
                                    = %1.2f x 10^6
    Nmm',T_p);
printf('\nThe permanent twist of the tube if Tp is
    removed      = %1.5f rad',phi);


//
    ---------------------------------------------------------------------------
    END
    ---------------------------------------------------------------------------
```

Chapter 6

Bending

Scilab code Exa 6.5 B5

```
clear all; clc;

disp("Scilab Code Ex 6.5 : ")

//Shear and Moment Diagrams:
p = [-1/9 -2 30]
x = roots(p)
y = (x(2));


   M = (30*y) - (y^2) - (y^3)/27;




//Display:

printf("\n\nThe magnitude of the maximum moment is =
    %1.0f kNm', M);
printf('\nRefer to the shear and moment diagrams in
   the book.');

```

```
//----------------------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------------------
```

Scilab code Exa 6.11 B11

```
clear all; clc;

disp("Scilab Code Ex 6.11 : ")

//Given:
l = 4.5; //m
R1 = 1.5; //kN
R2 = 3; //kN
uvl = 2; //kN/m

//Shear diagram:
x = sqrt((2*R1*l)/(uvl));
M = (R1*x) - (0.5*uvl*x^3)/(3*l);

//Display:


 printf('\n\nV becomes zero at x = %1.1fm',x);
 printf('\nThe magnitude of the maximum moment = %1
    .1f kNm',M);

//
    ----------------------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------------------
```

Scilab code Exa 6.13 B13

```
clear all; clc;

disp("Scilab Code Ex 6.13 : ")

//Given:
l_ab = 4; //m
l_cd = 4; //m
l_bc = 6; //m
Rb = 8; //kN
uvl = 2; //kN/m

//Moment diagram:
p = [-1/18 0 -3.6 17.6]
x = roots(p)
y = x(3);

//Display:

 printf('\n\nV becomes zero at x = %1.2f m',y);


//
    ____________________________________________________________
    END
    ____________________________________________________________
```

Scilab code Exa 6.14 B14

```
clear all; clc;
```

```

disp("Scilab Code Ex 6.14 : ")

//Given:
b = 60; //mm
h = 120; //mm
sigma_max = 20; //N/mm^2
c = b;

//Part (a):
I = (1/12)*b*h^3;
M1 = (sigma_max*I)/(c); //sigma_max = Mc/I Flexure Formula
M1 = M1*10^-6; //in kN/m

//Part (b):
y0=60;
y1=-60

M2 = integrate('-(20*y^2)','y',y0,y1);
M2 = M2*10^-6;

F = (0.5*sigma_max*b*b);
c = 2*(60 -(0.5*b)); //distance between centroids of both the volumes.
M = F*c/1000;

//Display:

 printf("\n\nThe internal moment M calculated using : ");
 printf('\na)The flexure formula = %1.2f kNm',M1);
 printf('\nb)The resultant of the stress distribution using the basic principles = %1.2f kNm',M2);


//
```

END

Scilab code Exa 6.15 B15

```
clear all; clc;

disp("Scilab Code Ex 6.15 : ")

//Given:
udl = 5; //kN/m
l1 = 3;//m
l2 = 6; //m
t = 20/1000; //mm
yb = 0.15;//m

//Section Property:
I_bar1 = (1/12)*(0.25)*(0.02^3);
Ad2 = (0.25)*(0.02)*(yb+(t/2))^2;
I_bar2 = (1/12)*(0.02)*(0.3^3);
I = 2*(I_bar1 + Ad2) + I_bar2;

//Bending stress:
c = 0.15 + t;
M= 22.5; //kNm

sigma_max = (M*c)/(I*1000);

sigma_B = (M*yb)/(I*1000);

//Display:

 printf('\n\nThe absolute maximum bending stress is
```

```
= %1.1 f  MPa',sigma_max);


//
    ____________________________________________________________________________________
    END
    ____________________________________________________________________________________
```

Scilab code Exa 6.16 B16

```
clear all; clc;

disp("Scilab Code Ex 6.16 : ")

//Given:
t1 = 15/1000; //m
t2 = 20/1000; //m
l = 250/1000; //m
b = 200/1000; //m
P = 2.4; //kN
l_a = 2; //m
l_b = 1; //m

//Internal Moment:
y1 = b/2;
y2 = t2/2;
A = (2*t1*b)+(t2*l);
y_bar = ((2*y1*t1*b)+(y2*t2*l))/A;

M = (P*l_a)+(1*y_bar);

//Section Property:
I1 = (1/12)*(l*t2^3) + (l*t2*(y_bar - y2)^2);
I2 = (1/12)*(t1*b^3) + (t1*b*(y1 - y_bar)^2);
```

```
I =I1+ 2*I2;

//Maximum Bending Stress:
c = b - y_bar;
sigma_max = (M*c)/(I*1000);

//Display:

  printf('\n\nThe maximum bending stress at section
     a-a = %1.1f MPa',sigma_max);


//
   ____________________________________________________________
   END
   ____________________________________________________________
```

Scilab code Exa 6.17 B17

```
clear all; clc;

disp("Scilab Code Ex 6.17 : ")

//Given:
b = 60/1000; //m
h = 30/1000; //m
M = 40; //Nm
c1= h/2;
rib_t = 5/1000; //m
rib_w = 10/1000;//m

//Without Ribs:
I1 = (1/12)*(b*h^3);
sigma_max1 = (M*c1)/(I1*10^6);
```

```

//With Ribs:
y1 = c1;
y2 = h+(rib_t/2);
A1 = h*b;
A2 = rib_t*rib_w;
y_bar = ((y1*A1)+2*(y2*A2))/(A1 + 2*A2);

c2 = h+rib_t - y_bar;
I2 = I1 + (b*h*(y_bar - y1)^2);
I3 = (1/12)*rib_w*rib_t^3 + (rib_w*rib_t*(y2 - y_bar
    )^2);
I = I2 + 2*I3;

sigma_max2 = (M*c2)/(I*10^6);

if(sigma_max2>sigma_max1)

  printf("\n\nThe maximum normal stress in the
      member without ribs = %1.2f MPa',sigma_max1);
 printf("\nThe maximum normal stress in the member
     with ribs = %1.2f MPa',sigma_max2);
 printf("\nThe ribs should be omitted.");

 end


//
    ----------------------------------------------------------------------
    END
    ----------------------------------------------------------------------
```

Scilab code Exa 6.18 B18

```
clear all; clc;

disp("Scilab Code Ex 6.18 : ")

//Given:
M = 12; //kNm
l_bc = 0.2; //m
l_be = 0.4; //m

//Internal Moment Components:
My = (-4/5)*M;
Mz = (3/5)*M;

Iy = (1/12)*(l_be*l_bc^3);
Iz = (1/12)*(l_bc*l_be^3);

//Bending Stress:
sigma_B = (-Mz*1000*(l_be/2))/Iz + (My*1000*(-l_bc
    /2))/Iy;
sigma_B = sigma_B/10^6;
sigma_C = (-Mz*1000*(l_be/2))/Iz + (My*1000*(l_bc/2)
    )/Iy;
sigma_C = sigma_C/10^6;
sigma_D = (-Mz*1000*(-l_be/2))/Iz + (My*1000*(l_bc
    /2))/Iy;
sigma_D = sigma_D/10^6;
sigma_E = (-Mz*1000*(-l_be/2))/Iz + (My*1000*(-l_bc
    /2))/Iy;
sigma_E = sigma_E/10^6;

//Orientation of Nuetral Axis:
z = (0.45)/(sigma_E + sigma_B);

//theta = -atan(4/3);
tanA = (Iz/Iy)*(-4/3);
alpha = atan(tanA);
alpha = alpha*(180/%pi);

```

```
//Display:


printf("\n\nThe normal stress at B = %1.2f MPa',
    sigma_B);
printf("\nThe normal stress at C = %1.2f MPa',
    sigma_C);
printf("\nThe normal stress at D = %1.2f MPa',
    sigma_D);
printf("\nThe normal stress at E = %1.2f MPa',
    sigma_E);
printf("\nThe orientation of the nuetral axis = %1
    .1f degrees',alpha);

//-----------------------------------------------------------------------------
    END
    -----------------------------------------------------------------------------
```

Scilab code Exa 6.19 B19

```
clear all; clc;

disp("Scilab Code Ex 6.19 : ")

//Given:
theta = 30*(%pi/180);
M = 15; //kNm
My = M*cos(theta);
Mz = M*sin(theta);
b = 0.1; //m
t1 = 0.04; //m
t2 = 0.03; //m

```

```
//Section Properties:
y1 = b/2;
y2 = b + t2/2;
A1 = (b*t1);
A2 = (b*2*t2);
z_bar = (y1*A1 + y2*A2)/(A1+A2);

Iz = (1/12)*(b*t1^3) + (1/12)*(t2*(2*b)^3);
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;

//Maximum Bending Stress:
l_b = b+t2 - z_bar;
sigma_B = (-Mz*1000*(-b))/Iz + (My*1000*(l_b))/Iy;
sigma_B = sigma_B/10^6;
sigma_C = (-Mz*1000*(t1/2))/Iz + (My*1000*(-z_bar))/
   Iy;
sigma_C = sigma_C/10^6;

sigma = max(abs(sigma_B),abs(sigma_C));

//Orientation of the nuetral axis:
theta1 = 60*(%pi/180);
alpha = atan((Iz/Iy)*tan(theta1));
alpha = alpha*(180/%pi);

//Display:


 printf("\n\nThe maximum normal stress in the beam =
    %1.2f MPa',sigma);
 printf("\n The orientation of the nuetral axis = %1
   .1f degrees',alpha);

 //
    ________________________________________________________________________
    END
```

Scilab code Exa 6.20 B20

```
clear all; clc;

disp("Scilab Code Ex 6.20 : ")

//Given:
M =20; //kN
Iy = 0.96*10^-3; //m^4
Iz = 7.54*10^-3; //m^4
theta = 57.1*(%pi/180);


//Internal moment Components:
My = M*sin(theta);
Mz = M*cos(theta);

//Bending Stress:
y_p = -0.2; //y Coordinate of P
z_p = 0.35; //z Coordinate of P

theta1 = (%pi/2)-(theta);
yp = -z_p*sin(theta1)+ y_p*cos(theta1);
zp = z_p*cos(theta1) + y_p*sin(theta1);

//Eq 6-17

sigma_p = ((Mz*-yp)/Iz) + ((My*zp)/Iy) ;
sigma_p = sigma_p/10^3;

//Orientation of the Nuetral Axis:
alpha = atan((Iz/Iy)*tan(theta));
```

```
alpha = alpha*(180/%pi);

//Display:


 printf("\n\nThe maximum normal stress at point P =
    %1.2f MPa',sigma_p);
 printf("\nThe orientation of the nuetral axis = %1
    .1f degrees',alpha);

 //
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 6.21 B21

```
clear all; clc;

disp("Scilab Code Ex 6.21 : ")

//Given:
M = 2; //kNm
Ew = 12; //GPa
Est = 200; //GPa
bw = 150/1000; //m
t = 20/1000; //m
rib = 9/1000;//m

//Section Properties:
n = (Ew/Est);
bst = n*bw;

y1 = t/2;
```

```
A1 = t*bw;
y2 = bw/2 + t;
A2 = rib*bw;

y_bar = (y1*A1 +y2*A2)/(A1+A2);

I1 = (1/12)*(bw)*(t^3) + A1*(y_bar - y1)^2;
I2 = (1/12)*(rib)*(bw^3) + A2*(y2-y_bar)^2;
Ina = I1+I2;

//Normal Stress:
sigma_B = (M*(bw+t-y_bar))/(Ina*1000);
sigma_C = (M*(y_bar))/(Ina*1000);

//Normal Stress in the wood:
sigmaB = n*sigma_B;

//Display:


 printf("\n\nThe normal stress at point B = %1.1f
    MPa',sigma_B);
 printf("\nThe normal stress at point C = %1.2f MPa
    ',sigma_C);
 printf("\nThe normal stress at point B in the wood
    = %1.2f MPa',sigmaB);

 //------------------------------------------------------------------
    END
    ------------------------------------------------------------------
```

Scilab code Exa 6.22 B22

```
clear all; clc;
```

```
disp("Scilab Code Ex 6.22 : ")

//Given:
sigma_allow_st = 168; //MPa
sigma_allow_w = 21; //MPa
Est = 200; //GPa
Ew = 12; //GPa
Iz = 7.93*10^6; //mm^4
A1 = 5493.75; //mm^2
t = 5; //mm
h = 100; //mm

//Without Board:
c = h+t;
M1 = (sigma_allow_st*Iz)/(c*10^6);

//With Board:
bw = 300;//mm
n = (Ew/Est);
bst = n*bw;

//For the transformed section:
y1 = 0;
y2 = 55;
A2 = bst*h;

y_bar = (y1*A1 + y2*A2)/(A1+A2);

I1 = Iz + A1*y_bar^2;
I2 = (1/12)*(bst*h^3) + (A2*(y2-y_bar)^2);
I = I1+I2;

c = c+y_bar;
M2 = (sigma_allow_st*I)/(c*10^6);

cw = c - y_bar;
Mw = (sigma_allow_w*I)/(n*cw*10^6);
```

```
M = min(Mw,M2);

//Display:

  printf("\n\nThe maximum bending moment without re-
     inforcement = %1.3f kNm',M1);
 printf("\nThe maximum bending moment with re-
    inforcement      = %1.2f kNm',M);

 //
    ----------------------------------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------------------------------
```

