

Scilab Textbook Companion for  
Advanced Mechanics of Materials  
by A. P. Boresi and R. J. Schmidt<sup>1</sup>

Created by  
Maloth Thirupathi  
BTECH  
Electrical Engineering  
IIT BOMBAY  
College Teacher  
None  
Cross-Checked by  
Spandana

May 2, 2015

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

# Book Description

**Title:** Advanced Mechanics of Materials

**Author:** A. P. Boresi and R. J. Schmidt

**Publisher:** John Wiley and Sons, USA

**Edition:** 6

**Year:** 2003

**ISBN:** 978-0-471-43881-6

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

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

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

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

# Contents

List of Scilab Codes	5
1 INTRODUCTION	8
2 THEORIES OF STRESS AND STRAIN	11
3 LINEAR STRESS STRAIN TEMPERATURE RELATION	19
4 INELASTIC MATERIAL BEHAVIOR	24
5 APPLICATIONS OF ENERGY METHODS	29
6 TORSION	36
7 BENDING OF STRAIGHT BEAMS	46
8 SHEAR CENTER FOR THIN WALL BEAM CROSS SECTIONS	56
9 CURVED BEAMS	62
10 BEAMS ON ELASTIC FOUNDATIONS	74
11 THE THICK WALL CYLLINDER	82
12 ELASTIC AND INELSTIC STABILITY OF COLUMNS	92
13 FLAT PLATES	94
14 STRESS CONCENTRATIONS	97

<b>15 FRACTURE MECHANICS</b>	<b>100</b>
<b>16 FATIGUE PROGRESSIVE FRACTURE</b>	<b>103</b>
<b>17 CONTACT STRESSES</b>	<b>109</b>

# List of Scilab Codes

Exa 1.1	Example1 . . . . .	8
Exa 1.2	Example2 . . . . .	9
Exa 1.3	Example3 . . . . .	10
Exa 2.1	Example1 . . . . .	11
Exa 2.2	Example2 . . . . .	13
Exa 2.4	Example4 . . . . .	14
Exa 2.5	Example5 . . . . .	16
Exa 2.7	Example7 . . . . .	17
Exa 3.1	Example1 . . . . .	19
Exa 3.7	Example7 . . . . .	19
Exa 3.8	Example8 . . . . .	21
Exa 4.1	Example1 . . . . .	24
Exa 4.2	Example2 . . . . .	26
Exa 4.3	Example3 . . . . .	27
Exa 5.1	Example1 . . . . .	29
Exa 5.2	Example2 . . . . .	30
Exa 5.6	Example6 . . . . .	31
Exa 5.7	Example7 . . . . .	32
Exa 5.9	Example9 . . . . .	32
Exa 5.10	Example10 . . . . .	33
Exa 5.12	Example12 . . . . .	34
Exa 6.1	Example1 . . . . .	36
Exa 6.2	Example2 . . . . .	37
Exa 6.3	Example3 . . . . .	38
Exa 6.4	Example4 . . . . .	39
Exa 6.5	Example5 . . . . .	40
Exa 6.7	Example7 . . . . .	41
Exa 6.8	Example8 . . . . .	42

Exa 6.9	Example9 . . . . .	43
Exa 6.10	Example10 . . . . .	44
Exa 7.1	Example1 . . . . .	46
Exa 7.2	Example2 . . . . .	47
Exa 7.3	Example3 . . . . .	48
Exa 7.4	Example4 . . . . .	49
Exa 7.5	Example5 . . . . .	50
Exa 7.6	Example6 . . . . .	51
Exa 7.7	Example7 . . . . .	52
Exa 7.8	Example8 . . . . .	53
Exa 7.9	Example9 . . . . .	54
Exa 8.1	Example1 . . . . .	56
Exa 8.2	Chapter8 Example 2 . . . . .	57
Exa 8.3	Example3 . . . . .	58
Exa 8.4	Example4 . . . . .	59
Exa 8.5	Example5 . . . . .	60
Exa 9.1	Example1 . . . . .	62
Exa 9.2	Example2 . . . . .	63
Exa 9.3	Example3 . . . . .	64
Exa 9.4	Example4 . . . . .	65
Exa 9.5	Example5 . . . . .	67
Exa 9.6	Example6 . . . . .	68
Exa 9.7	Example7 . . . . .	69
Exa 9.8	Example8 . . . . .	70
Exa 9.9	Example9 . . . . .	71
Exa 9.10	Example10 . . . . .	72
Exa 10.1	Example1 . . . . .	74
Exa 10.2	Example2 . . . . .	75
Exa 10.4	Example4 . . . . .	76
Exa 10.5	Example5 . . . . .	77
Exa 10.6	Example6 . . . . .	78
Exa 10.7	Example7 . . . . .	80
Exa 11.1	Example1 . . . . .	82
Exa 11.2	Example2 . . . . .	83
Exa 11.3	Example3 . . . . .	84
Exa 11.4	Example4 . . . . .	87
Exa 11.5	Example5 . . . . .	87
Exa 11.6	Example6 . . . . .	88

Exa 11.8	Example8 . . . . .	89
Exa 11.9	Example9 . . . . .	90
Exa 12.3	Example3 . . . . .	92
Exa 12.5	Example5 . . . . .	92
Exa 13.2	Example2 . . . . .	94
Exa 13.3	Example3 . . . . .	95
Exa 13.4	Example4 . . . . .	95
Exa 14.1	Example1 . . . . .	97
Exa 14.2	Example2 . . . . .	98
Exa 14.3	Example3 . . . . .	98
Exa 15.2	Example2 . . . . .	100
Exa 15.3	Example3 . . . . .	101
Exa 15.4	Example4 . . . . .	101
Exa 15.5	Example5 . . . . .	102
Exa 16.1	Example1 . . . . .	103
Exa 16.2	Example2 . . . . .	104
Exa 16.3	Example3 . . . . .	105
Exa 16.4	Example4 . . . . .	105
Exa 16.5	Example5 . . . . .	106
Exa 17.1	Example1 . . . . .	109
Exa 17.2	Example2 . . . . .	110
Exa 17.3	Example3 . . . . .	112



# Chapter 1

## INTRODUCTION

Scilab code Exa 1.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  // part (a)
5  a=700 // M Pa from figure 1.8
6  b=100 // M Pa from figure 1.8
7  m=1/6 // from figure 1.8
8  Y=450 // M Pa from figure 1.9
9  // calculations
10 sigma_u=a+m*b
11 // results
12 printf('\n part (a) \n')
13 printf(' The ultimate strength is sigma = %.f M Pa',
        sigma_u)
14 printf('\n and the yield strength is Y = %.f M Pa',Y
        )
15
16 // part (b)
17 c1=62 // from figure 1.8
18 d1=0.025 // from figure 1.8
19 c2=27 // from figure 1.10a
```

```

20 d2=0.04 // from figure 1.10a
21 // calculations
22 U_f1=c1*b*d1*10^6
23 U_f2=c2*b*d2*10^6
24 // results
25 printf('\n part (b)')
26 printf('\n The modulus of toughness for alloy steel
    is Uf = %.3e N/m^2',U_f1)
27 printf('\n and structural steel is Uf = %.3e N/m^2',
    U_f2)

```

---

#### Scilab code Exa 1.2 Example2

```

1  clc
2  // initialization of variables
3  clear
4  sigma=500 // Stress M Pa
5  eps=0.0073 // Strain
6  sigma_A=343 // M Pa from figure 1.9
7  eps_A=0.00172 // from figure 1.9
8  // part (a)
9  E=sigma_A/eps_A
10
11 // part (B)
12 eps_e=sigma/E
13 eps_p=eps-eps_e
14 // results
15 printf(' part (a) \n')
16 printf(' The modulus of elasticity of the rod is E =
    %.d G Pa',E/1000)
17 printf('\n part (b)')
18 printf('\n the permanent strain is = %.4f',eps_p)
19 printf('\n and the strain recovered is = %.4f',eps_e
    )

```

---

### Scilab code Exa 1.3 Example3

```
1  clc
2  // initialization of variables
3  clear
4  D=25 // kN
5  L=60 // kN
6  W=30 //kN
7  Y=250 // M Pa
8  safety=5/3 // AISC, 1989
9  // calculations
10 Q=(D+L+W)*10^3 // converted to N
11 A=safety*Q/Y
12 r=sqrt(A/%pi)+0.5 // additional 0.5 mm is for extra
    safety
13 d=2*r // diameter
14 // results
15 printf('Part (a) \n ')
16 printf('A rod of %.d mm in diameter, with a cross
    sectional area of %.d mm^2, is adequate',d,%pi*d
    ^2/4)
17 // The diameter is correct as given in the textbook.
    Area doesn't match due to rounding off error and
    partly because it's a design problem.
```

---

## Chapter 2

# THEORIES OF STRESS AND STRAIN

Scilab code Exa 2.1 Example1