Scilab code Exa 6.23 B23

```
clear all; clc;

disp("Scilab Code Ex 6.23 : ")

//Given:
M = 60; //kNm
Est = 200; //GPa
Econc = 25; //GPa
d = 25;//mm
r = d/2;
w = 300;//mm
ht =400; //mm

//Section Properties:
n = Est/Econc;
Ast = 2*%pi*r^2;
A = n*Ast;
```

```

p = [1 52.37 -20949.33]
h = roots(p)
h = h(2);

I = (1/12)*(w*h^3) +w*h*(h/2)^2 + A*(ht - h)^2;

//Normal Stress:
sigma_conc_max = (M*1000*h*1000)/(I);
sigma_conc = (M*1000*(ht-h)*1000)/(I);
sigma_st = n*sigma_conc;

//Display:


 printf("\n\nThe normal stress in each steel
    reinforcing rod = %1.2f MPa',sigma_st);
 printf("\nThe maximum normal stress in the concrete
     = %1.2f MPa',sigma_conc_max);
 //
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 6.24 B24

```
clear all; clc;

disp("Scilab Code Ex 6.24 : ")

//Given:
sigma = 140; //Mpa
ri = 90; //mm
ro = 110; //mm
```

```
a = 20; //mm

//Section Properties:

y = integrate('a*(1/r)','r',ri,ro)
R = (a*a)/y;

r_avg = (ri+ro)/2;
M1 = (-sigma*a*a*ro*(r_avg - R))/(R-ro);
M1 = M1*10^-6;

M2 = (sigma*a*a*ri*(r_avg - R))/(R-ri);
M2 = M2*10^-6;

M = min(M1,M2);

sigma1 = (M*(R - ro))/(a*a*ro*(r_avg - R));

//For a straight Bar:
I = (1/12)*(a*a^3);
c = 10; //mm
M_strt= (sigma*I)/c;
M_strt = M_strt*10^-6;

//Display:

 printf("\n\nThe maximum bending moment that can be
    applied to the bar = %1.3f kNm',M);
 printf("\nThe maximum bending moment that can be
    applied to a straight bar = %1.3f kNm',M_strt);
 //
    ----------------------------------------------------------------
    END
    ----------------------------------------------------------------
```

Scilab code Exa 6.25 B25

```
clear all; clc;

disp("Scilab Code Ex 6.25 : ")

//Given:
ri = 200/1000; //m
r1 = 250/1000; //m
ro = 280/1000; //m
M = 4; //kNm
a = 0.05; //m
h = 0.03; //m

//Section Properties:
A1 = a^2 ;
A2 =  (0.5*a*h);
A = A1+A2;
r_avg1 = (r1+ri)/2;
r_avg2 = r1+(h/3);
r_bar =((r_avg1*A1)+(r_avg2*A2))/A;

int_dA_r1 = a*log(r1/ri);
int_dA_r2 = (a*ro*log(ro/r1))/(ro-r1) - a;
R = (A)/(int_dA_r1+ int_dA_r2);
k= r_bar - R;

//Normal Stress:
sigma_B = (-M*(R-ri))/(A*ri*k*1000);
sigma_A = (-M*(R-ro))/(A*ro*k*1000);

sigma = max(abs(sigma_B),abs(sigma_A))
```

```
//Display:

 printf("\n\nThe maximum normal stress in the bar =
    %1.0f MPa',sigma);

 //----------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------
```

Scilab code Exa 6.26 B26

```
clear all; clc;

disp("Scilab Code Ex 6.26 : ")

//Given:
M = 5; //kNm
sigma_y = 500; //MPa
r = 16; //mm
h = 80; //mm
w = 120; //mm
r_h = r/h;
w_h = w/h;
k = 1.45;
c = h/(2000);
t = 20/1000; //m

//Calculations:
I = (1/12)*(t)*(h/1000)^3
sigma_max = (k*M*c)/(I*1000);

//Display:

 printf("\n\nThe maximum normal stress in the steel
```

```
    = %1.0f MPa',sigma_max);

//------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 6.27 B27

```
clear all; clc;

disp("Scilab Code Ex 6.27 : ")

//Given:
sigma_y = 250; //MPa
t = 12.5; //mm
w = 200; //mm
h = 225; //mm

//Maximum Elastic Moment:
yy = (h+t)/2;
I1 = (1/12)*(w*t^3) + (w*t*yy^2);
I = (1/12)*(t*h^3) + 2*(I1);
c = 125; //mm

My = (sigma_y*I)/(c); //Flexure Formula

//Plastic Moment:
C1= sigma_y*t*(h/2);
C2= sigma_y*t*(w);
Mp = (2*56.25*C1) + (2*yy*C2);

//Shape Factor:
k = Mp/My;

```

```
//Display:


 printf("\n\nThe shape factor for the beam = %1.2f
    ',k);

 //-----------------------------------------------------------------------------
    END
    -----------------------------------------------------------------------------
```

Scilab code Exa 6.28 B28

```
clear all; clc;

disp("Scilab Code Ex 6.28 : ")

//Given:
sigma_y = 250; //MPa
t = 15/1000; //m
w = 100/1000; //m
h = 120/1000; //m
c = 10/1000; //m

//Calculations:
d = ((sigma_y*t*w)+(sigma_y*t*h))/(sigma_y*t*2);

T = sigma_y*t*d*10^3;
C1 = sigma_y*t*c*10^3;
C2 = sigma_y*t*w*10^3;

Mp = (T*d/2)+(C1*c/2)+(C2*(c+t/2));

//Display:

```

```
 printf("\n\nThe plastic moment that can be resisted
     by the beam = %1.1f kNm',Mp);

//-------------------------------------------------------------------------------
    END
    ---------------------------------------------------------------------------
```

Scilab code Exa 6.29 B29

```
clear all; clc;

disp("Scilab Code Ex 6.29 : ")

//Given:
ep1 = 0.01;
ep2 = 0.05;
sig1 = 1050;//N/mm^2
sig2 = 1330;//N/mm^2
sig3 = 280; //N/mm^2
y = 0.3; //cm
h = 3; //cm
w = 2; //cm

//Calculations:
yy = (h/2)-y
T1 = (1/2)*(sig3*yy*w);
y1 = y +(2/3)*(yy);
T2 = yy*sig1*w;
y2 = y+(0.5*yy);
T3 = (0.5*y*sig1*w);
y3 = (2/3)*(y);

M = 2*(T1*y1 + T2*y2 + T3*y3);
```

```
M = M/1000;

//Display:


 printf("\n\nThe bending moment applied that will
    cause a strain of 0.05mm/mm = %1.2f kNm',M);

 //———————————————————————————————————————————————————————————————————————————
    END
    ———————————————————————————————————————————————————————————————————————————
```

Scilab code Exa 6.30 B30

```
clear all; clc;

disp("Scilab Code Ex 6.30 : ")

//Given:
sigma_y = 250; //MPa
t = 12.5; //mm
w = 200; //mm
h = 225; //mm
c = (h/2)+t;
I = 82.44*10^6;//mm^4
Mp = 188; //kN

//Calculations:
sigma_allow = (Mp*10^6*c)/(I);
y = (sigma_y*c)/(sigma_allow);

//Display:

 printf("\n\nThe point of zero normal stress = %1.2f
```

```
mm',y);
printf("\nThe Residual Stress distribution is shown
    in the text book.");

//------------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------------
```

Chapter 7

Transverse Shear

Scilab code Exa 7.1 TS1

```
clear all; clc;

disp("Scilab Code Ex 7.1 : ")

//Given:
V = 3; //kN
h = 125; //mm
b = 100; //mm
y_top = 50; //mm
x_right = 37.5; //mm

//Part (a):

//Section Properties:
I = (b*h^3)/12;
y_dash_1 = ((h-y_top)-(h/2));
A = y_top*b;
Q = (y_dash_1+(y_top/2))*A;

//Shear Stress:
tou_p = (V*Q)/(I*b); //tou = VQ/It
```

```
tou_p = tou_p*10^3;

//Part (b):

//Section Properties:
y_dash_2 = (y_dash_1+(y_top));
a_dash= b*y_dash_2;
Q_dash =(y_dash_2*a_dash)/2;

//Shear Stress:
tou_max = (V*Q_dash)/(I*b);
tou_max = tou_max*10^3;

//Display:

printf("\n\nThe shear stress in the beam at point P
      = %1.3f MPa',tou_p);
printf('\nThe maximum shear stress in the beam
         = %1.3f MPa',tou_max);

//---------------------------------------------------------------
   END
   ---------------------------------------------------------------
```

Scilab code Exa 7.2 TS2

```
clear all; clc;

disp("Scilab Code Ex 7.2 : ")

//Given:
V = 80; //kN
thick_1 = 20/1000; //m
thick_2 = 15/1000; //m
```

```
l = 300/1000; //m
y = 100/1000; //m
h = 2*y;
y_dash = y +thick_1/2;

//Part(a):

I1 = (thick_2*(h^3))/12;
I2 = (l*(thick_1^3))/12;
I3 = (l*thick_1*(y_dash)^2);
I = I1+2*(I2+I3); //Moment of inertia

Q_b = y_dash*l*thick_1;
//At B'
tou_b_dash = (V*Q_b)/(I*l*1000);
//At B
tou_b = (V*Q_b)/(I*thick_2*1000);

//At C:
Q_c = (y_dash*l*thick_1)+(y*thick_2*y/2);
tou_c = (V*Q_c)/(I*thick_2*1000);

//Part(b)


y0 = -0.1;
y1 = 0.1;

function Q =f(y),Q = ((0.735 - (7.5*y*y))*10^-3),
endfunction
Int =intg(y0,y1,f)

V_w = (V*Int*thick_2)/(I*thick_2);

//Display:

printf("\n\nThe shear stress at B dash                =
    %1.2f MPa', tou_b_dash);
```

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

//
  ------------------------------------------------------------------------
  END
  ------------------------------------------------------------------------
```

Scilab code Exa 7.3 TS3

```
clear all; clc;

disp("Scilab Code Ex 7.3 : ")

//Given:
udl = 6.5; //kN
l_bc = 8; //m
l = 150/1000;//m
t = 30/1000;//m

//Internal Shear:
w = udl*l_bc/2;
l_wc = l_bc/4;
l_bw = l_bc - l_wc;
V = (w*l_bw)/l_bc;
R_b = w - V;

//Section Properties:
y1= l/2;
A = (l*t);
```

```
y2= l+(t/2);
y_dash = (y1*A + y2*A)/(2*A);
I1 = (t*l^3)/12;
I2 = (A*(y_dash-y1)^2);
I3 = (l*t^3)/12;
I4 = (A*(y2 - y_dash)^2);
I = I1+I2+I3+I4;

Q = ((l+t)-(t/2)-y_dash)*A;

//Shear Stress:
tou_max = (V*Q)/(I*t*1000);

//Display:

printf("\n\nThe maximum shear stress in the glue
   necessary to hold the boards together     = %1.2f
   MPa',tou_max);


//--------------------------------------------------------------------
   END
   --------------------------------------------------------------------
```

Scilab code Exa 7.4 TS4

```
clear all; clc;

disp("Scilab Code Ex 7.4 : ")

//Given:

V = 850; //kN
l1 =250/1000; //m
```

```
l2 = 300/1000; //m
l3 = 125/1000;//m
t = 10/1000; //m
h = 200/1000; //m

A1 = l1*t;
A2 = l2*t;
A3 = l3*t;

y1 = l2+(t/2);
y2 = l2/2;
y3 = h+(t/2);

y_dash = (2*y2*A2 + A1*y1 + A3*y3)/(2*A2 + A1 + A3);

I1 = ((l1*t^3)/12) +(A1 * (l2+(t/2)-y_dash)^2);
I2 = ((t*l2^3)/12) +(A2 * (y_dash - (l2/2))^2);
I3 = ((l3*t^3)/12) +(A1 * (h+(t/2)-y_dash)^2);
I = 2*I2 + I1 + I3;

Q_b = (l2+(t/2) - y_dash)*A1; //Q = y'A'
Q_c = (h+(t/2) - y_dash)*A3; //Q = y'A'

//Shear Flow:

q_b = (V*Q_b)/I;
q_c = (V*Q_c)/I;

q_b = q_b/(2*1000);
q_c = q_c/(2*1000);

//Display:

printf("\n\nThe shear flow at B, resisted by the
    glue is      = %1.2f MN/m',q_b);
printf("\nThe shear flow at C, resisted by the glue
    is     = %1.4f MN/m',q_c);

```

```
//
    ____________________________________________
    END
    ____________________________________________
```

Scilab code Exa 7.5 TS5

```
clear all; clc;

disp("Scilab Code Ex 7.5 : ")

//Given:
V = 80; //N
t = 1.5; //cm
a = 7.5; //cm
b = a-2*t; //cm
F_nail= 30; //N

//Section Properties:
I = (a*a^3 - b*b^3 )/12;
Q_b = (((a-2*t)/2)+(t/2))*a*t; //Q = y'A'
Q_c = (((a-2*t)/2)+(t/2))*(a-2*t)*t; //Q = y'A'

//Shear Flow:
q_b = (V*Q_b)/I;
q_c = (V*Q_c)/I;

s_b = F_nail/(q_b/2);
s_c = F_nail/(q_c/2);

//Display:
```

```
printf("\n\nThe maximum spacing of nails required at
    B is      = %1.0f cm',s_b);
printf("\nThe maximum spacing of nails required at C
    is     = %1.1f cm',s_c);



//
    ______________________________________________________________________________
    END
    ______________________________________________________________________________
```

Scilab code Exa 7.6 TS6

```
clear all; clc;

disp("Scilab Code Ex 7.6 : ")

//Given:
F = 40; //N
s = 9; //cm
h = 5; //cm
t = 0.5; //cm
w = 3; //cm
w_3 = w/3; //cm

//Calculations:

I = (w*h^3)/12 - (2*w_3*(h - 2*t)^3)/12;

//Case 1:

```

```
Q1 = ((h-t)/2)*(w*t);
V1 =((F/s)*I)/Q1 ;  //q = VQ/I

//Case2:

Q2 = ((h-t)/2)*(w_3*t);
V2 =((F/s)*I)/Q2 ;  //q = VQ/I

//Display:


printf("\n\nThe largest vertical shear that can be
   supported in Case 1     = %1.1f N',V1);
printf("\nThe largest vertical shear that can be
   supported in Case 2     = %1.1f N',V2);

//
   ------------------------------------------------------------
   END
   ------------------------------------------------------------
```

Scilab code Exa 7.7 TS7

```
clear all; clc;

disp("Scilab Code Ex 7.7 : ")

//Given:
V = 10; //kN
b1 = 6; //cm
h1 = 8; //cm
t = 1; //cm
b2 = b1-2*t;
h2 = h1-2*t; //cm
```

```
b3 = 4; //cm

//Calculations:
I = ((b1*h1^3)/12) - ((b2*h2^3)/12);

q_b = 0;

Q_c = ((b1/2)+(t/2))*(b3+(t))*t;
q_c = (V*Q_c*100)/(I); //Q = VQ/I

Q_d = (2*h1/4*t*b3) + ((b1/2)+(t/2))*b3*t;
q_d = (V*Q_d*100)/(I); //Q = VQ/I

//Display:


printf("\n\nVariation of shear flow at B  = %1.1f N/
   mm',q_b);
printf("\nVariation of shear flow at C  = %1.1f N/mm
   ',q_c);
printf('\nVariation of shear flow at D  = %1.1f N/mm
   ',q_d);

//
   ________________________________________________________________
   END
   ________________________________________________________________
```

Chapter 8

Combined Loadings

Scilab code Exa 8.1 CL1

```
clear all; clc;

disp("Scilab Code Ex 8.1 : ")

//Given:
di = 1.2*1000; //m
ri = di/2;
t = 12; //mm
sigma = 140; //MPa

//Cylindrical Pressure Vessel:

p1 = (t*sigma)/ri; //sigma = pr/t

//Spherical Vessel:

p2 = (2*t*sigma)/(ri); //sigma = pr/2t

//Display:

printf("\n\nThe maximum internal pressure the
```

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

//--------------------------------------------------------------------------------
    END
    --------------------------------------------------------------------------------
```

Scilab code Exa 8.2 CL2

```
clear all; clc;

disp("Scilab Code Ex 8.2 : ")

//Given:
P = 15000; //N
a = 40; //mm
b = 100; //mm

//Stress Components:

//Normal Force:
A = a*b;
sigma = P/A;

//Bending Moment:
I = (a*b^3)/12; //I = (1/12)*bh^3
M = P*(b/2);(b/2);
c = b/2;
sigma_max =(M*c)/I;

//Superposition:
```

```
x = ((sigma_max-sigma)*b)/((sigma_max+sigma)+(
    sigma_max-sigma));
sigma_b = (sigma_max-sigma);
sigma_c = (sigma_max + sigma);

//Display:

printf("\n\nThe state of stress at B     = %1.1f MPa
    (tensile)',sigma_b);
printf('\nThe state of stress at C     = %1.1f MPa (
    compressive)',sigma_c);

//————————————————————————————————————————————————————————————————
    END
    ————————————————————————————————————————————————————————————————
```

Scilab code Exa 8.3 CL3

```
clear all; clc;

disp("Scilab Code Ex 8.3 : ")

//Given:
ri = 600/1000; //m
t = 12/1000; //m
ro = ri+t;
sp_wt_water = 10; //kN/m^3
sp_wt_steel = 78; //kN/m^3
l_a = 1; //m depth of point A from the top

//Internal Loadings:
v = (%pi*l_a)*(ro^2 - ri^2);
W_st = sp_wt_steel*v;

```

```
p = sp_wt_water*l_a; //Pascal's Law

//Stress Components:

//Circumferential Stress:
sigma1 = (p*ri)/t;

//Longitudinal Stress:
A_st = (%pi)*(ro^2 - ri^2);
sigma2 = W_st/A_st;

//Display:


printf("\n\nThe state of stress at A (
    Circumferential)     = %1.1f kPa',sigma1);
printf('\nThe state of stress at A (Longitudinal)
          = %1.1f kPa',sigma2);

//------------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------------
```

Scilab code Exa 8.4 CL4

```
clear all; clc;

disp("Scilab Code Ex 8.4 : ")

//Given:
y_c = 125/1000; //m
x_c = 1.5; //m
y_b = 1.5; //m
x_b = 6; //m
```

```
udl = 50; //kN/m
l_udl = 2.5; //m
l = 250/1000; //m
width = 50/1000; //m


//Internal Loadings:
N = 16.45; //kN
V = 21.93; //kN
M = 32.89; //kNm

//Stress Components:

//Normal Force:
A = l*width;
sigma1 = N/(A*1000);

//Shear Force:
tou_c = 0;

//Bending Moment:
c = y_c;
I = (1/12)*(width*l^3);
sigma2 = (M*c)/(I*1000);

//Superposition:
sigmaC = sigma1+sigma2;

//Display:


printf('\n\nThe stress due to normal force at C
    = %1.2f MPa',sigma1);
printf('\nThe stress due to shear force at C       =
    %1.2f MPa',tou_c);
printf('\nThe stress due to bending moment at C    =
    %1.2f MPa',sigma2);
printf('\nThe resultant stress at C                =
```

```
%1.1f MPa',sigmaC);

//
_______________________________________________________________________________
END
_______________________________________________________________________________
```

Scilab code Exa 8.5 CL5

```
clear all; clc;

disp("Scilab Code Ex 8.5 : ")

//Given:
r = 0.75*10; //mm
f_x =500;//N
f_y =800;//N
l1 = 8*10; //mm
l2 = 10*10; //mm
l3 = 14*10; //mm

//Stress Components:

//Normal Force:
A1 = (%pi*r^2);
sigma1 = f_x/A1; //stress = P/A

//Shear Force:
y_bar = (4*r)/(3*%pi);
A2 = A1/2;
Q = y_bar*A2; //Q = yA
V = f_y;
I = (1/4)*(%pi*r^4);
t = 2*r;
```

```
tou_a = (V*Q)/(I*t); //Shear = VQ/It

//Bending Moment:
M_y = f_x*l3;
c = r;
sigma_A = (M_y*c)/I;

//Torsional Moment:
T = f_y*l3;
J = (0.5*%pi*r^4);
tou_A = (T*c)/J;

//Resultant:
res_normal= sigma1+sigma_A;
res_shear = tou_a+tou_A;

//Display:

printf('\n\nThe stress due to normal force at A
        = %1.2f MPa',sigma1);
printf('\nThe stress due to shear force at A
        = %1.2f MPa',tou_a);
printf('\nThe stress due to bending moment at A
        = %1.2f MPa',sigma_A);
printf('\nThe stress due to torsional moment at A
        = %1.2f MPa',tou_A);
printf('\nThe resultant normal stress component at A
        = %1.2f MPa',res_normal);
printf('\nThe resultant shear stress component at A
        = %1.2f MPa',res_shear);

//
    ---------------------------------------------------------------------------
    END
    ---------------------------------------------------------------------------
```

Scilab code Exa 8.6 CL6

```
clear all; clc;

disp("Scilab Code Ex 8.6 : ")

//Given:
P = 40; //kN
l_ab = 0.4; //m
l_bc = 0.8; //m

//Stress Components:

//Normal Force:
A = l_ab*l_bc;
sigma = P/A;

//Bendng Moments:
M_x = P*l_ab/2;
cy = l_ab/2;
Ix = (1/12)*(l_bc*l_ab^3); //I = (1/12)*(bh^3)
sigma_max_1 = (M_x*cy)/Ix; //sigma = My/I

M_y = P*l_bc/2;
cx = l_bc/2;
Iy = (1/12)*(l_ab*l_bc^3); //I = (1/12)*(bh^3)
sigma_max_2 = (M_y*cx)/Iy; //sigma = My/I

//Superposition:
stress_A = -sigma + sigma_max_1 + sigma_max_2;
stress_B = -sigma - sigma_max_1 + sigma_max_2;
stress_C = -sigma - sigma_max_1 - sigma_max_2;
stress_D = -sigma + sigma_max_1 - sigma_max_2;

```

```
e = abs((stress_B*l_ab)/(stress_A-stress_B));
h = abs((stress_B*l_bc)/(stress_A-stress_B));

//Display:


printf('\n\nThe normal stress at corner A
    = %1.0f kPa',stress_A);
printf('\nThe normal stress at corner B              =
    %1.0f kPa',stress_B);
printf('\nThe normal stress at corner C              =
    %1.0f kPa',stress_C);
printf('\nThe normal stress at corner D              =
    %1.0f kPa',stress_D);
printf('\nThe line of zero stress along AB           =
    %1.4f m',e);
printf('\nThe line of zero stress along AD           =
    %1.3f m',h);

//
    ____________________________________________________________
    END
    ____________________________________________________________
```

Chapter 9

Stress Transformation

Scilab code Exa 9.1 ST1

```

clear all; clc;

disp("Scilab Code Ex 9.1 : ")

//Given:
tou = 25; //MPa
sigma1 = 50; //MPa
sigma2 = 80; //MPa
phi = 30*(%pi/180);

// Calculations:
sigma_x1 = (sigma1*cos(phi)*cos(phi))- (tou*cos(phi)
    *sin(phi)) - (sigma2*sin(phi)*sin(phi))- (tou*sin
    (phi)*cos(phi));
tou1 = (sigma1*cos(phi)*sin(phi))+ (tou*cos(phi)*cos
    (phi)) + (sigma2*sin(phi)*cos(phi))- (tou*sin(phi
    )*sin(phi));
sigma_x2 = (tou*cos(phi)*sin(phi))- (sigma2*cos(phi)
    *cos(phi)) + (tou*sin(phi)*cos(phi))+ (sigma1*sin
    (phi)*sin(phi));
```

```
tou2 = (tou*cos(phi)*cos(phi))+ (sigma2*cos(phi)*sin
    (phi)) - (tou*sin(phi)*sin(phi))+ (sigma1*sin(phi
    )*cos(phi));

//Display:

printf("\n\nThe normal stress component in the x
    diection is        = %1.2f MPa',sigma_x1);
printf("\n  The shear stress component in the x
    diection is     = %1.1f MPa',tou1);
printf("\n  The normal stress component in the y
    diection is     = %1.1f MPa',sigma_x2);
printf("\n  The shear stress component in the y
    diection is     = %1.1f MPa',tou2);

//
    ----------------------------------------------------------------------
    END
    ----------------------------------------------------------------------
```

Scilab code Exa 9.2 ST2

```
clear all; clc;


disp("Scilab Code Ex 9.2 : ")

//Given:
phi = -30*(%pi/180);
theta = 60*(%pi/180);
sigma_x = -80; //MPa
sigma_y = 50; //MPa
tou_xy = -25; //MPa

```

```
//Plane CD:
sigma_x1 = (sigma_x+sigma_y)/2 + ((sigma_x-sigma_y)*
    cos(2*phi))/2 + (tou_xy*sin(2*phi)); //Eqn 9.1
tou_xy1 = ((-(sigma_x - sigma_y)*sin(2*phi))/2) + (
    tou_xy*cos(2*phi)); //Eqn 9.2

//Plane BC:
sigma_x2 = (sigma_x+sigma_y)/2 + ((sigma_x-sigma_y)*
    cos(2*theta))/2 + (tou_xy*sin(2*theta)); //Eqn
    9.1
tou_xy2 = (-(sigma_x - sigma_y)*sin(2*theta))/2 +
    tou_xy*cos(2*theta); //Eqn 9.2

//Display:

printf('\n\nThe normal stress of plane CD inclined
    at 30 degrees    = %1.1f MPa',sigma_x1);
printf('\nThe shear stress of plane CD inclined at
    30 degrees     = %1.1f MPa',tou_xy1);
printf('\nThe normal stress of plane BC inclined at
    60 degrees    = %1.2f MPa',sigma_x2);
printf('\nThe shear stress of plane BC inclined at
    60 degrees     = %1.1f MPa',tou_xy2);

//
    ----------------------------------------------------------------
    END
    ----------------------------------------------------------------
```