```
1 clc
2 // initialization of variables
3 clear
4 sig_xx=-10 // MPa
5 sig_yy=30 // MPa
6 sig_xy=15 // MPa
7 sig_xz=0 // MPa
8 sig_yz=0 // MPa
9 sig_zz=0 //MPa
10 I1=sig_xx+sig_yy+sig_zz
11 I2=sig_xx*sig_yy-sig_xy^2+sig_zz*sig_xx-sig_xz^2+
    sig_zz*sig_yy-sig_yz^2
12 M=[sig_xx sig_xy sig_xz
13     sig_xy sig_yy sig_yz
14     sig_xz sig_yz sig_zz]
15 I3=det(M)
16 p=[1 -I1 I2 -I3]
17 sigma=roots(p)
18 printf('I1 = %d I2 = %d I3 = %d ',I1,I2,I3)
```

```

19 printf('\n Sigma_1 = %d  Sigma_2 = %d  Sigma_3 = %d
    ',sigma(1),sigma(3),sigma(2))
20 // We have:
21 // {S_xx-S S_xy S_xz
22 //   S_xy S_yy-S S_yz
23 //   S_xz S_yz S_zz-S}*{l m n}=0
24 // Substituting for Sigma_1
25 a1=sig_xx-sigma(1)
26 a2=sig_xy
27 a3=sig_xz
28 b1=sig_xy
29 b2=sig_yy-sigma(1)
30 b3=sig_yz
31 c1=sig_xz
32 c2=sig_yz
33 c3=sig_zz-sigma(1)
34 // You can solve it using the matrices but since the
    system is imcomplete we get
35 n1=0
36 // b1*l1+b2*m1=0
37 // This implies m1=-b1/b2*l1
38 // We also have l1^2+m1^2+n1^2=1
39 l1=1/sqrt(1+(b1/b2)^2)
40 m1=-b1/b2*l1
41 printf('\n N1 = %.4 fi + %.4 fj ',l1,m1)
42 printf('\n or \n N1 = %.4 fi + %.4 fj ',-l1,-m1)
43 // Similarly Substituting for Sigma_2
44 a1=sig_xx-sigma(3)
45 a2=sig_xy
46 a3=sig_xz
47 b1=sig_xy
48 b2=sig_yy-sigma(3)
49 b3=sig_yz
50 c1=sig_xz
51 c2=sig_yz
52 c3=sig_zz-sigma(3)
53 // here, l2 = m2 = 0
54 l2=0

```

```

55 m2=0
56 n2=sqrt(1)
57 printf('\n N2 = %.4fk ',n2)
58 printf('\n or \n N2 = %.4fk ',-n2)
59 // Similarly Substituting for Sigma_3
60 a1=sig_xx-sigma(2)
61 a2=sig_xy
62 a3=sig_xz
63 b1=sig_xy
64 b2=sig_yy-sigma(2)
65 b3=sig_yz
66 c1=sig_xz
67 c2=sig_yz
68 c3=sig_zz-sigma(2)
69 // On solving , we get
70 l3=1/sqrt(1+(b1/b2)^2)
71 m3=-b1/b2*l3
72 n3=0
73 printf('\n N3 = %.4 fi + %.4 fj ',l3,m3)
74 printf('\n or \n N3 = %.4 fi + %.4 fj ',-l3,-m3)

```

---

### Scilab code Exa 2.2 Example2

```

1  clc
2  // initialization of variables
3  clear
4  sig_xx=20 // MPa
5  sig_yy=10 // MPa
6  sig_xy=30 // MPa
7  sig_xz=-10 // MPa
8  sig_yz=80 // MPa
9  I2=-7800 // (MPa)^2
10 // part (a)
11 // Assume sig_zz=k and evaluate determinants to
    solve for k

```

```

12 det1=sig_xx*sig_yy-sig_xy^2
13 //det2=k*sig_xx-sig_xz^2
14 //det3=k*sig_yy-sig_yz^2
15 k=(I2-det1+sig_xz^2+sig_yz^2)/(sig_xx+sig_yy)
16 sig_zz=k
17 I1=sig_xx+sig_yy+sig_zz
18 M=[sig_xx sig_xy sig_xz
19     sig_xy sig_yy sig_yz
20     sig_xz sig_yz sig_zz]
21 I3=det(M)
22 // p=poly([1 -I1 I2 -I3], "x")
23 p=[1 -I1 I2 -I3]
24 sigma=roots(p)
25 // results
26 printf('\n part (a) \n')
27 printf(' The unknown stress component is = %.d M Pa
        and the stress invariants I1, I2, I3 are
        respectively %.d , %.d , %.d ',sig_zz,I1,I2,I3)
28 printf('\n The principal stresses are sigma1= %.3f ,
        sigma2=%.3f , sigma3=%.3f M Pa',sigma(2),sigma
        (3),sigma(1))

```

---

#### Scilab code Exa 2.4 Example4

```

1 clc
2 // initialization of variables
3 clear
4 sig_xx=120 // MPa
5 sig_yy=55 // MPa
6 sig_xy=-55 // MPa
7 sig_xz=-75 // MPa
8 sig_yz=33 // MPa
9 sig_zz=-85 // MPa
10 // Direction cosines at point A
11 lA=1/sqrt(3)

```

```

12 mA=1/sqrt(3)
13 nA=1/sqrt(3)
14 // Direction cosines at point B
15 lB=1/sqrt(2)
16 mB=1/sqrt(2)
17 nB=0
18 // calculations
19 I1=sig_xx+sig_yy+sig_zz
20 I2=sig_xx*sig_yy-sig_xy^2+sig_zz*sig_xx-sig_xz^2+
    sig_zz*sig_yy-sig_yz^2
21 M=[sig_xx sig_xy sig_xz
22     sig_xy sig_yy sig_yz
23     sig_xz sig_yz sig_zz]
24 I3=det(M)
25 p=[1 -I1 I2 -I3]
26 sig=roots(p)
27 sig=gsort(sig)
28 sigma(1)=sig(1)
29 sigma(3)=sig(2)
30 sigma(2)=sig(3)
31 // results
32 printf('\n The principal stresses are sigma1= %.3f ,
    sigma2=%.3f , sigma3=%.3f M Pa',sigma(1),sigma
    (2),sigma(3))
33 // Finding about the circles
34 C11=(sigma(2)+sigma(3))/2
35 C21=(sigma(1)+sigma(3))/2
36 C31=(sigma(1)+sigma(2))/2
37 C12=0
38 C22=0
39 C32=0
40 R1=(sigma(2)-sigma(3))/2
41 R2=(sigma(1)-sigma(3))/2
42 R3=(sigma(1)-sigma(2))/2
43 SnnA=lA^2*sigma(1)+mA^2*sigma(2)+nA^2*sigma(3)
44 SnsA=sqrt(lA^2*sigma(1)^2+mA^2*sigma(2)^2+nA^2*sigma
    (3)^2-SnnA^2)
45 SnnB=lB^2*sigma(1)+mB^2*sigma(2)+nB^2*sigma(3)

```



```

46 SnsB=sqrt(lB^2*sigma(1)^2+mB^2*sigma(2)^2+nB^2*sigma
    (3)^2-SnnB^2)
47 printf('\n The details of circles are given below')
48 printf('\n C1 : (%.2f M Pa, %.e) R1 = %.2f M Pa \n'
    ,C11,C12,R1)
49 printf('\n C2 : (%.2f M Pa, %.e) R2 = %.2f M Pa \n'
    ,C21,C22,R2)
50 printf('\n C3 : (%.2f M Pa, %.e) R3 = %.2f M Pa \n'
    ,C31,C32,R3)
51 printf('\n at point A')
52 printf('\n Normal stress = %.d M Pa and shear stress
    = %.2f M Pa',SnnA,SnsA)
53 printf('\n at point B')
54 printf('\n Normal stress = %.d M Pa and shear stress
    = %.2f M Pa',SnnB,SnsB)

```

---

### Scilab code Exa 2.5 Example5

```

1  clc
2  // initialization of variables
3  clear
4  sig_xx=80 // MPa
5  sig_yy=60 // MPa
6  sig_xy=20 // MPa
7  sig_xz=40 // MPa
8  sig_yz=10 // MPa
9  sig_zz=20 // MPa
10 // Direction cosines at point A
11 l=1/sqrt(6)
12 m=2/sqrt(6)
13 n=1/sqrt(6)
14 // calculations
15 SpX=sig_xx*l+sig_xy*m+sig_xz*n
16 SpY=sig_xy*l+sig_yy*m+sig_yz*n
17 SpZ=sig_xz*l+sig_yz*m+sig_zz*n

```

```

18 // result
19 printf('part (a)')
20 printf('\n The stress vector is = %.3f i + %.3f j +
    %.3f k', SpX, SpY, SpZ)
21 // part b
22 I1=sig_xx+sig_yy+sig_zz
23 I2=sig_xx*sig_yy-sig_xy^2+sig_zz*sig_xx-sig_xz^2+
    sig_zz*sig_yy-sig_yz^2
24 M=[sig_xx sig_xy sig_xz
25     sig_xy sig_yy sig_yz
26     sig_xz sig_yz sig_zz]
27 I3=det(M)
28 p=[1 -I1 I2 -I3]
29 sigma=roots(p)
30 tau_max=(sigma(1)-sigma(3))/2
31 tau_oct=sqrt((sigma(1)-sigma(2))^2+(sigma(1)-sigma
    (3))^2+(sigma(2)-sigma(3))^2)*1/3
32 n=tau_max/tau_oct
33 printf('\n part (b)')
34 printf('\n The principal stresses are sigma1= %.3f ,
    sigma2=%.3f , sigma3=%.3f M Pa', sigma(1), sigma
    (2), sigma(3))
35 printf('\n and maximum shear stress is = %d M Pa',
    tau_max)
36 printf('\n part (c)')
37 printf('\n octahedral shear stress is %.3f MPa ',
    tau_oct)
38 printf('\n Comparing tau_oct and tau_max, we see
    that \n')
39 printf(' tau_max = %.3f tau_oct ', n)

```

---

### Scilab code Exa 2.7 Example7

```

1 clc
2 // initialization of variables

```

```
3 clear
4 tau_max=160 //MPa
5 S_max=0
6 //S_min=-S_o
7 S_min=S_max-2*tau_max
8 S_o=-S_min
9 printf('part (a)')
10 printf('\n Sigma_o = %d MPa',S_o)
```

---

## Chapter 3

# LINEAR STRESS STRAIN TEMPERATURE RELATION

Scilab code Exa 3.1 Example1

```
1 clc
2 // initialization of variables
3 clear
4 // part (a)
5 E=72 // G Pa
6 v=0.33 // Poisson's ratio
7 h=2 // mm
8 R=600 // mm
9 //calculations
10 sig_cir=E*h/(2*(1-v^2)*R)
11 // results
12 printf('\n part (a) \n')
13 printf(' The maximum circumferential stress is %.d M
    Pa',sig_cir*10^3)
```

---

Scilab code Exa 3.7 Example7

```

1  clc
2  // initialization of variables
3  clear
4  tR=0.02 // t/R ration
5  E_A=69 //G Pa
6  v_A=0.33 // Poisson's ratio
7  alpha_A=21.6*10^-6 // /degree Celcius (Coefficient
    of expansion)
8  E_S=207 // G Pa
9  v_S=0.280
10 alpha_S=10.8*10^-6 // /degree Celcius (Coefficient
    of expansion)
11 // calculations
12 // Sig_LA=a*p+b*delT+c*sig_thS
13 // Sig_LS=v_S*Sig_thS+d*delT
14 E_S=E_S*10^9
15 E_A=E_A*10^9
16 a=1/tR*E_A/E_S
17 b=-2/3*alpha_S*E_S
18 c=-E_A/E_S
19 d=-alpha_S*E_S
20 // SigthS=e*p+f*delT
21 // SigthA=g*p+h*delT
22 e=37.16
23 f=0.8639*10^6
24 g=1/tR-e
25 h=-f
26 // results
27 p=689.4 // kPa
28 delT=100 // degree Celcius
29 p=p*10^3 // Pa
30 SigthA=g*p+h*delT
31 SigthS=e*p+f*delT
32 Sig_LA=a*p+b*delT+c*SigthS
33 Sig_LS=v_S*SigthS+d*delT
34 printf('Thus, for p = %.1f k Pa and delT = %.d
    degree celcius \n',p/10^3,delT)
35 printf(' SigthA = %.1f M Pa,    Sig_LA = %.d M Pa \n'

```

```

    ,SigthA/10^6, Sig_LA/10^6)
36 printf(' SigthS = %.1f M Pa,      Sig_LS = %.d M Pa',
    SigthS/10^6, Sig_LS/10^6)

```

---

### Scilab code Exa 3.8 Example8

```

1  clc
2  // initialization of variables
3  clear
4  // Material constants
5  Ex=14700 // M Pa
6  Ey=1000  // M Pa
7  Ez=735   // M Pa
8  Gxy=941  // M Pa
9  Gxz=1147 // M Pa
10 Gyz=103  // M pa
11 Vxy=0.292
12 Vxz=0.449
13 Vyz=0.39
14 // Stresses at a point
15 Sxx=7 // M pa
16 Syy=2.1 // M Pa
17 Szz=-2.8 //M Pa
18 Sxy=1.4 // M Pa
19 Sxz=0 //M Pa
20 Syz=0 // M Pa
21 // part (a)
22 th=1/2*atan(2*Sxy/(Sxx-Syy))*180/%pi
23 I1=Sxx+Syy+Szz
24 I2=Sxx*Syy-Sxy^2+Szz*Sxx-Sxz^2+Szz*Syy-Syz^2
25 M=[Sxx Sxy Sxz
26     Sxy Syy Syz
27     Sxz Syz Szz]
28 I3=det(M)
29 p=[1 -I1 I2 -I3]

```

```

30 S=roots(p)
31 // results
32 printf('Part (a) \n')
33 printf('The maximum principal stress is S1 = %.2f M
    Pa', S(1))
34 printf('\n and occurs in direction th = %.1f degrees
    ',th)
35 printf('\n and the intermediate principal stress S2
    = %.2f M Pa occurs in the direction th = %.1f
    degrees \n',S(3),th+90)
36 printf(' The minimum principal stress is S3 = Szz =
    %.1f M Pa', S(2))
37 Ex=Ex*10^6
38 Ey=Ey*10^6
39 Ez=Ez*10^6
40 Gxy=Gxy*10^6
41 Gxz=Gxz*10^6
42 Gyz=Gyz*10^6
43 // part (b) is to find strains
44 Exx=Sxx/Ex-Vxy*Syy/Ey-Vxz*Szz/Ez
45 Eyy=-Vxy*Sxx/Ex+Syy/Ey-Vyz*Szz/Ez
46 Ezz=-Vxz*Sxx/Ex-Vyz*Syy/Ey+Szz/Ez
47 Exy=Sxy/Gxy
48 Exz=Sxz/Gxz
49 Eyz=Syx/Gyz
50 printf('\n Part (b)')
51 printf('\n The strains are')
52 printf('\n Exx = %.2e , Eyy = %.2e , Ezz = %.4e',
    Exx,Eyy,Ezz)
53 printf('\n Exy = %.4e , Exz = %.2d , Eyz = %.2d',
    Exy,Exz,Eyz)
54 // Wrong Exx value in the textbook
55 th=1/2*atan(Exy/(Exx-Eyy))
56 th=th*180/%pi
57 th2=th+90
58 printf('\n part (c)')
59 printf('\n theta = %.2f or theta = %.2f degrees',th
    ,th2)

```

60 // Wrong theta too since Ex given in textbook is  
wrong

---



# Chapter 4

## INELASTIC MATERIAL BEHAVIOR

Scilab code Exa 4.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  P=170 //kN
5  A=645 // (mm)^2
6  // part (a)
7  E=211.4 // G Pa (from figure)
8  Y=252.6 // M Pa (from figure)
9  Beta=0.0799 // G Pa (from figure)
10 Ey=Y/E
11 // The stress strain law given is
12 // Sigma= E*eps for eps< Ey
13 // Sigma= (1-Beta)*Y + Beta*E*eps otherwise
14
15 // part (b)
16 th=atan(1.8/2.4) // radians
17 F=P/(2*cos(th))
18 F=F*10^3 //N
19 A=A*10^-6 //m^2
```

```

20 E=E*10^9 //Pa
21 Y=Y*10^6 //Pa
22 L=3.0 //m
23 Sigma=F/A
24 if(Sigma<Y)
25     eps=Sigma/E
26 else
27     eps=(Sigma-(1-Beta)*Y)/(Beta*E)
28 end
29 u=eps*L/cos(th)
30 u=u*10^3 //mm
31 // results
32 printf('part (b)\n')
33 printf(' Deflection = %.3f mm',u)
34
35 // part (c)
36 P=270 //kN
37 F=P/(2*cos(th))
38 F=F*10^3 //N
39 Sigma=F/A
40 if(Sigma<Y)
41     eps=Sigma/E
42 else
43     eps=(Sigma-(1-Beta)*Y)/(Beta*E)
44 end
45 u=eps*L/cos(th)
46 u=u*10^3 //mm
47 // results
48 printf('\n part (c)\n')
49 printf(' Deflection = %.3f mm for P = %.d kN',u,P)
50
51 P=300 //kN
52 F=P/(2*cos(th))
53 F=F*10^3 //N
54 Sigma=F/A
55 if(Sigma<Y)
56     eps=Sigma/E
57 else

```

```

58     eps=(Sigma-(1-Beta)*Y )/(Beta*E)
59 end
60 u=eps*L/cos(th)
61 u=u*10^3 //mm
62 // results
63 printf('\n Deflection = %.3f mm for P = %.d kN',u,P)