Scilab code Exa 9.5 ST5

```
clear all; clc;

disp("Scilab Code Ex 9.5 : ")
```

```

//Given:
sigma_x = -20; //MPa
sigma_y = 90; //MPa
tou_xy = 60; //MPa

//Orientation of Element:
theta_p2 = atan((2*tou_xy)/(sigma_x - sigma_y));
theta_p2 = theta_p2/2;
theta_p1 = (180+2*theta_p2)/2;

//Principal Stresses:

sigma1 = ((sigma_x+sigma_y)/2)+(sqrt(((sigma_x -
   sigma_y)/2)^2 + tou_xy^2));
sigma2 = ((sigma_x+sigma_y)/2)- sqrt(((sigma_x-
   sigma_y)/2)^2 + tou_xy^2);
sigma_x2 = ((sigma_x+sigma_y)/2)+ (((sigma_x-sigma_y
   )/2)*cos(2*theta_p2)) + (tou_xy*sin(2*theta_p2));

//Display:

printf("\n\nThe first principal stress is
                          = %1.0f  MPa',sigma1);
printf("\nThe second principal stress is
                         = %1.1f  MPa',sigma2);
printf('\nThe normal stress acting on the 23.7
   degrees plane  = %1.1f MPa',sigma_x2);

//
   ------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------
```

Scilab code Exa 9.6 ST6

```
clear all; clc;

disp("Scilab Code Ex 9.6 : ")

//Given:
sigma_x = -20; //MPa
sigma_y  = 90; //MPa
tou_xy =60; //Mpa

//Orientation of Element:
theta_s2 = atan(-(sigma_x - sigma_y)/(2*tou_xy));
theta_s2 = theta_s2/2;
theta_s1 = %pi + 2*theta_s2;
theta_s1 = theta_s1/2;

//Maximum in plane Shear Stress:
tou_max = (sqrt(((sigma_x - sigma_y)/2)^2 + tou_xy
   ^2));
tou_xy1 = -(sigma_x - sigma_y)*(sin(2*theta_s2))/2 +
    (tou_xy*cos(2*theta_s2));

//Average Normal Stress:
sigma_avg = (sigma_x+sigma_y)/2;

//Display:

printf("\n\nThe maximum in-plane shear stress is
          = %1.1f   MPa',tou_xy1);
printf("\nThe average normal stress is
                   = %1.0f   MPa',sigma_avg);

//
   ------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------
```

Scilab code Exa 9.9 ST9

```
clear all; clc;

disp("Scilab Code Ex 9.9 : ")

//Given:
sigma_x = -12; //MPa
sigma_y = 0;
tou_xy = -6; //MPa

//Construction of the circle:
sigma_avg = (sigma_x+sigma_y)/2;
R = sqrt((-sigma_x+sigma_avg)^2 + (tou_xy)^2);

//Principal Stresses:
sigma2 = -R+sigma_avg;//From the Mohr's circle
sigma1 = R+sigma_avg;

theta_p2 = atan((-tou_xy)/(-sigma_x+sigma_avg));
theta_p2 = theta_p2/2*(180/%pi);

//Display:

printf('\n\nThe first principal stress is
              = %1.2f  MPa',sigma1);
printf('\nThe second principal stress is          =
    %1.1f  MPa',sigma2);
printf('\nThe direction of the principal plane is  =
    %1.1f  degrees',theta_p2);



//
```

END

Scilab code Exa 9.10 ST10

```
clear all; clc;

disp("Scilab Code Ex 9.10 : ")

//Given:
sigma_x = -20; //MPa
sigma_y = 90; //MPa
tou_xy = 60; //MPa

//Construction of the circle:
sigma_avg = (sigma_x+sigma_y)/2;
R = sqrt(((sigma_x-sigma_avg))^2 + (tou_xy)^2);

//Maximum In plane Shear Stress:
tou_max = R;

theta_s1 = atan(-(sigma_x - sigma_avg)/(tou_xy));
theta_s1 = theta_s1/2*(180/%pi);

//Display:

printf('\n\nThe maximum in-plane shear stresses are
     = %1.1f  MPa',tou_max);
printf('\n
   = %1.1f  MPa',sigma_avg);
printf('\nThe orientation of the element is
   = %1.1f  degrees',theta_s1);

```

```
//
    ----------------------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------------------
```

Scilab code Exa 9.11 ST11

```

clear all; clc;

disp("Scilab Code Ex 9.11 : ")

//Given:
sigma_x  = -8; //MPa
sigma_y = 12; //MPa
tou_xy = -6; //Mpa

//Construction of the circle:
sigma_avg = (sigma_x+sigma_y)/2;

R = sqrt( 10^2 + tou_xy^2);

//Stresses on 30 degree element:
phi = atan(6/10);
psi = (%pi/3) - phi;

//On face BD:
sigma_x1 = 2 - (R*cos(psi));
tou_xy1 = (R*sin(psi));

//On face DE:
sigma_x2 = 2 + (R*cos(psi));
tou_xy2 = -(R*sin(psi));

```

```
//Display:

printf('\n\nThe normal stress on plane BD inclined
   at 30 degrees is    = %1.2f  MPa',sigma_x1);
printf('\nThe normal stress on plane DE inclined at
   60 degrees is    = %1.1f  MPa',sigma_x2);
printf('\nThe shear stress is
                                             = %1.2f
   MPa',tou_xy1);


//
   ____________________________________________________________
   END
   ____________________________________________________________
```

Scilab code Exa 9.12 ST12

```
clear all; clc;

disp("Scilab Code Ex 9.12 : ")

//Given:
P = 900; //N
T = 2.5; //Nm
d = 40/1000; //m
r = d/2;
c = r;

//Stress Components:
J = (%pi/2)*(r^4);
tou = (T*c)/(J*1000);

A = (%pi*r^2);
```

```
sigma = P/(A*1000);

//Principal Stresses:
sigma_avg = (0 + sigma)/2;

R = sqrt( sigma_avg^2 + tou^2);
sigma1 = sigma_avg + R;
sigma2 = sigma_avg - R;

//Display:

printf('\n\nThe prinicpal stresses at point P are:');
printf('\n %1.1f  kPa',sigma1);
printf('\n %1.1f  kPa',sigma2);

//-----------------------------------------------------------------------------END-----------------------------------------------------------------------------
```

Scilab code Exa 9.13 ST13

```
clear all; clc;

disp("Scilab Code Ex 9.13 : ")

//Given:
w = 120; //kN/m
I = 67.4*(10^-6);
V= 84; //kN
M = 30.6; //kNm
t = 10/1000; //m

```

```
//Stress Components:
y = 0.200/2;
sigma = -(M*10^3*y)/(I*10^6);

Q = (0.100 + 0.015/2)*(0.175)*(0.015);
tou = (V*Q*10^3)/(I*t*10^6);

//Principal Stresses:

k = sigma/2;
R = sqrt( (-sigma+k)^2 + tou^2);
sigma1 = R + k;
sigma2 = k -R ;

theta_p2 = atan(-tou/(sigma-k));
theta_p2 =theta_p2/2*(180/%pi);

//Display:


printf('\n\nThe prinicpal stresses at point P are:')
    ;
printf('\n %1.1f  MPa',sigma1);
printf('\n %1.1f  MPa',sigma2);
printf('\nThe angle of rotation of the plane  %1.1f
     degrees',theta_p2);

//
    ________________________________________________________________
    END
    ________________________________________________________________
```

Scilab code Exa 9.14 ST14

```
clear all; clc;

disp("Scilab Code Ex 9.14 : ")

//Given:
tou = 40; //kPa
sigma = -20; //kPa

//Principal Stresses:
sigma_avg = sigma/2;
R = sqrt( (-sigma + sigma_avg)^2 + tou^2);
sigma_max = sigma_avg + R  ;
sigma_min = sigma_avg - R  ;

theta = atan(tou/(-sigma+sigma_avg));
theta = theta/2;

//Absolute Maximum Shear Stress:
tou_max = (sigma_max - sigma_min)/2;
sigma_avg = (sigma_max + sigma_min)/2;

//Display:

printf('\n\nThe prinicpal stresses at the point are:
   ');
printf('\n  %1.1f  kPa',sigma_max);
printf('\n %1.1f  kPa',sigma_min);
printf('\nThe absolute maximum shear stress at the
   point  %1.1f kPa',tou_max);

//
   ________________________________________________________________________
   END
   ________________________________________________________________________
```

Scilab code Exa 9.15 ST15

```
clear all; clc;

disp("Scilab Code Ex 9.15 : ")

//Given:
sigma_max = 32; //MPa
sigma_min = 0; //MPa
sigma_int = 16; //MPa

tou_max = (sigma_max - sigma_min)/2 ; //MPa
sigma_avg = (sigma_max + sigma_min)/2 ; //MPa

tou_in_plane = (sigma_max - sigma_int)/2;
sigma_avg2 = sigma_avg + (tou_in_plane);

//Display:

printf('\n\nThe maximum absolute shear stress    = %1
    .2f  MPa',tou_max);

//
    ________________________________________________________________________________
    END
    ________________________________________________________________________________
```

Chapter 10

Strain Transformation

Scilab code Exa 10.1 StnT1

```
clear all; clc;

disp("Scilab Code Ex 10.1 : ")

//Given:
ep_x = 500; //Normal Strain
ep_y = -300; //Normal Strain
gamma_xy = 200; //Shear Strain
theta = 30*(%pi/180);
theta = theta*-1;

ep_x_new = ((ep_x+ep_y)/2) + ((ep_x - ep_y)/2)*cos
   (2*theta) + (gamma_xy/2)*sin(2*theta);

gamma_xy_new = -((ep_x - ep_y)/2)*sin(2*theta) + (
   gamma_xy/2)*cos(2*theta);
gamma_xy_new = 2*gamma_xy_new;

phi = 60*(%pi/180);
ep_y_new = (ep_x+ep_y)/2 + ((ep_x - ep_y)/2)*cos(2*
   phi) + (gamma_xy/2)*sin(2*phi);
```

```
//Display:


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);
printf('\nThe equivalent strain acting on the
   element in the y plain oriented at 30 degrees
   clockwise    = %1.1f *10^-6',ep_y_new);
printf('\nThe equivalent shear strain acting on the
   element
                                                = %1
   .0f *10^-6',gamma_xy_new);

//
   ____________________________________________________________________
   END
   ____________________________________________________________________
```

Scilab code Exa 10.2 StnT2

```
clear all; clc;

disp("Scilab Code Ex 10.2 : ")

//Given:
ep_x = -350;//(*10^-6) Normal Strain
ep_y = 200; //*(10^-6) Normal Strain
gamma_xy = 80; //*(10^-6) Shear Strain

//Orientation of the element:
tan_thetap = (gamma_xy)/(ep_x - ep_y);
thetap1 = (0.5)*(atan(tan_thetap));
```

```
//Principal Strains:

k = (ep_x + ep_y)/2;
l = (ep_x - ep_y)/2;
tou = gamma_xy/2;
R = sqrt( (l)^2 + tou^2);
ep1 = R + k;
ep2 = k -R ;
ep = [ep1 ep2];

ep_x1 = k + l*cos(2*thetap1)+ tou*sin(2*thetap1);
thetap1 = thetap1*(180/%pi);
thetap2 = (90 + thetap1);
thetap =[thetap1 thetap2];


//Display:

printf('\n\nThe orientation of the element in the
    positive counterclockwise direction     = %1.2f
    degrees, %1.2f  degrees ',thetap);
printf('\nThe principal strains are

    = %1.0f *10^-6 , %1.0f *10^-6 ',ep);
printf('\nThe principal strain in the new x
    direction is                            = %1.0
    f *10^-6 ',ep_x1);

//
    ________________________________________________________________________
    END
    ________________________________________________________________________
```

Scilab code Exa 10.3 StnT3

```
clear all; clc;

disp("Scilab Code Ex 10.3 : ")

//Given:
ep_x = -350;//(*10^-6) Normal Strain
ep_y = 200; //*(10^-6) Normal Strain
gamma_xy = 80; //*(10^-6) Shear Strain

//Orientation of the element:
tan_thetap = -(ep_x - ep_y)/(gamma_xy);
thetap1 = (0.5)*(atan(tan_thetap));

//Maximum in-plane shear strain:

l = (ep_x - ep_y)/2;
tou = gamma_xy/2;
R = sqrt( l^2 + tou^2);
max_inplane_strain = 2*R;

gamma_xy_1 = (-l*sin(2*thetap1)+ tou*cos(2*thetap1))*2;
strain_avg = (ep_x + ep_y)/2;

thetap1 = thetap1*(180/%pi);
thetap2 = (90 + thetap1);
thetap =[thetap1 thetap2];

//Display:

printf('\n\nThe orientation of the element                = %1.1f degrees, %1.1f  degrees ',thetap);
printf('\nThe maximum in-plane shear strain    = %1.0f *10^-6 ',max_inplane_strain);
printf('\nThe average strain
```

```
    = %1.0f *10^-6 ',strain_avg);

//
    ________________________________________________________________________
    END
    ________________________________________________________________________
```

Scilab code Exa 10.4 StnT4

```
clear all; clc;

disp("Scilab Code Ex 10.4 : ")

//Given:
ep_x = 250;//(*10^-6) Normal Strain
ep_y = -150; //*(10^-6) Normal Strain
gamma_xy = 120; //*(10^-6) Shear Strain

//Construction of the circle:
strain_avg = (ep_x + ep_y)/2;
tou = gamma_xy/2;
R = sqrt((ep_x - strain_avg)^2 + (tou^2));

//Principal Strains:
ep1 = (strain_avg + R);
ep2 = (strain_avg - R);
strain = [ep1 ep2];

tan_thetap = (tou)/(ep_x - strain_avg);
thetap1 = (atan(tan_thetap))/2;
thetap1 = thetap1*(180/%pi);

//Display:

```

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

//
   ______________________________________________________________________________
   END
   ______________________________________________________________________________
```

Scilab code Exa 10.5 StnT5

```
clear all; clc;

disp("Scilab Code Ex 10.5 : ")

//Given:
ep_x = 250;//(*10^-6) Normal Strain
ep_y = -150; //*(10^-6) Normal Strain
gamma_xy = 120; //*(10^-6) Shear Strain

//Orientation of the element:
thetas = 90 - 2*8.35;
thetas1 = thetas/2;

//Maximum in-plane shear strain:

l = (ep_x - ep_y)/2;
tou = gamma_xy/2;
R = sqrt( l^2 + tou^2);
max_inplane_strain = 2*R;

```

```
strain_avg = (ep_x + ep_y)/2;


//Display:

printf('\n\nThe orientation of the element
            = %1.1f degrees ',thetas1);
printf('\nThe maximum in-plane shear strain
   = %1.0f *10^-6 ',max_inplane_strain);
printf('\nThe average strain
   = %1.0f *10^-6 ',strain_avg);

//
   ----------------------------------------------------------------------------------------------------
   END
   ----------------------------------------------------------------------------------------------------
```

Scilab code Exa 10.6 StnT6

```
clear all; clc;

disp("Scilab Code Ex 10.6 : ")

//Given:
ep_x = -300;//(*10^-6) Normal Strain
ep_y = -100; //*(10^-6) Normal Strain
gamma_xy = 100; //*(10^-6) Shear Strain
theta = 20; //degrees


//Construction of the circle:
strain_avg = (ep_x+ ep_y)/2;
tou = gamma_xy/2;
```

```
R = sqrt((-ep_x + strain_avg)^2 + tou^2);

//Strains on Inclined Element:
theta1 = 2*theta;

phi = atan((tou)/(-ep_x +strain_avg));
phi = phi*(180/%pi);
psi = theta1 - phi;
psi = psi*(%pi/180);

ep_x1 = -(-strain_avg+ R*cos(psi));
gamma_xy1 = -(R*sin(psi))*2;

ep_y1 = -(-strain_avg - R*cos(psi));

//Display:

printf('\n\nThe normal strain in the new x direction
          = %1.0f *10^-6 ',ep_x1);
printf('\nThe normal strain in the new y direction
          = %1.1f *10^-6 ',ep_y1);
printf('\nThe shear strain in the new xy direction
          = %1.0f *10^-6 ',gamma_xy1);
printf('\nThe average strain
                                   = %1.0f *10^-6 ',
    strain_avg);

//
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 10.7 StnT7

```
clear all; clc;

disp("Scilab Code Ex 10.7 : ")

//Given:
ep_x = -400;//(*10^-6) Normal Strain
ep_y = 200; //*(10^-6) Normal Strain
gamma_xy = 150; //*(10^-6) Shear Strain

//Maximum in-plane Shear Strain:
strain_avg = (ep_x+ ep_y)/2;
tou = gamma_xy/2;

R = sqrt((-ep_x + strain_avg)^2 + tou^2);
strain_max = strain_avg + R;
strain_min = strain_avg - R;

max_shear_strain = strain_max - strain_min;

//Absolute Maximum Shear Strain:
abs_max_shear = max_shear_strain;

//Display:

printf('\n\nThe maximum in-plane principal strain     = %1.0f *10^-6 ',strain_max);
printf('\nThe minimum in-plane principal strain   = %1.0f *10^-6 ',strain_min);
printf('\nThe maximum in-plane shear strain   = %1.0f *10^-6 ',max_shear_strain);
printf('\nThe absolute maximum shear strain   = %1.0f *10^-6 ',abs_max_shear);
printf('\nThe average strain   = %1.0f *10^-6 ',strain_avg);
//--------------------------------------------------------------------------------END--------------------------------------------------------------------------------
```

Scilab code Exa 10.8 StnT8

```
clear all; clc;

disp("Scilab Code Ex 10.8 : ")

//Given:
ep_a = 60;//(*10^-6) Normal Strain
ep_b = 135; //*(10^-6) Normal Strain
ep_c = 264; //*(10^-6) Normal Strain

theta_a = 0;
theta_b = 60*(%pi/180);
theta_c = 120*(%pi/180);

//Using matrices to solve the equations:
a1 = (cos(theta_a))^2;
b1 = (sin(theta_a))^2;
c1 = cos(theta_a)*sin(theta_a);

a2 = (cos(theta_b))^2;
b2 = (sin(theta_b))^2;
c2 = cos(theta_b)*sin(theta_b);

a3 = (cos(theta_c))^2;
b3 = (sin(theta_c))^2;
c3 = cos(theta_c)*sin(theta_c);

A = [a1 b1 c1 ; a2 b2 c2; a3 b3 c3 ]
b = [ep_a ; ep_b ; ep_c];
strain = A\b;

ep_x = strain(1);
```

```
ep_y = strain(2);
gamma_xy = strain(3);

strain_avg = (ep_x + ep_y )/2;
tou = gamma_xy/2;

R = sqrt((-ep_x + strain_avg)^2 + tou^2);

ep1 = strain_avg + R;
ep2 = strain_avg - R;
ep = [ep1 ep2];

tan_thetap =atan(-tou/(-ep_x + strain_avg));
thetap = tan_thetap/2;
thetap2 = thetap*(180/%pi);

//Display:


printf('\n\nThe maximum in-plane principal strains
    are      = %1.0f *10^-6 , %1.1f *10^-6',ep);
printf('\nThe angle of orientation
                              = %1.1f degrees',thetap2);

//
    ------------------------------------------------------------------
    END
    ------------------------------------------------------------------
```

Scilab code Exa 10.9 StnT9

```
clear all; clc;

disp("Scilab Code Ex 10.9 : ")
```

```

//Given:
E_st = 200*10^9; //GPa
nu_st = 0.3; //Poisson's ratio
ep1 = 272 *10^-6;
ep2 = 33.8 *10^-6;

//Solving for sigma using matrices:
A = [1 -nu_st; -nu_st 1];
b = [(ep1*E_st) ; (ep2*E_st)];
sigma = A\b;

sigma1= sigma(1)/(10^6);
sigma2= sigma(2)/(10^6);

//Display:

printf('\n\nThe principal stresses at point A are = %1.0f MPa , %1.1f MPa',sigma1, sigma2);


//-------------------------------------------------------------------------- END --------------------------------------------------------------------------
```

Scilab code Exa 10.10 StnT10

```
clear all; clc;

disp("Scilab Code Ex 10.10 : ")

//Given:
a = 300; //mm
```

```
b = 50; //mm
t = 20; //mm
E_cu = 120*10^3; //MPa
nu_cu = 0.34;// Poisson's ratio

//By inspection:
sigma_x = 800; //MPa
sigma_y = -500; //MPa
tou_xy = 0;
sigma_z = 0;

//By Hooke's Law:
ep_x = (sigma_x/E_cu) - (nu_cu/E_cu)*(sigma_y +
    sigma_z);
ep_y = (sigma_y/E_cu) - (nu_cu/E_cu)*(sigma_x +
    sigma_z);
ep_z = (sigma_z/E_cu) - (nu_cu/E_cu)*(sigma_y +
    sigma_x);

//New lengths:

a_dash = a + ep_x*a;
b_dash = b + ep_y*b;
t_dash = t + ep_z*t;

//Display:

printf('\n\nThe new length      = %1.2fmm ',a_dash);
printf('\nThe new width       = %1.2f mm ',b_dash);
printf('\nThe new thickness   = %1.2f mm ',t_dash);
//
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 10.11 StnT11

```
clear all; clc;

disp("Scilab Code Ex 10.11 : ")

//Given:
p = 20; //kPa
E = 600; //kPa
nu = 0.45
a = 4; //cm
b = 2; //cm
c = 3; //cm

//Dilatation:
sigma_x = -p;
sigma_y = -p;
sigma_z = -p;

e = ((1-2*nu)/E)*(sigma_x + sigma_y + sigma_z);

//Change in Length:
ep = (sigma_x - nu*(sigma_y + sigma_z))/E;

del_a = ep*a;
del_b = ep*b;
del_c = ep*c;

//Display:

printf('\n\nThe change in length a      = %1.4fcm ',
   del_a);
printf('\nThe change in length b      = %1.5fcm ',
   del_b);
```

```
printf('\nThe change in length c       = %1.4fcm ',
   del_c);
//
   ____________________________________________________________________
   END
   ____________________________________________________________________
```

Scilab code Exa 10.12 StnT12

```
clear all; clc;

disp("Scilab Code Ex 10.12 : ")

//Given:
di = 60/1000; //m
ri = di/2;
d0 = 80/1000;//m
ro = d0/2;
T = 8000;//Nm
M = 3500; //Nm
sigma_y_sqr = 250^2; //MPa

//Calculations:
c = ro;
J = (%pi/2)*(ro^4 - ri^4)*(10^6);
I = (%pi/4)*(ro^4 - ri^4)*(10^6);
tou_a = (T*c)/J;
sigma_a = (M*c)/I;

sigma_avg = (0-sigma_a)/2;

R = sqrt(116.4^2 + sigma_avg^2);
sigma1 = sigma_avg + R;
sigma2 = sigma_avg - R;
```

```
test = (sigma1^2 - (sigma1*sigma2) + sigma2^2);


if(test<sigma_y_sqr)
    printf("\n\nThe material within the pipe will
       not yield.");
end

//
  ________________________________________________________________________________
  END
  ________________________________________________________________________________
```

Scilab code Exa 10.13 StnT13

```
clear all; clc;

disp("Scilab Code Ex 10.13 : ")

//Given:
T = 400; //Nm
sigma_ult = 150*10^6; //N/m^2

//Calculations:

x = T/(%pi/2);
r_3 = [x/sigma_ult];
r = nthroot(r_3, 3);
r= r*1000; //in mm

//Display:

printf('\n\nThe smallest radius of the solid cast
```

```
    iron shaft      = %1.2fmm ',r);

//
   ____________________________________________________________
   END
   ____________________________________________________________
```

Scilab code Exa 10.14 StnT14

```
clear all; clc;

disp("Scilab Code Ex 10.14 : ")

//Given:
r = 0.5; //cm
sigma_yield = 360; //MPa
T = 3.25; //kN/cm
A= (%pi*r^2);
P = 15; //kN
J = (%pi/2)*(r^4);
sigma_y_sqr = sigma_yield^2;

//Calculations:
sigma_x = -(P/A)*10;
sigma_y = 0;
tou_xy = (T*r*10)/J;

k = (sigma_x + sigma_y)/2;
R = sqrt(k^2 + (tou_xy^2));

sigma1 = k+R;
sigma2 = k-R;
l = sigma1 - sigma2;

```

```
//Maximum Shear Stress Theory:
test1 = abs(l);

if(test1 >= sigma_yield)

    printf("\n\nFailure occurs by Maximum Shear
      Stress Theory.');
end


//Maximum Distortion-Energy Theory:
test2 = (sigma1^2 - (sigma1*sigma2) + sigma2^2);


if(test2<sigma_y_sqr)

    printf("\n\nFailure will not occur by Maximum
      Distortion-Energy Theory.');
end
```

Chapter 11

Design of Beams and Shafts

Scilab code Exa 11.1 DBS1

```
clear all; clc;

disp("Scilab Code Ex 11.1 : ")

//Given:
sigma_allow = 170; //MPa
tou_allow = 100; //MPa

//Shear and Moment Diagrams:
V_max = 90; //kN
M_max = 120; //kNm

//Bending Stress:
S_reqd = (M_max*(10^3))/sigma_allow;

W = [60 67 64 74 80 100];
S = [1120 1200 1030 1060 984 987];

i = find(min(W));
S_chosen = S(i);
flag1 = 0;
```

```
flag2 = 0;

if (S_reqd<S_chosen)
   flag1 =1;
end

//Shear Stress:
d = 455; //mm
tw = 8; //mm
tou_avg = (V_max*10^3)/(d*tw);

if(tou_avg<tou_allow)
    flag2 =1;
end

if(flag1==1 & flag2==1)


     printf("\n\nUse a W460X60 standard shape.');
end

//----------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------
```

Scilab code Exa 11.2 DBS2

```
clear all; clc;

disp("Scilab Code Ex 11.2 : ")

//Given:
l = 200/1000; //m
t = 30/1000; //m
```

```
sigma_allow = 12; //MPa
tou_allow = 0.8; //MPa
V_nail = 1.50; //kN
l_bc = 2; //m
l_cd = 2; //m

//Shear and Moment Diagrams:
V_max = 1.5; //kN
M_max = 2; //kNm

//Bending Stress:
y1 = l/2;
A1 = l*t;
y2 = l+(t/2);
A2 = t*l;
y_dash = (y1*A1 + y2*A2)/(A1 + A2);