```

---

### Scilab code Exa 4.2 Example2

```

1  clc
2  // initialization of variables
3  clear
4  // Material properties
5  E=200 //GPa
6  A=100 //mm^2
7  Y1=500 //M Pa
8  Y2=250 // MPa
9  // calculations
10 E=E*10^9 // Pa
11 A=A*10^-6 //m^2
12 Y1=Y1*10^6 // Pa
13 Y2=Y2*10^6 //Pa
14 L_FG=1 //m
15 L_CD=2 // m
16 P1=Y2*A
17 e=P1*L_FG/(E*A)
18 e_FG=e
19 e_CD=e
20 P2=E*A*e_FG/L_FG
21 P3=E*A*e_CD/L_CD
22 Py=2*P1+2*P2+P3
23 //results
24 printf('part (a) \n')
25 printf(' Yield Load Py = %.1f kN and the
        displacement is %.2f mm',Py/10^3,e*10^3)

```

```

26
27 // part(b)
28 P4=Y1*A
29 e=P4*L_FG/(E*A)
30 P5=E*A*e/L_CD
31 P=2*P1+2*P4+P5
32 printf('\n part (b) \n')
33 printf(' Yield Load P = %.1f kN and the displacement
        is %.2f mm',P/10^3,e*10^3)
34 // Fully plastic load
35 P6=Y2*A*2
36 Pp=2*P1+2*P4+P6
37 e_CD=P6*L_CD/(E*A)
38 printf('\n Fully Plastic Load Pp = %.1f kN and the
        displacement is %.2f mm',Pp/10^3,e_CD*10^3)

```

---

### Scilab code Exa 4.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  // Stresses
5  Sxx=100 // MPa
6  Syy=-14 // MPa
7  Sxy=50 // MPa
8  Y=300 // MPa
9  // part (a)
10 Szz=0 // MPa
11 Syz=0 //MPa
12 Sxz=0 // MPa
13 // To calculate principal stresses
14 I1=Sxx+Syy+Szz
15 I2=Sxx*Syy-Sxy^2+Szz*Sxx-Sxz^2+Szz*Syy-Syz^2
16 M=[Sxx Sxy Sxz
17     Sxy Syy Syz

```

```

18     Sxz Syz Szz]
19 I3=det(M)
20 p=[1 -I1 I2 -I3]
21 Sigma=roots(p)
22 Smax=Sigma(1)
23 Smin=Sigma(2)
24 // Smax=max(Sigma)
25 // Smin=min(Sigma)
26 tau_max=Y/2
27 SF=tau_max*2/(Smax-Smin)
28 printf('part (a)\n')
29 printf(' SF = %.2f if the material obeys Tresca
        criterion ',SF)
30
31 // part (b)
32 SF=sqrt(2)*Y/sqrt((Smax^2)+(Smin^2)+(Smin-Smax)^2)
33 printf('\n part (b)')
34 printf('\n SF = %.2f if the material obeys von Mises
        criterion ',SF)

```

---

# Chapter 5

## APPLICATIONS OF ENERGY METHODS

Scilab code Exa 5.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  // part (b)
5  K1=2 //N/mm (K1=E1A1/L1)
6  K2=3 //N/mm (K2=E2A2/L2)
7  b1=400 // mm (b1=h)
8  h=400 // mm
9  b2=300 //mm
10 u=30 //mm
11 v=40 //mm
12 // calculations
13 // Units conversion
14 K1=K1*10^3
15 K2=K2*10^3
16 b1=b1*10^-3
17 b2=b2*10^-3
18 h=h*10^-3
19 u=u*10^-3
```

```

20 v=v*10^-3
21 L1=sqrt(b1^2+h^2)
22 L2=sqrt(b2^2+h^2)
23 N1=sqrt((b1+u)^2+(h+v)^2)-L1
24 N2=sqrt((b1+u)^2+(h+v)^2)
25 N3=sqrt((b2-u)^2+(h+v)^2)-L2
26 N4=sqrt((b2-u)^2+(h+v)^2)
27 P=K1*(b1+u)*N1/N2-K2*(b2-u)*N3/N4
28 Q=K1*(h+v)*N1/N2+K2*(h+v)*N3/N4
29 // results
30 printf('part (b)')
31 printf('\n P = %.1f N',P)
32 printf('\n Q = %.1f N',Q)

```

---

### Scilab code Exa 5.2 Example2

```

1 clc
2 // initialization of variables
3 clear
4 // Loads and stresses and dimensions
5 P=10 //kN
6 Q=30 //kN
7 S0=70 //MPa
8 eps0=0.001
9 b1=400 //mm
10 h=400 //mm
11 b2=300 //mm
12 A1=300 //mm^2
13 A2=300 //mm^2
14 // calculations
15 // Units conversion
16 P=P*10^3
17 Q=Q*10^3
18 S0=S0*10^6
19 b1=b1*10^-3

```

```

20 b2=b2*10^-3
21 h=h*10^-3
22 A1=A1*10^-6
23 A2=A2*10^-6
24 L1=sqrt(b1^2+h^2)
25 L2=sqrt(b2^2+h^2)
26 a=L1*(Q*b2+P*h)/(A1*S0*h*(b1+b2))
27 b=L2*(Q*b1-P*h)/(A2*S0*h*(b1+b2))
28 c=L1^2*eps0/(b1+b2)
29 d=L2^2*eps0/(b1+b2)
30 u=c*sinh(a)-d*sinh(b)
31 v=b2/h*c*sinh(a)+b1/h*d*sinh(b)
32 // results
33 printf('u = %.4f mm',u*10^3)
34 printf('\n v = %.4f mm',v*10^3)

```

---

#### Scilab code Exa 5.6 Example6

```

1  clc
2  // initialization of variables
3  clear
4  // Material constants
5  E=200 //GPa
6  G=77.5 // GPa
7  Lh=5 // Lh = L/h
8  // part (b)
9  rhs1=1.8*Lh*E/G
10 rhs2=7*12*Lh^3/16
11 LHS=1.8*Lh*E/G+7*12*Lh^3/16
12 e=rhs1/LHS*100
13 printf('The error in neglecting small terms is %.2f
    per cent',e)

```

---



### Scilab code Exa 5.7 Example7

```
1 clc
2 // initialization of variables
3 clear
4 // Specifications
5 T=2 //kN.m
6 E=72 // G Pa
7 G=27 // GPa
8 b=30 //mm
9 h=40 //mm
10 d=60 //mm
11 l1=400 //mm
12 l2=800 //mm
13 // calculations
14 E=E*10^9
15 G=G*10^9
16 b=b*10^-3
17 h=h*10^-3
18 d=d*10^-3
19 l1=l1*10^-3
20 l2=l2*10^-3
21 T=T*10^3 //N.m
22 Ix=b*h^3/12
23 J=%pi*d^4/32
24 thB= 2*l1^3/3*0.001^2*T/(E*Ix)+T*l2/(G*J)
25 printf('The rotation of shaft B is th = %.3f rad',
        thB)
26 // Wrong answer to an extent in the textbook
```

---

### Scilab code Exa 5.9 Example9

```
1 clc
2 // initialization of variables
3 clear
```

```

4 // specification
5 R=65 //mm
6 E=200 //GPa
7 G=77.5 //GPa
8 v=0.29
9 P=6 //kN
10 //calculations
11 R=R*10^-3
12 E=E*10^9
13 G=G*10^9
14 P=P*10^3
15 A=30^2*10^-6
16 I=30^4/12*10^-12
17 q_p1=3*pi*P*R/(4*E*A)+1.2*3*pi*P*R/(4*G*A)+(9*pi
    /4+2)*P*R^3/(E*I)
18 printf('part (a)')
19 printf('\n qp = %.4f mm',q_p1*10^3)
20 //part (b)
21 // if Un and Us are neglected
22 q_p2=(9*pi/4+2)*P*R^3/(E*I)
23 e=(q_p1-q_p2)/q_p1*100
24 printf('\n part (b)')
25 printf('\n error = %.2f per cent',e)

```

---

#### Scilab code Exa 5.10 Example10

```

1 clc
2 // initialization of variables
3 clear
4 // part (b)
5 // Specifications
6 P=150 //N
7 R=200 //mm
8 d=20 //mm
9 E=200 //GPa

```

```

10 G=77.5 //GPa
11 //calculations
12 R=R*10^-3
13 d=d*10^-3
14 E=E*10^9
15 G=G*10^9
16 r1=R+d/2
17 r2=R-d/2
18 A=314*10^-6
19 I=7850*10^-12 //m^4
20 Ax=3*pi/4*P*R/(E*A)
21 Sh=3*pi/4*1.33*P*R/(G*A)
22 M=(7*pi/4+1)*P*R^3/(E*I)
23 //qc=3*pi/4*P*R/(E*A)+3*pi/4*1.33*P*R/(G*A)+(7*pi
    /4+1)*P*R^3/(E*I)
24 qc=Ax+Sh+M
25 printf('qc = %.2f mm among which due to Axial is %.4
    f mm, %.4f mm is due to shear, and %.4f mm is due
    to moment',qc*10^3,Ax*10^3,Sh*10^3,M*10^3)
26 printf('\n which means The concentrations of axial
    loads and shear are negligible')