I1 = (t*l^3)/12 + (t*l*(y_dash - y1)^2);
I2 = (l*t^3)/12 + (t*l*(y2 - y_dash)^2);
I =I1 + I2;

c = y_dash;
sigma = (M_max*c)/(I);
flag1 = 0;
sigma_allow = sigma_allow*1000; //kPa

if(sigma<sigma_allow)
    flag1 = 1;
end

//Shear Stress:
y3 = y_dash/2;
A3 = y_dash*t;
Q = y3*A3;

tou = (V_max*Q)/(I*t);
tou_allow = tou_allow*1000; //kPa
flag2 =0;
```

```

if(tou<tou_allow)
    flag2 = 1;
end

//Nail Spacing:
y4a = (l+t-y_dash);
y4 = y4a - (t/2);
A4 = l*t;
Q4 = y4*A4;
V_bc = 1.5; //kN
V_cd = 1; //kN

q_bc = (V_bc*Q4)/I;
q_cd = (V_cd*Q4)/I;

s_bc = (V_nail)/(q_bc);
s_cd = (V_nail)/(q_cd);

chosen_bc = 150; //mm
chosen_cd = 250; //mm

if(flag1==1 & flag2==1)

    printf('\n\nThe design is safe in bending and shear.');
    printf('\nThe calculated nail spacing BC = %1.3f m',s_bc);
    printf('\nThe calculated nail spacing CD = %1.3f m',s_cd);
    printf('\nThe chosen nail spacing BC     = %1.0f mm',chosen_bc);
    printf('\nThe chosen nail spacing CD     = %1.0f mm',chosen_cd);
end

//
---------------------------------------------------------------------------
```

END

Scilab code Exa 11.3 DBS3

```
clear all; clc;

disp("Scilab Code Ex 11.3 : ")

//Given:
udl = 12; //kN/m
h_by_a = 1.5;
sigma_allow = 9; //MPa
tou_allow = 0.6; //MPa

//Shear and Moment Diagrams:
V_max = 20; //kN
M_max =10.67; //kNm

//Bending Stress:
S_reqd = (M_max)/(sigma_allow*1000);
c = h_by_a/2;
a_cube = (S_reqd*c*12)/(1.5^3); //S_reqd = I/c
a = a_cube^(1/3);


A = a*h_by_a*a;
tou_max = (1.5*V_max)/(A*1000);


if(tou_max>tou_allow)
    a_sqr = (3/2)*(V_max)/(h_by_a*tou_allow*1000);
    a =sqrt(a_sqr);
end
```

```

//Display:

    printf("\n\nThe smallest width for the laminated wooden beam = %1.3f m', a);

//------------------------------------------------------------------------------------ END ------------------------------------------------------------------------------------
```

Scilab code Exa 11.6 DBS6

```
clear all; clc;

disp("Scilab Code Ex 11.6 : ")

//Given:
tou_allow = 50*10^6; //MPa
T = 7.5; //Nm
R_ah = 150; //N
R_av = 475; //N
l_ac = 0.25; //m

mc = R_ah*l_ac;
m = R_av*l_ac;

M_c = sqrt(m^2 + mc^2);

k = sqrt(M_c^2 + T^2);
c1 = (2*k)/(%pi*tou_allow);
c = c1^(1/3);

d = 2*c*1000;

```

```
//Display:

    printf("\n\nThe smallest allowable diameter of
       the shaft = %1.1f mm', d);

//————————————————————————————————————————————————
   END
   ————————————————————————————————————————————————
```

Chapter 12

Deflection of Beams and Shafts

Scilab code Exa 12.6 DefBS6

```
clear all; clc;

disp("Scilab Code Ex 12.6 : ")

//Display:
    printf("\n\nRefer to the relation derived in the
        book.');

//-------------------------------------------------------------------------------
    END
    -------------------------------------------------------------------------------
```

Scilab code Exa 12.10 DefBS10

```
clear all; clc;

```

```
disp("Scilab Code Ex 12.10 : ")

//Given:
E = 200*10^6; //kN/m^2
I = 17*10^-6;//mm^4
l_ac = 2; //m
l_cF = 4; //m
l_Fb = 2; //m
l_cb = 6; //m
l_aF = 6; //m
l_ab = 8; //m
F = 16; //kN
R_b = (F*l_cb)/l_ab;
R_a = F - R_b;

mc = R_a*l_ac;
mf = R_b*l_Fb;
theta_ca = (0.5*l_ac*mc)/(E*I);

A1 = 0.5*l_aF*mf;
t1_ba = (l_Fb + l_aF/3)*(A1);

A2 = 0.5*l_Fb*mf;
t2_ba = (l_Fb*2*A2)/3;

t_ba = (t1_ba+t2_ba)/(E*I);

theta_c = (t_ba/l_ab)-(theta_ca);

//Display:

printf("\n\nThe slope at point C of the steel beam
      = %1.5f rad',theta_c);


//-----------------------------------------------------------------------------
   END
   ----------------------------------------------------------------------------
```

Scilab code Exa 12.12 DefBS12

```
clear all; clc;

disp("Scilab Code Ex 12.12 : ")

//Given:
E = 200; //kN/m^2
I = 50*10^6;//mm^4
l_ab = 4; //m
l_bc = 4; //m
l_ac = l_ab+l_bc;
R_a = -25; //kN
R_b = 50; //kN
R_c = 25; //kN

mb = R_a*l_ab;

//Moment-Area Theorem:

t_ca = (l_ab*0.5*l_ac*mb*(10^3)^3)/(E*I);
t_ba = (l_ab*0.5*l_ab*mb*(10^3)^3)/(E*I*3);

del_c = -t_ca + 2*t_ba;

//Display:

printf("\n\nThe displacement at point C for the
    steel overhanging beam      = %1.1f mm',del_c);


//--------------------------------------------------------------------------
    END
```

```
//
```

Scilab code Exa 12.13 DefBS13

```
clear all; clc;

disp("Scilab Code Ex 12.13 : ")

//Given:
w = 2; //kN/m
L = 8; //m
P = 8; //kN

//Calculations:
EI_theta_A1 = (3*w*L^3)/(128); //ThetaA1 = (3wL^3)
    /(128EI)
EI_nu_C1 = (5*w*L^4)/(768); //NuC1 = (5wL^4)/(768EI)

EI_theta_A2 = (P*L^2)/(16); //theta_A2 = (PL^2)/(16
    EI)
EI_nu_C2 = (P*L^3)/(48); //nu_C2 = (PL^3)/(48EI)

theta_A = EI_theta_A1 + EI_theta_A2;
nu_C = EI_nu_C1 + EI_nu_C2;

//Display:

printf('\n\nThe slope at A in terms of EI
                    = %1.0f/EI  kNm^2',theta_A);
printf('\nThe displacement at point C in terms of EI
       = %1.0f/EI  kNm^3',nu_C);

//
```

END
____

Scilab code Exa 12.14 DefBS14

```
clear all; clc;

disp("Scilab Code Ex 12.14 : ")

//Given:
w = 5; //kN/m
l_ab = 4; //m
l_bc = 2; //m
P = 10; //kN
M = w*l_ab; //kNm

//Calculations:
EI_theta_B1 = (w*l_ab^3)/(24); //ThetaB1 = (wL^3)
    /(24EI)
EI_nu_C1 = l_bc*EI_theta_B1;

EI_theta_B2 = (M*l_ab)/(3); //
EI_nu_C2 = l_bc*EI_theta_B2;

EI_nu_C3 = (P*l_bc^3)/(3); //nuC3 = (PL^3)/(24EI)

nu_C = -EI_nu_C1 + EI_nu_C2 + EI_nu_C3;

//Display:

printf('\n\nThe displacement at end C of the
    overhanging beam, in terms of EI   = %1.1f/EI kNm
    ^3',nu_C);

```

```
//
  ______________________________________________________________________
  END
  ______________________________________________________________________
```

Scilab code Exa 12.15 DefBS15

```
clear all; clc;

disp("Scilab Code Ex 12.15 : ")

//Given:
w = 4; //kN/m
l = 10; //m
l_bc =3; //m

//Calculations:
EI_theta_B = (w*l^3)/(24); //ThetaB1 = (wL^3)/(24EI)
EI_nu_B = (w*l^4)/(30); //nuB = (wL^4)/(30EI)

nu_C = EI_nu_B + (EI_theta_B*l_bc);

//Display:

printf('\n\nThe displacement at end C of the
    cantilever beam, in terms of EI  = %1.0f/EI kNm
    ^3',nu_C);

//
  ______________________________________________________________________
  END
  ______________________________________________________________________
```

Scilab code Exa 12.16 DefBS16

```
clear all; clc;

disp("Scilab Code Ex 12.16 : ")

//Given:
k = 45; //kN/m
F = 3; //kN
E = 200*10^6; //kPa
l_ab = 3; //m
l_ac = 1; //m
l_cb = 2; //m
I = 4.687*10^-6; //m^4
R_a = (F*l_cb)/(l_ab);
R_b = F-R_a;

//Calculations:
nu_a = (R_a)/k;
nu_b = (R_b)/k;

nu_c1 = nu_b + (l_cb/l_ab)*(nu_a - nu_b);
nu_c2 = ((F*l_ac*l_cb)*(l_ab^2 - l_ac^2 - l_cb^2))
    /(6*E*I*l_ab);

nu_c = nu_c1 + nu_c2;
nu_C = nu_c*1000;

//Display:

printf('\n\nThe vertical displacement of the force
    at C   = %1.3f mm',nu_C);

//
```

```
----------------------------------------------------------------------------------
END
----------------------------------------------------------------------------------
```

Scilab code Exa 12.21 DefBS21

```
clear all; clc;

disp("Scilab Code Ex 12.21 : ")

//Given:
l = 3; //m
l_af = l/2; //m
P = 8; //kN
w = 6; //kN/m

//Compatibility Equation:
EI_nu_b1 = (w*l^4)/8 + (5*P*l^3)/48; //nu_b = (wl^4)/8EI + (5Pl^3)/48EI
EI_nu_b2 = (l^3)/3;

B_y = EI_nu_b1 / EI_nu_b2;

//Display:

printf("\n\nThe reactions at roller support B   = %1.2f kN',B_y);


//----------------------------------------------------------------------------------
END
----------------------------------------------------------------------------------
```

Scilab code Exa 12.22 DefBS22

```
clear all; clc;

disp("Scilab Code Ex 12.22 : ")

//Given:
l = 8; //m
l_ab = l/2; //m
l_bc = l/2; //m
l_af = l_ab/2; //m
b = 12/1000; //m
w = 24; //kN/m
E = 200*10^6; //Kn/m^2
I = 80*10^-6;// m^4

//Compatibility Equation:
nu_b = (5*w*l^4)/(768*E*I); //nu_b = (5wl^4)/768EI
nu_b_byBy = (l^3)/(48*E*I); //nu_b' = (Pl^3)/48EI

B_y = (nu_b-b)/nu_b_byBy;

C_y = ((w*l_ab*l_af) - (B_y*l_ab))/l;

A_y = (w*l_ab - B_y - C_y);

//Display:

printf('\n\nThe reaction at A    = %1.0f kN',A_y);
printf('\nThe reaction at B    = %1.0f kN',B_y);
printf('\nThe reaction at C    = %1.0f kN',C_y);


//
```

END

Scilab code Exa 12.23 DefBS23

```
clear all; clc;

disp("Scilab Code Ex 12.23 : ")

//Given:
d = 12; //mm
E = 210; //GPa
I = 186*10^6; //mm^4
P = 40; //kN
l_bc = 3; //m
l_ab = 4;//m
l = 5; //m

//Compatibility Equation: nuB'' = nuB - nuB'
A = (%pi/4)*(d^2);

nuB1_by_Fbc = (l_bc*1000)/(A*E*1000); //nuB'' = PL/
    AE
nuB2 = (5*P*1000*(l_ab*1000)^3)/(48*E*1000*I); //nuB
     = (5PL^3)/(48EI)
nuB2_by_Fbc = ((l*1000)^3)/(3*E*1000*I); //nuB' = (
    PL^3)/(3EI)

F_bc = (nuB2)/(nuB1_by_Fbc + nuB2_by_Fbc );
F_bc = F_bc/1000; //in kN

//Display:

```

```
printf('\n\nThe force in the rod due to loading      =
    %1.3f kN',F_bc);


//
    ______________________________________________________________________________________________
    END
    ______________________________________________________________________________________________
```

Scilab code Exa 12.24 DefBS24

```
clear all; clc;

disp("Scilab Code Ex 12.24 : ")

//Given:
l_ab = 4; //m
l = l_ab/2;
w = 9; //kN/m

//Compatibility Equations:

EI_thetaB = (w*l_ab^3)/(48); //thetaB = (wL^3)/(48EI
    )
EI_nuB = (7*w*l_ab^4)/(384); //nuB = (7wl^4)/(384EI)

//Only redundant By applied:
EI_thetaB_by_By = (l_ab^2)/(2); //thetaB' = (PL^2)
    /(2EI)
EI_nuB_by_By = (l_ab^3)/(3); //nuB' = (PL^3)/(3EI)

//Only redundant Mb is applied:
EI_thetaB_by_Mb = l_ab; //thetaB'' = (ML)/(EI)
EI_nuB_by_Mb = (l_ab^2)/(2); //nuB'' = (ML^2)/(2EI)
```

```
//Solving for By and Mb using matrices:

A = [EI_thetaB_by_By EI_thetaB_by_Mb; EI_nuB_by_By
   EI_nuB_by_Mb ];
b = [-EI_thetaB; -EI_nuB ] ;
moments = A\b;

By = moments(1);
Mb = moments(2);