```

---

#### Scilab code Exa 5.12 Example12

```

1 clc
2 // initialization of variables
3 clear
4 // Material properties and dimensions
5 E=72 //G Pa
6 P=10 //kN
7 Q=5 //kN
8 Aab=150 //mm^2
9 Abc=900 //mm^2
10 Acd=900 //mm^2
11 Ade=900 //mm^2

```

```

12 Abd=150 //mm^2
13 Abe=150 //mm^2
14 Lab=2 //m
15 Lbc=2.5 //m
16 Lbd=1.5 //m
17 Lbe=2.5 //m
18 Lcd=2 //m
19 Lde=2 //m
20 // calculations
21 E=E*10^9
22 P=P*10^3
23 Q=Q*10^3
24 Aab=150
25 Aab=Aab*10^-6
26 Abc=Abc*10^-6
27 Acd=Acd*10^-6
28 Ade=Ade*10^-6
29 Abd=Abd*10^-6
30 Abe=Abe*10^-6
31 M=0
32 Nab=4/3*(Q+2*P)-5*M/(3*Lbe)
33 dNab=-5/(3*Lbe)
34 Nbc=-5/3*(Q+P)
35 dNbc=0
36 Nbd=Q
37 dNbd=0
38 Nbe=5*P/3-4/3*M/Lbe
39 dNbe=-4/(3*Lbe)
40 Ncd=-4*P/3+5/3*M/Lbe
41 dNcd=5/(3*Lbe)
42 Nde=Ncd
43 thBE=Nab*Lab*dNab/(E*Aab)+Nbc*Lbc*dNbc/(E*Abc)+Nbd*
    Lbd*dNbd/(E*Abd)+Nbe*Lbe*dNbe/(E*Abe)+2*Ncd*Lcd*
    dNcd/(E*Lcd)
44 printf('The rotation of member BE is %.5f rad',thBE)
45 // Wrong answer in the text

```

---

# Chapter 6

## TORSION

Scilab code Exa 6.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  // part (a)
5  a=22 //mm
6  b=25 //mm
7  T=500 //N m
8  // calculations
9  a=a*10^-3
10 b=b*10^-3
11 J=%pi*(b^4-a^4)/2
12 tau_max=T*b/J
13 printf(' part (a) \n')
14 printf(' Maximum shear stress in shaft = %.1f M Pa '
        ,tau_max/10^6)
15 // part (b)
16 G=77 //GPa
17 G=G*10^9
18 th=T/(G*J)
19 printf('\n part (b)')
20 printf('\n The angle of twist per unit length is = %
```

.4 f rad/m',th)

---

### Scilab code Exa 6.2 Example2

```
1  clc
2  // initialization of variables
3  clear
4  T=113 //Nm
5  L1=1 //m
6  L2=1.27 //m
7  Y=414 //MPa
8  G=77 //GPa
9  SF=2
10 // part (a)
11 T1=T*2
12 T2=T
13 Y=Y*10^6
14 G=G*10^9
15 tau_max=0.25*Y
16 r1=(2*T1/(%pi*tau_max))^(1/3)
17 d1=2*r1
18 r2=(2*T2/(%pi*tau_max))^(1/3)
19 d2=2*r2
20 inch=25.4 //mm
21 printf(' part (a) \n')
22 printf(' d1 = %.2 f mm    d2 = %.2 f mm',d1*10^3,d2
    *10^3)
23 printf('\n Since the dimensons are not standard, we
    choose d1 = %.1 f mm and d2 = %.2 f mm',inch,0.75*
    inch)
24 // part (b)
25 d1=inch*10^-3
26 r1=d1/2
27 d2=0.75*inch*10^-3
28 r2=d2/2
```

```

29 J1=%pi*r1^4/2
30 th1=T1/(G*J1)
31 J2=%pi*r2^4/2
32 th2=T2/(G*J2)
33 beta_c=L1*th1+L2*th2
34 bet_deg=beta_c*180/%pi
35 printf('\n part (b)')
36 printf('\n The angle of twist = %.3f rad = %.1f
    degrees',beta_c,bet_deg)
37 // Change is answer for US people convenience

```

---

### Scilab code Exa 6.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  tau_Y=190 //MPa
5  G=27 //GPa
6  L=2 //m
7  Do=60 //mm
8  Di=40 //mm
9  SF=2 // Factor of safety
10 // Angle of twist can't be greater than 0.2 rad
11 thM=0.2 //rad
12 Do=Do*10^-3
13 Di=Di*10^-3
14 G=G*10^9
15 tau_Y=tau_Y*10^6
16 J=%pi/2*((Do/2)^4-(Di/2)^4)
17 T=tau_Y*J*2/(Do*SF)
18 printf(' part (a)')
19 printf('\n Design torque T = %.3f kN.m',T/10^3)
20
21 // part (b)
22 T=G*J*thM/SF

```

```
23 printf('\n part (a)')
24 printf('\n Design torque limited by angle of twist
    is T = %.3f kN.m',T/10^3)
```

---

#### Scilab code Exa 6.4 Example4

```
1  clc
2  // initialization of variables
3  clear
4  // Material specifications
5  G=77.5 //GPa
6  // Following values of torsion are obtained from
   figure
7  Toa=-12.5 //kN
8  Tab=-8.5 //kN
9  Tbc=1.5 //kN
10 D1=10 //cm
11 D2 =5 //cm
12 D3 =D1 //cm
13 Loa=500 //mm
14 Lab=400 //mm
15 Lbc=300 //mm
16 // calculations
17 G=G*10^9
18 Toa=Toa*10^3
19 Tab=Tab*10^3
20 Tbc=Tbc*10^3
21 D1=D1*10^-2
22 D2=D2*10^-2
23 D3=D3*10^-2
24 Loa=Loa*10^-3
25 Lab=Lab*10^-3
26 Lbc=Lbc*10^-3
27 r1=D1/2
28 Joa=%pi*r1^4/2
```



```

29 tau0A=-Toa*D1/(2*Joa)
30 r2=D2/2
31 r3=D3/2
32 Jbc=%pi*r2^4/2
33 Jab=%pi*r3^4/2
34 tauBC=Tbc*D2/(2*Jbc)
35 tau=max(tau0A,tauBC)
36 printf('The maximum shear stress is = %.2f M Pa in
        segment OA',tau/10^6)
37 // part (b)
38 psiA=Toa*Loa/(G*Joa)
39 psiBA=Tab*Lab/(G*Jab)
40 psiB=psiA+psiBA
41 psiCB=Tbc*Lbc/(G*Jbc)
42 psiC=psiB+psiCB
43 printf('\n PsiA = %.5f rad   PsiB = %.5f rad   PsiC =
        %.5f rad ',psiA,psiB,psiC)

```

---

### Scilab code Exa 6.5 Example5

```

1  clc
2  // initialization of variables
3  clear
4  // Shaft specifications
5  Pi=100 //kW
6  f1=100 //Hz
7  f2=10 //Hz
8  tau_Y=220 //MPa
9  SF=2.5 // Safety factor
10 Po=100 //kW
11 //calculations
12 Pi=Pi*10^3
13 tau_Y=tau_Y*10^6
14 Po=Po*10^3
15 Tin=Pi/(2*%pi*f1)

```

```

16 Tout=Po/(2*%pi*f2)
17 Din=(16*SF*Tin/(tau_Y*%pi))^(1/3)
18 Dout=(16*SF*Tout/(tau_Y*%pi))^(1/3)
19 printf(' Din = %.2f mm and Dout = %.2f mm',Din
        *10^3,Dout*10^3)

```

---

### Scilab code Exa 6.7 Example7

```

1  clc
2  // initialization of variables
3  clear
4  // Flange specifications
5  T=5000 //Nm
6  b_f=266 //mm
7  d=779 //mm
8  t_w=16.5 //mm
9  t_f=30 //mm
10 G=200 // GPa
11 // calculations
12 b_f=b_f*10^-3
13 d=d*10^-3
14 t_w=t_w*10^-3
15 t_f=t_f*10^-3
16 G=G*10^9
17 // calculations
18 k1=0.308 // flange (b/h)<10
19 Jf=2*k1*b_f*t_f^3
20 k1=0.333 // web (b/h)>10
21 Jw=k1*(d-2*t_f)*t_w^3
22 J=Jf+Jw
23 // part (a)
24 hmax=0.015
25 tau_max=2*T*hmax/J
26 printf('part (a)\n')
27 printf(' Maximum shear stress is = %.2f MPa',tau_max

```

```

        /10^6)
28 // part (b)
29 th=T/(G*J)
30 printf('\n part (b)')
31 printf(' \n The angle of twist per unit length is =
        %.5f rad/m',th)