//Display:

printf('\n\nThe vertical force at B for the beam
   = %1.3f kN',By);
printf('\nThe moment at B for the beam             =
   %1.2f kNm',Mb);

//
   ________________________________________________________________
   END
   ________________________________________________________________
```

Chapter 13

Buckling of Columns

Scilab code Exa 13.1 BoC1

```
clear all; clc;

disp("Scilab Code Ex 13.1 : ")

//Given:
l = 7.2*1000; //mm
E = 200; //GPa
ro = 75; //mm
ri = 70; //mm
sigma_y = 250; //MPa

//Calculations:
I = (%pi/4)*(ro^4 - ri^4)
A = (%pi)*(ro^2 -ri^2);

Pcr = (%pi^2*(E*10^6)*I*(1000)^-2)/(l^2); //Pcr = (
    %pi^2*EI)/(l^2)

sigma_cr = (Pcr*1000)/A;

if(sigma_cr<sigma_y)
```

```

    printf("\n\nThe maximum allowable axial load
       that the column can support     = %1.1f kN',
       Pcr);
end

//-----------------------------------------------------------------------------
   END
   -----------------------------------------------------------------------------
```

Scilab code Exa 13.2 BoC2

```
clear all; clc;

disp("Scilab Code Ex 13.2 : ")

//Given:
E = 200; //GPa
I = 15.3*10^6; //mm^4
l= 4*1000; //mm
A = 5890; //mm^2
sigma_y = 250; //MPa

//Calculations:

Pcr = ((%pi^2)*E*10^6*I*1000^-2)/(l^2); //Pcr = (%pi
   ^2*EI)/(l^2)

sigma_cr = (Pcr*1000)/A;

if(sigma_cr>sigma_y)
    Pcr = (sigma_y*A);
    Pcr = Pcr/1000; //in kN
end
```

```

//Display:

printf("\n\nThe maximum allowable axial load that the column can support     = %1.1f kN',Pcr);

//---------------------------------------------------------------------------------------- END ----------------------------------------------------------------------------------------
```

Scilab code Exa 13.3 BoC3

```
clear all; clc;

disp("Scilab Code Ex 13.3 : ")

//Given:
E = 200;//GPa
Ix = 13.4*10^-6;
Iy = 1.83*10^-6;
l = 8;
KLx = 0.5*l; //m
KLy = 0.7*(l/2); //m
rx = 66.2; //mm
ry = 24.5; //mm

Pcrx = (%pi^2*E*10^6*Ix)/(KLx^2); //Pcr = (%pi^2*EI)/(l^2)
Pcry = (%pi^2*E*10^6*Iy)/(KLy^2); //Pcr = (%pi^2*EI)/(l^2)

Pcr = min(Pcrx,Pcry);
A = 3060; //mm^2
sigma_cr = Pcr/A;
```

```

sl_ratio_x = (KLx*1000)/(rx);
sl_ratio_y = (KLy*1000)/(ry);
s_ratio = max(sl_ratio_x, sl_ratio_y);

//Display:

printf("\n\nThe maximum load that the column can support without buckling      = %1.0f kN',Pcr);
printf("\nThe largest slenderness ratio                                       = %1.1f N/mm^2',s_ratio);
//----------------------------------------------------------------------------------------------END----------------------------------------------------------------------------------------------
```

Scilab code Exa 13.4 BoC4

```
clear all; clc;

disp("Scilab Code Ex 13.4 : ")

//Given:
E = 70;//GPa
Ix = 61.3*10^-6;
Iy = 23.2*10^-6;
l = 5;
KLx = 2*l; //m
KLy = 0.7*(l); //m
FS = 3; //Factor of safety
sigma_y = 215; //MPa


```

```
Pcrx = (%pi^2*E*10^6*Ix)/(KLx^2); //Pcr = (%pi^2*EI)
    /(l^2)
Pcry = (%pi^2*E*10^6*Iy)/(KLy^2); //Pcr = (%pi^2*EI)
    /(l^2)

Pcr = min(Pcrx,Pcry);
A = 7.5*10^-3; //mm^2
P_allow = Pcr/FS;
sigma_cr = (Pcr*10^-3)/A;


if(sigma_cr<sigma_y)

  printf("\n\nThe largest allowable load that the
      column can support      = %1.0f kN',P_allow);
end

//------------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------------
```

Scilab code Exa 13.5 BoC5

```
clear all; clc;

disp("Scilab Code Ex 13.5 : ")

//Given:
E = 200*10^3; //MPa
sigma_y = 250; //MPa
x1 = 50; //mm
y1 = 75; //mm
z1 = 4.5; //m
e = 25; //mm
```

```

Ix = (1/12)*x1*(y1*2)^3;
A = x1*2*y1;
rx = sqrt(Ix/A);
L = z1*1000;
KL = 1*L;

sl_ratio = KL/rx;
c = y1;
ec_r = e*c/(rx^2);
P_a = 83; //MPa
A = 7500; //mm^2
P = P_a*A;
P = P/1000; //in kN

k = (L/(2*rx))*(sqrt(P/(E*A)));
sigma_max = (P*1000/A)*(1+ec_r*sec(k)); //Secant Formula

l = sqrt((P*1000)/(E*Ix));
nu_max = e*(sec(l*L/2)-1);

//Display:

printf('\n\nThe allowable eccentric load that can be applied on the column  = %1.1fkN',P);
printf('\nThe maximum deflection of the column due to the loading           = %1.0f mm',nu_max);

//-----------------------------------------------------------------------------END-----------------------------------------------------------------------------
```

Scilab code Exa 13.6 BoC6

```
clear all; clc;

disp("Scilab Code Ex 13.6 : ")

//Given:
z1 = 4*1000; //mm
e = 200; //mm
KLy = 0.7*z1;
Iy = 20.4*10^6;
E = 200*10^3; //N/mm^2
sigma_y =250; //MPa

//y-y Axis Buckling:
Pcry = (%pi^2*E*10^6*Iy)/(KLy^2); //Pcr = (%pi^2*EI)
    /(l^2)
Pcry = Pcry/1000;

//x-x Axis Yielding:
Kx= 2;
KLx = Kx*z1;
c = (z1-KLy)/2;
rx = 89.9;

//Solved by applying the Secant Formula and then
    finding Px by trial and error:

trial_Px = 419.4; //kN

A = 7850;//mm^2
sigma = (trial_Px*1000)/(A);

if(Pcry>trial_Px & sigma<sigma_y)
printf('\n\nThe maximum eccentric load that the
    column can support = %1.1fkN',trial_Px);
printf('\nFailure will occur about the x-x axis.');

```

```
end

//
    ________________________________________________________________________
    END
    ________________________________________________________________________
```

Scilab code Exa 13.7 BoC7

```
clear all; clc;

disp("Scilab Code Ex 13.7 : ")

//Given:
d = 30; //mm
r = d/2;
L = 600; //mm
sigma_pl = 150;//MPa

//Calculations:
I = (%pi/4)*(r^4);
A = %pi*r^2;
r_gyr = sqrt(I/A);
K = 1;
sl_ratio = (K*L)/(r_gyr);
flag1 = 0;

//Assuming the critical stress is elastic:
E = 150/0.001;
sigma_cr1 = (%pi^2*E)/(sl_ratio^2); //Pcr = (%pi^2*
    EI)/(l^2)


if(sigma_cr1 > sigma_pl)
```

```
    Et = (270 - 150)/(0.002 - 0.001);
    sigma_cr2 = (%pi^2*Et)/(sl_ratio^2); //Pcr = (%pi^2*EI)/(l^2)

   if(sigma_cr2>150 & sigma_cr2<270)
       Pcr = sigma_cr2*A;
       Pcr = Pcr/1000; //in kN
        printf('\n\nThe critical load when used as a pin supported column = %1.0fkN',Pcr);

   end


end
```

Scilab code Exa 13.8 BoC8

```
clear all; clc;

disp("Scilab Code Ex 13.8 : ")

//Given:
E = 200*10^3; //MPa
sigma_y = 250; //MPa
L = 5*1000; //mm
K = 1;
A = 19000; //mm^2
rx = 117; //mm
ry = 67.4; //mm

//Calculations:
sl_ratio = (K*L)/(ry);
sl_ratio_c = sqrt((2*%pi^2*E)/(sigma_y));

if(sl_ratio>0 & sl_ratio<sl_ratio_c)
```

```
    num = (1 - (sl_ratio^2/(2*sl_ratio_c^2)))*
       sigma_y;
    denom1 = (5/3) + ((3/8)*sl_ratio/sl_ratio_c);
    denom2 = (sl_ratio^3)/(8*sl_ratio_c^3);
    sigma_allow = num/(denom1 - denom2);

    P = sigma_allow*A;
    P = P/1000;
       printf('\n\nThe largest load the pin
          supported column can safely bear = %1.0f
          kN',P);

   end

//
   ------------------------------------------------------------------
   END
   ------------------------------------------------------------------
```

Scilab code Exa 13.9 BoC9

```
clear all; clc;

disp("Scilab Code Ex 13.9 : ")

//Given:
P = 80; //kN
E = 210*10^3; //MPa
sigma_y = 360; //MPa
L = 5000; //mm
K = 0.5;

//Calculations:
I_by_d = (1/4)*(%pi)*(0.5^4);
```

```
A_by_d = (1/4)*(%pi);
r_by_d = sqrt(I_by_d/A_by_d);

sl_ratio_c = sqrt((2*%pi^2*E)/(sigma_y));
sigma_allow = (P*1000)/A_by_d;

d4 = (sigma_allow*23*(K*L)^2*16)/(12*%pi^2*E);
d = d4^(1/4);

//Check:
d = ceil(d);
r = d/4;
KL_r = (K*L)/r;


if(KL_r>sl_ratio_c & KL_r<200)
        printf('\n\nThe smallest diameter of the rod
            as allowed by AISC specification = %1.0
           fmm',d);

end

//
    ------------------------------------------------------------------
    END
    ------------------------------------------------------------------
```

Scilab code Exa 13.10 BoC10

```
clear all; clc;

disp("Scilab Code Ex 13.10 : ")

//Given:
```

```
L = 750; //mm
P = 60; //kN
sigma = 195; //N/mm^2
K = 1;

//Calculations:
b2 = (P*1000)/(2*sigma);
b = sqrt(b2);

A = 2*b*b;
Iy = (1/12)*(2*b*b^3);
ry = sqrt(Iy/A);

sl_ratio = (K*L)/(ry);



if(sl_ratio>12)
    b4 = (P*1000*2598.1^2)/(2*378125); //Eqn 13.26
    b = b4^(1/4);

    sl_ratio = (2598.1)/(b);
    w = 2*b;

    if(sl_ratio>55)
        printf('\n\nThe thickness of the bar  = %1.0
            fmm',b);
       printf('\nThe width of the bar      = %1.0fmm
           ',w);
   end
end

//
    ____________________________________________________________
    END
    ____________________________________________________________
```

Scilab code Exa 13.11 BoC11

```
clear all; clc;

disp("Scilab Code Ex 13.11 : ")

//Given:
P = 20*10^3; //N
y1 = 150; //mm
x1 = 40; //mm
A = (x1*y1);
d = 40;
K = 1;

//Eqn 13.29

L2 = (3718*A*d^2)/(P);
L = sqrt(L2);
KL_d = (K*L)/(d);

if(KL_d>26 & KL_d<=50)
    printf('\n\nThe greatest allowable length L as
        specified by the NFPA  = %1.0f mm',L);

end

//
    ________________________________________________________________________
    END
    ________________________________________________________________________
```

Scilab code Exa 13.12 BoC12

```
clear all; clc;

disp("Scilab Code Ex 13.12 : ")

//Given:
L = 1600; //mm
K = 2;
l = 80; //mm
b = 40; //mm
e = 20; //mm
c = 40; //mm

//Calculations:
I1 = (1/12)*(l*b^3);
A = l*b;
r = sqrt(I1/A);
sl_ratio = (K*L)/(r);

//Eqn 13.26:
sigma_allow = (378125)/(sl_ratio^2);

I2 = (1/12)*(b*l^3);
coefficient = (1/A) + (e*c)/I2;
sigma_max = sigma_allow;
P = sigma_max/coefficient;
P = P/1000;

//Display:

printf('\n\nThe load that can be supported if the
   column is fixed at its base = %1.2f kN',P);

//
   ------------------------------------------------------------------------
   END
   ------------------------------------------------------------------------
```

Scilab code Exa 13.13 BoC13

```
clear all; clc;

disp("Scilab Code Ex 13.13 : ")

//Given:
sigmaB_allow = 160; //MPa
E = 200; //GPa
sigma_y = 250; //MPa
K= 1;
A = 3790; //mm^2
Ix = 17.1*10^6; //mm^4
ry = 38.2; //mm
d = 157; //mm
c= d/2;
e = 750; //mm
L = 4000; //mm

sl_ratio = (K*L)/(ry);
sl_ratio_c = sqrt((2*%pi^2*E*1000)/(sigma_y));



if(sl_ratio<sl_ratio_c)
    num = (1 - (sl_ratio^2/(2*sl_ratio_c^2)))*
       sigma_y;
    denom1 = (5/3) + ((3/8)*sl_ratio/sl_ratio_c);
    denom2 = (sl_ratio^3)/(8*sl_ratio_c^3);
    sigmaA_allow = num/(denom1 - denom2);

    coeffP = 1/(sigmaA_allow*A) + (e*c)/(Ix*
       sigmaB_allow);
```

```
    P = 1/coeffP;

    sigA = (P/A)/(sigmaA_allow);
    P = P/1000; //in kN


    if(sigA < 0.15)
        printf('\n\nThe maximum allowable value of eccentric load  = %1.2f kN',P);
    end
end

//-------------------------------------------------------------------------------END-------------------------------------------------------------------------------
```

Scilab code Exa 13.14 BoC14

```

clear all; clc;

disp("Scilab Code Ex 13.14 : ")

//Given:
K = 2;
d= 60; //mm
L = 1200; //mm
e = 80; //mm
c = d;
A = 60*120; //mm^2
l = 60; //mm
b = 120;//mm

```

```
//Calculations:
sl_ratio = (K*L)/(d);

if(sl_ratio>26 & sl_ratio<50)
    sigma_allow = (3718)/(sl_ratio^2);
    sigma_max = sigma_allow;

    I = (1/12)*(l*b^3);
    coeffP = (1/A) + (e*c)/(I);
    P = sigma_max/coeffP;
    P = P/1000; //kN

     printf('\n\nThe eccentric load that can be
        supported = %1.2f kN',P);
end

//Answer given in the textbook varies.