```

---

### Scilab code Exa 6.8 Example8

```

1  clc
2  // initialization of variables
3  clear
4  // Rod dimensions and material properties
5  b1=60 //mm
6  l1=3 //m
7  l2=1.5 //m
8  h1=40 //mm
9  b2=40 //mm
10 h2=30 //mm
11 G=77.5 //GPa
12 T1=750 //Nm
13 T2=400 //Nm
14 // calculations
15 b1=b1*10^-3
16 h1=h1*10^-3
17 b2=b2*10^-3
18 h2=h2*10^-3
19 G=G*10^9
20 // for the left portion of the rod
21 k1l=0.196
22 k2l=0.231
23 // for the right portion of the rod
24 k1r=0.178
25 k2r=0.223
26 T=T1+T2

```

```

27 tau_maxL=T/(k2l*b1*(h1)^2)
28 tau_maxR=T2/(k2r*b2*(h2)^2)
29 tau_max=max(tau_maxL,tau_maxR)
30 J1=b1*h1^3/12+h1*b1^3/12
31 J2=b2*h2^3/12+h2*b2^3/12
32 bet=T*l1/(G*J1)+T2*l2/(G*J2)
33 printf(' The maximum shear stress is = %.1f MPa',
        tau_max/10^6)
34 printf('\n twist = %.4f rad',bet)
35 //wrong answer for twist in the text

```

---

#### Scilab code Exa 6.9 Example9

```

1  clc
2  // initialization of variables
3  clear
4  Do=22 //mm
5  Di=18 //mm
6  Dm=20 //mm
7  tD=0.1 // t/D
8  //part (a)
9  tau=70 //MPa
10 G=77.5 //GPa
11 //calculations
12 Do=Do*10^-3
13 Di=Di*10^-3
14 Dm=Dm*10^-3
15 tau=tau*10^6
16 G=G*10^9
17 A=%pi*Dm^2/4
18 t=Dm*tD
19 T1=2*A*tau*t
20 th1=tau*%pi*Dm/(2*G*A)
21 J=%pi/32*(Do^4-Di^4)
22 r=Dm/2

```

```

23 T2=tau*J/r
24 th2=tau/(G*r)
25 printf('part (a)\n')
26 printf(' Using formula_1 T = %.2f Nm theta = %.7f
      rad/mm ',T1,th1*10^-3)
27 printf('\n Using formula_2 T = %.2f Nm theta = %.7f
      rad/mm ',T2,th2*10^-3)
28 //part (b)
29 h=1 //mm
30 h=h*10^-3
31 b=10*pi
32 b=b*10^-3
33 T=8*b*h^2*tau/3
34 th=tau/(2*G*h)
35 printf('\n part (b)')
36 printf('\n T = %.3f N.m      theta = %.7f rad/mm ',T,
      th*10^-3)

```

---

### Scilab code Exa 6.10 Example10

```

1  clc
2  // initialization of variables
3  clear
4  G=26 //GPa
5  tau_max=40.0 //MPa
6  t1=4.5 //mm
7  t3=1.5 //mm
8  t2=3 //mm
9  l1=3*60 //mm
10 l3=60 //mm
11 r2=30 //mm
12
13 //calculations
14 // 1 indicates coefficient of q1
15 // 2 indicates coefficient of q2

```

```

16
17 l2=r2*%pi
18 G=G*10^3
19 A1=l3^2
20 A2=%pi*r2^2/2
21 T1=2*A1
22 T2=2*A2
23 tha1=l1/t1+l3/t3
24 tha1=tha1/(2*G*A1)
25 tha2=-l3/t3
26 tha2=tha2/(2*G*A1)
27 thb1=-l3/t3
28 thb1=thb1/(2*G*A2)
29 thb2=l2/t2+l3/t3
30 thb2=thb2/(2*G*A2)
31 // Since tha=thb
32 Qr=(thb2-tha2)/(tha1-thb1)
33 printf('q1/q2 = %.3f ',Qr)
34 q2=tau_max*t2
35 q1=Qr*q2
36 qdif=q1-q2
37 tau_1=q1/t1
38 tau_2=q2/t2
39 tau_3=qdif/t3
40 T=2*A1*q1+2*A2*q2
41 th=tha1*q1+tha2*q2
42 printf('\n T = %.3f kN.m',T/10^6)
43 printf('\n theta = %.4f rad/m',th*10^3)

```

---

# Chapter 7

## BENDING OF STRAIGHT BEAMS

Scilab code Exa 7.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  E=200 //G Pa
5  Y=250 //M Pa
6  SF=1.9
7  w=1 //kN/m
8  L=3 //m
9  S_max=Y
10 // Calculations
11 E=E*10^9
12 Y=Y*10^6
13 w=w*10^3
14 Mx=-SF*w*L^2/2
15 S_max=S_max*10^6
16 k=2 // c_max=h/k
17 //Formula to be used
18 // S_max=abs(Mx)*c_max/Ix
19 // Note that c_max=h/2 and Ix=h^4/24
```

```
20 h=(abs(Mx)*24/(k*S_max))^(1/3)
21 printf('h = %.4f m',h)
```

---

### Scilab code Exa 7.2 Example2

```
1  clc
2  // initialization of variables
3  clear
4  P1=1.5 //kN
5  P2=4.5 //kN
6  // part (a)
7  A=1000 //mm^2
8  A1=500 //mm^2
9  A2=500 //mm^2
10 // calculation
11 A=A*10^-6
12 A1=A1*10^-6
13 A2=A2*10^-6
14 y1=25*10^-3
15 y2=55*10^-3
16 c1=(A1*y1+A2*y2)/A
17 c2=60*10^-3-c1 // c1+c2=60 mm
18 y_1=c1-25*10^-3
19 y_2=c2-5*10^-3
20 b1=50*10^-3
21 h1=10*10^-3
22 h2=50*10^-3
23 b2=10*10^-3
24 Ix=1/12*b1*h1^3 + A1*y_1^2 + 1/12*b2*h2^3 + A2*y_2^2
25 printf('part (a)')
26 R1=2550 //N
27 Vy=750 //N
28 Mx=975 //N.m
29 S_zzT=Mx*c1/Ix
30 S_zzC=Mx*(-c2)/Ix
```



```

31 printf('\n Maximum Tensile stress = %.1f MPa',S_zzT
    /10^6)
32 printf('\n Maximum Compressive stress = %.1f MPa',
    S_zzC/10^6)

```

---

### Scilab code Exa 7.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  P=12 //kN
5  Phi=%pi/3
6  // calculations
7  L=3 //m
8  P=12 //kN
9  A=10000 //mm^2
10 Ix=39.69*10^6 //mm^4
11 yo=82 //mm
12 Iy=30.73*10^6 //mm^4
13 Ixy=0
14 P=P*10^3
15 Ix=Ix*10^-12
16 Iy=Iy*10^-12
17 alpha=atan(-Ix/(Iy*tan(Phi)))
18 M=-L*P
19 Mx=M*sin(Phi)
20 yA=-118*10^-3 //m
21 xA=-70*10^-3 //m
22 xB=-xA
23 yB=82*10^-3 //m
24 S_A=Mx*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
25 S_B=Mx*(yB-xB*tan(alpha))/(Ix-Ixy*tan(alpha))
26 printf(' Sigma A = %.1f M Pa \n',S_A/10^6)
27 printf(' Sigma B = %.1f M Pa',S_B/10^6)

```

---

#### Scilab code Exa 7.4 Example4

```
1  clc
2  // initialization of variables
3  clear
4  P=4 //kN
5  L=1.2 //m
6  A=1900 //mm^2
7  Ix=2.783*10^6 //mm^4
8  Iy=1.003*10^6 //mm^4
9  Ixy=-0.973*10^6 //mm^4
10 P=P*10^3
11 Ix=Ix*10^-12
12 Iy=Iy*10^-12
13 Ixy=Ixy*10^-12
14 A=1900 //mm^2
15 xo=19.74 //mm
16 yo=39.74 //mm
17 Phi=2*%pi/3
18 Nr=Ixy-Ix/tan(Phi)
19 Dr=Iy-Ixy/tan(Phi)
20 alpha=atan(Nr/Dr)
21 M=L*P
22 Mx=M*sin(Phi)
23 yA=39.74*10^-3 //m
24 xA=-60.26*10^-3 //m
25 xB=19.74*10^-3
26 yB=-80.26*10^-3 //m
27 S_A=Mx*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
28 S_B=Mx*(yB-xB*tan(alpha))/(Ix-Ixy*tan(alpha))
29 printf('part (a)')
30 printf('\n Sigma A = %.1f M Pa \n',S_A/10^6)
31 printf(' Sigma B = %.1f M Pa',S_B/10^6)
32
```

```

33 // part (b)
34 th=1/2*atan(-2*Ixy/(Ix-Iy))
35 th1=0.415 //rad
36 th2=-1.156 //rad
37 IX=Ix*(cos(th1))^2+Iy*(sin(th1))^2-2*Ixy*sin(th1)*
    cos(th1)
38 IY=Ix+Iy-IX
39 Phi=2*%pi/3-th1
40 alphA=-IX/(IY*tan(Phi))
41 alpha=alphA+th1
42 XA=xA*cos(th1)+yA*sin(th1)
43 YA=yA*cos(th1)-xA*sin(th1)
44 XB=xB*cos(th1)+yB*sin(th1)
45 YB=yB*cos(th1)-xB*sin(th1)
46 MX=M*sin(Phi)
47 MY=-M*cos(Phi)
48 S_A=MX*YA/IX-MY*XA/IY
49 S_B=MX*YB/IX-MY*XB/IY
50 printf('\n part (b)')
51 printf('\n Sigma A = %.1f M Pa \n',S_A/10^6)
52 printf(' Sigma B = %.1f M Pa',S_B/10^6)

```

---

### Scilab code Exa 7.5 Example5

```

1 clc
2 // initialization of variables
3 clear
4 A=3085.9 //mm^2
5 Ix=29.94e-6 //m^4
6 Iy=4.167e-6 //m^4
7 Ixy=0
8 ybar=207.64 //mm
9 tau_max=165e6 //Pa
10 // calculations
11 A=A*1e-6

```

```

12 ybar=ybar*1e-3
13 Mxk=-6.1*cos(%pi/6) // Mx=Mxk*P
14 Myk=-6.1*sin(%pi/6) //My=Myk*P
15 // Equation to be followed
16 // S_zz=Mx*y/Ix-My*x/Iy
17 // At A x=100 mm y=-92.36 mm
18 x=100e-3
19 y=-92.36e-3
20 S_zzA=Mxk*y/Ix-Myk*x/Iy //Sigma_zz=S_zz*P
21 // At B x=-100 mm y=-92.36 mm
22 x=-100e-3
23 y=-92.36e-3
24 S_zzB=Mxk*y/Ix-Myk*x/Iy //Sigma_zz=S_zz*P
25 // At C x=-3.125 mm y=207.64 mm
26 x=-3.125e-3
27 y=207.64e-3
28 S_zzC=Mxk*y/Ix-Myk*x/Iy //Sigma_zz=S_zz*P
29 // To find P
30 P=2*tau_max/max(S_zzA,S_zzB,S_zzC)
31 printf('P = %.2f kN',P/10^3)

```

---

### Scilab code Exa 7.6 Example6

```

1 clc
2 // initialization of variables
3 clear
4 P=35 //kN
5 Phi=5*%pi/9
6 E=72e9 //Pa
7 L=3 //m
8 Ix=39.69*10^6 //mm^4
9 Iy=30.73*10^6 //mm^4
10 Ixy=0
11 // calculations
12 P=P*1e3

```

```

13 Ix=Ix*10^-12
14 Iy=Iy*10^-12
15 alpha=atan(-Ix/(Iy*tan(Phi)))
16 M=P*L/4
17 Mx=M*sin(Phi)
18 yA=-118*10^-3 //m
19 xA=70*10^-3 //m
20 xB=-xA
21 yB=82*10^-3 //m
22 S_comp=Mx*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
23 S_tens=Mx*(yB-xB*tan(alpha))/(Ix-Ixy*tan(alpha))
24 printf(' Tensile strength = %.1f M Pa \n',S_tens
        /10^6)
25 printf(' Compressive Strength = %.1f M Pa',S_comp
        /10^6)
26 v=P*L^3*sin(Phi)/(48*E*Ix)
27 u=-v*tan(alpha)
28 delta=sqrt(u^2+v^2)
29 printf('\n The total deflection is %.2f mm',delta
        *10^3)

```

---

#### Scilab code Exa 7.7 Example7

```

1 clc
2 // initialization of variables
3 clear
4 L=3 //m
5 Ix=56.43e6 //mm^4
6 Iy=18.11e6 //mm^4
7 Ixy=22.72e6 //mm^4
8 Phi=%pi/3
9 E=200e9 //Pa
10 Y=300e6 //Pa
11 // calculations
12 Ix=Ix*10^-12

```

```

13 Iy=Iy*10^-12
14 Ixy=Ixy*10^-12
15 yA=-120*10^-3 //m
16 xA=-91*10^-3 //m
17 Nr=Ixy-Ix/tan(Phi)
18 Dr=Iy-Ixy/tan(Phi)
19 alpha=atan(Nr/Dr)
20 // M=-L*P To know P we do the following
21 Mxk=-L*sin(Phi)//Mx=Mxk*P
22 P=Y*(Ix-Ixy*tan(alpha))/(Mxk*(yA-xA*tan(alpha)))
23 printf('P = %.2f kN \n',P/10^3)
24 v=P*L^3*sin(Phi)/(3*E*(Ix-Ixy*tan(alpha)))
25 u=-v*tan(alpha)
26 delta=sqrt(u^2+v^2)
27 printf(' deflection = %.2f mm',delta*10^3)
28 // Wrong calculation starting from v in Textbook

```

---

### Scilab code Exa 7.8 Example8

```

1 clc
2 // initialization of variables
3 clear
4 Ix=937e+06 //mm^4
5 Iy=18.7e+6 //mm^4
6 Ixy=0
7 yA=305 //mm
8 xA=90.5 //mm
9 Phi=1.5533 //rad
10 // calculations
11 Ix=Ix*10^-12
12 Iy=Iy*10^-12
13 Ixy=Ixy*10^-12
14 yA=yA*10^-3 //m
15 xA=xA*10^-3 //m
16 alpha=atan(-Ix/(Iy*tan(Phi)))

```

```

17 Mxk=sin(Phi) // Mx=Mxk*M
18 Sigma_Ak1=Mxk*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
19 //Sigma_A=Aigma_Ak*M
20 // When the plane of the loads coincide with the y
    axes
21 Sigma_Ak2=yA/Ix
22 ratio=Sigma_Ak1/Sigma_Ak2
23 percent=(ratio-1)*100
24 printf('alpha = %.3f rad',alpha)
25 printf('\n The maximum stress in the beam is
    increased %.1f percent when the plane of the
    loads is merely 1 degree from the symmetrical
    vertical plane',percent)
26 // Wrong alpha given in the textbook

```

---

### Scilab code Exa 7.9 Example9

```

1 clc
2 // initialization of variables
3 clear
4 Y=280 //MPa
5 AB=40 //mm
6 BC=60 //mm
7 // calculations
8 Y=Y*10^6
9 alpha=atan(BC/AB)
10 C11=20/3 //mm
11 C12=-10 //mm
12 C21=-20/3 //mm
13 C22=10 //mm
14 Beta=atan((C11-C21)/(C22-C12))
15 Phi=%pi/2+Beta
16 d=sqrt((AB/2-C11)^2+(BC/2-C22)^2)
17 d=d*10^-3 //m
18 At=1/2*AB*BC/2*10^-6

```

```
19 Mp=At*Y*d
20 printf('Mp = %.3 f kN.m',Mp/10^3)
```

---



## Chapter 8

# SHEAR CENTER FOR THIN WALL BEAM CROSS SECTIONS

Scilab code Exa 8.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  t=4 //mm
5  // calculations
6  l1=100 //mm See figure
7  l2=50 //mm See figure
8  ybar=125 //mm
9  t=t*10^-3
10 ybar=ybar*10^-3
11 l1=l1*10^-3
12 l2=l2*10^-3
13 Ix=2*t*(2*(l1+l2))^3/12-t*(2*l1)^3/12
14 qAk=l1*t*ybar // qA=qAk*V
15 qBk=qAk+l1*t*l1/2
16 qave=qAk+2/3*(qBk-qAk)
17 F2k=200*qave*10^-3 //F2=F2k*V
```

```

18 D0=100/tan(30*%pi/180) // from figure
19 // Now we need to solve the following equation
20 // (D0-e)*V=D0*F2
21 e=D0*(1-F2k/Ix)
22 printf('e = %.1f mm',e)

```

---

### Scilab code Exa 8.2 Chapter8 Example 2

```

1  clc
2  // initialization of variables
3  clear
4  // Defining the legs
5  a=50 //mm Top horizontal leg
6  b=100 //mm Verical leg
7  c=100 //mm bottom leg
8  t=4 //mm
9  Ix=1.734e6 //mm^4
10  Iy=0.876e6 //mm^4
11  Ixy=-0.5e6 //mm^4
12  I=[Ix Ixy
13     Ixy Iy]
14  theta=1/2*atan(-2*Ixy/(Ix-Iy))
15  Q=[cos(theta) -sin(theta)
16     sin(theta)  cos(theta)]
17  I_1=Q*I*Q' // I_1=[IX IXY| IXY IY]
18  // Finding out the centroidal coordinates
19  // We have x_bar = Summation(Ai*Xi)/Summation(Ai)
20  // We take D as reference
21  Aa=a*t
22  Ab=b*t
23  Ac=c*t
24  A=Aa+Ab+Ac
25  x_D=((Ac*c/2)+(Aa*a/2))/A
26  y_D=((Ab*b/2)+(Aa*b))/A
27  //Finding out B coordinates

```