//
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Chapter 14

Energy Methods

Scilab code Exa 14.1 EM1

```
clear all; clc;

disp("Scilab Code Ex 14.1 : ")

//Given:
sigma_y = 310; //N/mm^2
db =18; //mm
rb = db/2;
Ab = %pi*(rb^2);
E = 210*10^3; //N/mm^2
da1 = 20; //mm
ra1 = da1/2;
Aa1 = %pi*(ra1^2);
La1 = 50;//mm
La2= 6; //mm
da2 =18; //mm
ra2 = da2/2;
Aa2 = %pi*(ra2^2);
Lb = 56; //mm


```

```
//Bolt A:
P_max = sigma_y*Ab;
Uia = (P_max^2/(2*E))*(La1/Aa1 + La2/Aa2); //Ui = (N^2L)/(2AE)
Uia = Uia/1000;

//Bolt B:
Uib = (P_max^2/(2*E))*(Lb/Ab);
Uib = Uib/1000;

//Display:
    printf('\n\nThe greatest amount of strain energy absorbed by bolt A    = %1.3f J',Uia);
    printf('\nThe greatest amount of strain energy absorbed by bolt B    = %1.3f J',Uib);

 //---------------------------------------------------------------------------END---------------------------------------------------------------------------
```

Scilab code Exa 14.5 EM5

```
clear all; clc;

disp("Scilab Code Ex 14.5 : ")

//Given:
G = 75*10^9; //N/m^2
ro = 80/1000; //m
t = 15/1000; //m
ri = ro - t;
l1 = 750/1000; //m
l2 = 300/1000; //m
```

```
T1 = 40; //Nm
T2 =15; //Nm

//Calculations:

J = (%pi/2)*(ro^4 - ri^4);

//Eqn 14-22
U1 = (T1^2*l1)/(2*G*J);
U2 = (T2^2*l2)/(2*G*J);
Ui = U1 + U2;
Ui = Ui*10^6; //in micro Joule

//Display:

   printf('\n\nThe strain energy stored in the shaft
        = %1.0fX10^-6 J',Ui);

//
  ________________________________________________________________________________
  END
  ________________________________________________________________________________
```

Scilab code Exa 14.6 EM6

```
clear all; clc;

disp("Scilab Code Ex 14.6 : ")

//Given:
l_ab = 1; //m
l_bc = 2; //m
N_ab = 11.547*1000; //N
Nb = 20*1000; //N
```

```
Nc = -23.094*1000; //N
N_ac = -20*1000; //N
A = 100/(1000^2); //mm^2
E = 200*10^9; //N/m^2
P = 20*10^3;//N

//Eqn 14-26
P_by_2 = P/2;
l_ac = sqrt(l_bc^2 - l_ab^2);
del = 0;

N2= [N_ab^2 Nc^2 N_ac^2];
L = [l_ab l_bc l_ac];

for i = 1:3
    del = del + (N2(i)*L(i))/(2*A*E);
end

del_bh = del/P_by_2;
del_bh = del_bh*1000;

//Display:

printf('\n\nThe horizontal displacement at point B = %1.2fmm',del_bh);

//--------------------------------------------------------------------------------END--------------------------------------------------------------------------------
```

Scilab code Exa 14.8 EM8

```
clear all; clc;
```

```

disp(" Scilab Code Ex 14.8 : ")

//Given:
ro = 60; //mm
ri = 50; //mm
E = 70; //kN/mm^2
W = 600; //kN
L = 240; //mm
h = 0;

//Part a:

A = (%pi)*(ro^2 - ri^2);
del_st= (W*L)/(A*E);

//Part b:

del_max = del_st*(1 + sqrt(1 + 2*(h/del_st)));

//Display:

  printf('\n\nThe maximum displacement at the top of
     the pipe for gradually applied load   = %1.4f
   mm',del_st);
    printf('\nThe maximum displacement at the top of
       the pipe for suddenly applied load    = %1.4
      f mm',del_max);

//
  ------------------------------------------------------------------------
  END
  ------------------------------------------------------------------------
```

Scilab code Exa 14.9 EM9

```
clear all; clc;

disp("Scilab Code Ex 14.9 : ")

//Given:
W = 6000; //N
h = 50; //mm
E = 210*1000; //N/mm^2
L = 5000; //mm
I = 87.3*10^6; //mm^2

//Calculations:

del_st = (W*L^3)/(48*E*I);
del_max = del_st*(1 + sqrt(1 + 2*(h/del_st)));

c = 252/2;
sigma_max = (12*E*del_max*c)/(L^2);

//Display:

    printf('\n\nThe maximum bending stress in the steel beam    = %1.2f N/mm^2',sigma_max);
   printf('\nThe maximum deflection in the beam           = %1.3f mm',del_max);

//
    ------------------------------------------------------------------------
    END
    ------------------------------------------------------------------------
```

Scilab code Exa 14.10 EM10

```
clear all; clc;

disp("Scilab Code Ex 14.10 : ")

//Given:
m = 80*1000; //kg
v = 0.2; //m/s
l_ac = 1.5; //m
E = 200*10^9; //N/m^2
w = 0.2; //m
I = (1/12)*(w^4);
l_ab = 1000; //mm

//Calculations:
del_Amax = sqrt((m*v^2*l_ac^3)/(3*E*I));

P_max = (3*E*I*del_Amax)/(l_ac^3);
theta_A = (P_max*l_ac^2)/(2*E*I);
del_Amax = del_Amax*1000;
del_Bmax = del_Amax + (theta_A*l_ab);


//Display:

    printf('\n\nThe maximum horizontal displacement
       of the post at B due to impact    = %1.1f mm',
       del_Bmax);


//
    ________________________________________________________________________________
    END
    ________________________________________________________________________________
```

Scilab code Exa 14.11 EM11

```
clear all; clc;

disp("Scilab Code Ex 14.11 : ")

//Given:
A = 400*10^-6; //m^2
E = 200*10^6; //kN/m^2
P = 100; //kN

//Virtual Work Equation:

n = [0 0 -1.414 1];
N = [-100 141.4 -141.4 200];
L = [4 2.828 2.828 2];
del_cv = 0;

for i=1:4
    del_cv = del_cv + (n(i)*N(i)*L(i))/(A*E);
end

del_cv = del_cv*1000;

//Display:

    printf('\n\nThe vertical displacement of joint C
        of the steel truss   = %1.1f mm',del_cv);



//
  ________________________________________________________________________
  END
  ________________________________________________________________________
```

Scilab code Exa 14.12 EM12

```
clear all; clc;

disp("Scilab Code Ex 14.12 : ")

//Given:
A = 300*10^-6; //m^2
E = 210*10^6; //kN/m^2
P = 60; //kN
F_ac = 1.25; //kN

//Part a:

//Virtual Work Equation:

n = [0 1.25 0 -0.75];
N = [0 75 -60 -45];
L = [1.5 2.5 2 1.5];
del_ch = 0;

for i=1:4
    del_ch = del_ch + (n(i)*N(i)*L(i))/(A*E);
end

del_chA = del_ch*1000;

//Part b:

del_L = -6; //mm
del_chB = F_ac*del_L;

if(del_chB<0)

```

```

//Display:

    printf('\n\nThe horizontal displacement of joint
        C if a force is applied to B    = %1.3f mm',
       del_chA);
    printf('\nThe horizontal displacement of joint C
        if AC is fabricated short     = %1.1f mm',
       del_chB);
end



//
   ________________________________________________________________________________
   END
   ________________________________________________________________________________
```

Scilab code Exa 14.13 EM13

```
clear all; clc;

disp("Scilab Code Ex 14.13 : ")

//Given:
del_T = 60; //degree celcius
alpha = 12*10^-6; //per degree celcius
E = 200*10^6; //kN/m^2
A = 250*10^-6; //m^2
L = 4; //m

//Virtual Work Equation:
n = 1.155; //kN
N = -12; //kN
```

```
del_bh = (n*N*L)/(A*E) + (n*alpha*del_T*L);
del_bh = del_bh*1000;

//Display:

printf('\n\nThe horizontal displacement of joint B
    of the truss  = %1.2f mm',del_bh);

//
    ____________________________________________________________________
    END
    ____________________________________________________________________
```

Scilab code Exa 14.16 EM16

```
clear all; clc;

disp("Scilab Code Ex 14.16 : ")

//Given:
I = 175.8*10^-6; //m^4
E = 200*10^6; //kN/m^2
Ra = 1;//kN
l_ab = 3; //m
l_bc = 6; //m


//Virtual Work Equation:
m1 = -1; //*x1
M1 = -2.5; //*x1^3
m2 = -0.5; //*x2

x10 = 0;
```

```
x11 = l_ab;
I1 = integrate('m1*M1*(x1^4)','x1',x10,x11);

x20 = 0;
x21 = l_bc;
I2 = integrate('m2*123.75*(x2^2)','x2',x20,x21);

x20 = 0;
x21 = l_bc;
I3 = integrate(' -m2*22.5*(x2^3)','x2',x20,x21);

In = I1 + I2 + I3;
del_A = (In)/(E*I);
del_A = del_A*1000;


//Display:

    printf('\n\nThe displacement of point A of the
       steel beam   = %1.1f mm',del_A);

//
   ------------------------------------------------------------------
   END
   ------------------------------------------------------------------
```

Scilab code Exa 14.17 EM17

```
clear all; clc;

disp("Scilab Code Ex 14.17 : ")

//Given:
E = 210*10^3; //N/mm^2
```

```
P = 40*10^3;//N
A_ab = 1250; //mm^2
A_ac = 625; //mm^2
A_cd = 1250; //mm^2
A_bc = 625; //mm^2

N_by_P = [0 0 1.67 -1.33];
L = [4000 3000 5000 4000];
A = [A_ab A_bc A_ac A_cd];
N = zeros(4);
sum = 0;


for i =1:4
    N(i) = N_by_P(i)*P;
    num(i) = N(i)*N_by_P(i)*L(i);

end

for i = 1:4
    sum = sum + (num(i)/(A(i)*E)); //By Castigliano's Second theorem.
end

del_ch = sum;

//Display:
    printf('\n\nThe horizontal displacement of joint C of the steel truss    = %1.2f mm',sum);

//
-----------------------------------------------------------------------------------------------END-----------------------------------------------------------------------------------------------
```

Scilab code Exa 14.18 EM18

```
clear all; clc;

disp("Scilab Code Ex 14.18 : ")

//Given:
E = 200*10^6; //kN/m^2
P = 0;//N
A = 400*10^-6; //m^2

N_by_P = [0 0 -1.414 1];
L = [4 2.828 2.828 2];
N = [-100 141.4 -141.4 200];
sum = 0;


for i =1:4
    num(i) = N(i)*N_by_P(i)*L(i);
end

for i = 1:4
    sum = sum + (num(i)/(A*E)); //By Castigliano's
       Second theorem.
end

del_ch = sum*1000;

//Display:
  printf('\n\nThe vertical displacement of joint C
     of the steel truss   = %1.1f mm',del_ch);

//
  ____________________________________________________________________________
```

END

Scilab code Exa 14.21 EM21

```
clear all; clc;

disp("Scilab Code Ex 14.21 : ")

//Given:
I = 125*10^-6; //m^4
E = 200*10^6; //kN/m^2
Rc = 5;//kN
l_ac = 6; //m
l_cb = 4; //m


//Castigliano's Second Theorem:
m = 0.4/9;

x10 = 0;
x11 = l_ac;
I11 = integrate('4.4*(x1^2)','x1',x10,x11);
I12 = integrate('-m*(x1^4)','x1',x10,x11);
I1 = I11 + I12;

x20 = 0;
x21 = l_cb;
I21 = integrate('6*0.6*(x2^2)','x2',x20,x21);
I22 = integrate('18*0.6*(x2)','x2',x20,x21);
I2 = I21+I22;

In = I1 + I2 ;
del_cv = (In)/(E*I);
```

```
del_cv = del_cv*1000;


//Display:

    printf('\n\nThe vertical displacement of point C
        of the steel beam   = %1.1f mm',del_cv);

//
    ----------------------------------------------------------------------------------------------------
    END
    ----------------------------------------------------------------------------------------------------
```