```

28 xb=a-x_D
29 yb=b-y_D
30 x=[xb;yb]
31 X=Q'*x //New coordinates of B in transformed system
32 function y=f(1),
33     y=t*I_1(1)*(X(2)+1/2*I*sin(theta)),
34 endfunction
35 F3=intg(0,a,f) // This is the coefficient of VY
36 e_X=b*F3
37 printf('eX = %.2 f mm',e_X)
38 // To find eY
39 function y1=g(1),
40     y1=t*I_1(4)*(X(1)-1/2*I*cos(theta)),
41 endfunction
42 F3=intg(0,a,g) // This is the coefficient of VX
43 e_Y=b*F3
44 printf('\n eY = %.2 f mm',e_Y)
45 XC=Q'*[x_D
46     y_D]
47 XC=XC+[e_X
48     -e_Y]
49 printf('\n In terms of intial coordinates , the shear
        center C is located at \n XC = %.2 f mm',XC(1))
50 printf('\n YC = %.2 f mm',XC(2))
51 xC=Q*XC
52 printf('\n The x and y coordinates of shear center C
        are \n xC = %.2 f mm',xC(1))
53 printf('\n yC = %.2 f mm',xC(2))

```

---

### Scilab code Exa 8.3 Example3

```

1 clc
2 // initialization of variables
3 clear
4 t1=1 //mm

```

```

5 t2=2 //mm
6 oT=9.67 //mm distance between base and the centroid
   of each T-ection
7 y2_bar=100+10+1+oT //mm (follwos from the figure)
8 A1=400 //mm^2
9 y1_bar=100 //mm
10 A2=324 //mm^2
11 Ix=2*A1*y1_bar^2+2*A2*y2_bar^2
12 q1k=A2*y2_bar //q1=q1k*Vy/Ix
13 F1k=(oT+t1/2)*q1k // Fi=Fik*Vy/Ix
14 F2k=60*q1k
15 F3k=(10+t1/2)*q1k
16 q2k=q1k+(A1*y1_bar)
17 F4k=(10+t2/2)*q2k
18 F5k=200*q2k
19 V_pk=2*(F1k+2*F3k+F5k)/Ix // V_p=V_pk*Vy
20 e=(-2*F1k*71-2*F3k*11+F2k*221+F4k*200)/Ix
21 printf('e = %.2f mm',e)

```

---

#### Scilab code Exa 8.4 Example4

```

1 clc
2 // initialization of variables
3 clear
4 b=300 //mm
5 h=500 //mm
6 t1=20 //mm
7 t2=10 //mm
8 t3=t2
9 Ix=687.5e+06 //mm^4
10 q_P=b*t2*h/2
11 q_Q=q_P+h/2*t1*h/4
12 q_S=h/2*t3*h/4
13 q_A=-1/(h/t1+b/t2+h/t3+b/t2)*((-q_P-2/3*(q_Q-q_P))*h
   /t1-q_P/2*b/t2+2/3*q_S*h/t3-q_P/2*b/t2)

```

```

14 e=1/Ix*((444.4+2/3*625)*b*h+444.4/2*177.76*h-q_A
    /(1000*2)*122.24*h)
15 V1=(q_P-q_A+2/3*(q_Q-q_P))*h
16 V2=(q_A+2/3*q_S)*h
17 V=V1+V2
18 printf('e = %d mm',e*10^3)
19 printf('\n V = %d kN',V/1000)

```

---

### Scilab code Exa 8.5 Example5

```

1  clc
2  // initialization of variables
3  clear
4  a=500 //mm
5  b=a
6  t1=5 //mm
7  t2=10 //mm
8  t3=20 //mm
9  // calculations
10 Ix=2343.75e+06 //mm^4
11 q_B=b*t2*a/2
12 q_C=q_B+a/2*t1*a/4
13 q_S=a/2*t3*a/4
14 q_G=2*b*t2*a/2
15 q_H=q_G+a/2*t3*a/4
16 // th_L = th_R = 0
17 // Writing the above in following form
18 //Ab=c ; b={q_A q_F}
19 A11=a/t1+b/t2+a/t3+b/t2
20 A12=a/t3
21 c1=(q_B+2/3*(q_C-q_B))*a/t1 + 1/2*q_B*b/t2 - 2/3*q_S
    *a/t3 + 1/2*q_B*b/t2
22 A21=A12
23 A22=a/t3+2*b/t2+a/t3+2*b/t2
24 c2=(q_G+2/3*(q_H-q_G))*a/t3+1/2*q_G*2*b/t2-2/3*q_S*a

```

```

        /t3+1/2*q_G*2*b/t2
25 A=[A11 A12
26     A21 A22]
27 c=[c1
28     c2]
29 b=inv(A)*c
30 q_A=b(1)/1000 //kN/mm
31 q_F=b(2)/1000 //kN/mm
32 q_B=q_B/1000
33 q_C=q_C/1000
34 q_S=q_S/1000
35 q_G=q_G/1000
36 q_H=q_H/1000
37 b=a // rewriting to it's initial value
38 // To find out e, balance the moments
39 e=-((q_B-q_A+2/3*(q_C-q_A-(q_B-q_A)))*a*b + 1/2*(
        q_B-q_A)*219.1*a - 1/2*q_A*280.9*a + 1/2*q_F
        *471.9*a -1/2*(q_G-q_F)*528.1*a-(q_G-q_F+2/3*(
        q_H-q_F-(q_G-q_F)))*a*2*b)
40 e=e/Ix
41 printf('e = %.1f mm',e*10^3)
42 // Wrong answer in the text

```

---

# Chapter 9

## CURVED BEAMS

Scilab code Exa 9.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  a=30 //mm
5  c=80 //mm
6  b=50 //mm
7  P=9.5 //kN
8  d=100 //mm position of P
9  // calculations
10 P=P*10^3
11 A=b^2
12 A=b*(c-a)
13 Am=b*log(c/a)
14 R=(a+c)/2
15 p=d+R
16 Mx=p*P
17 r=a
18 S_thB=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
19 r=c
20 S_thC=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
21 printf('The maximum tensile stress is (at point B) =
```

```

        %.1f MPa',S_thB)
22 printf('\n The maximum cpmressive stress is (at
        point C) = %.1f MPa',S_thC)

```

---

### Scilab code Exa 9.2 Example2

```

1  clc
2  // initialization of variables
3  clear
4  // part (c)
5  r_A=1.47 //m
6  theta=%pi
7  // S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
        theta))*10^5 //kPa
8  r=r_A
9  S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
        theta))*10^5 //kPa
10 S_A=S_th
11
12 r_B=1.53 //m
13 r=r_B
14 S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
        theta))*10^5 //kPa
15 S_B=S_th
16 printf('part (c)')
17 printf('\n The tensile stress at A is %.2f MPa',S_A
        /1000)
18 printf('\n The compressive stress at B is %.2f MPa',
        S_B/1000)
19
20 // part (d)
21 theta=%pi/2
22 r=r_A
23 S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
        theta))*10^5 //kPa

```



```

24 S_A=S_th
25 r=r_B
26 S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
    theta))*10^5 //kPa
27 S_B=S_th
28 printf('\n part (d)')
29 printf('\n The tensile stress at A is %.2f MPa',S_A
    /1000)
30 printf('\n The compressive stress at B is %.2f MPa',
    S_B/1000)
31
32 //part (e)

```

---

### Scilab code Exa 9.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  Y=500 //MPa
5  SF=2.00
6  A1=1658.76 //mm^2
7  R1=73.81 //mm
8  Am1=22.64 //mm
9  A2=6100 //mm^2
10 R2=126.62 //mm
11 Am2=50.57 //mm
12 A3=115.27 //mm^2
13 R3=186.01 //mm
14 Am3=0.62 //mm
15 A=A1+A2+A3
16 Am=Am1+Am2+Am3
17 R=(A1*R1+A2*R2+A3*R3)/A
18 rB=60 //mm
19 rC=rB+24+100+5 //follows from figure
20 //P unknown, so put unity to solve for it later

```

```

21 P=1
22 Mx=116.37*P
23 r=rB
24 S_thB=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
25 r=rC
26 S_thC=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
27 S_th=max(abs(S_thB),abs(S_thC))
28 Pf=Y/S_th
29 P=Pf/SF
30 printf('P = %.d N',P)

```

---

#### Scilab code Exa 9.4 Example4

```

1  clc
2  // initialization of variables
3  clear
4  // part (b)
5  // Following is the formula used in evaluating the
   circumferential stress
6  // Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
7  // Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
8  // S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+
   Nr/Dr)
9  ri=60 //mm
10 ro=180 //mm
11 t=50 //mm
12 th=%pi/2
13 // For, maximum tensile stress r=ri
14 r=ri
15 Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
16 Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
17 // Question was asked in terms of P, so let it be
   unity
18 P=1
19 S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/

```

```

    Dr)
20 S_max1=S_th
21 // For maximum compressive stress r=ro
22 r=ro
23 Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
24 Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
25 S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/
    Dr)
26 S_max2=S_th
27 printf('part (b)')
28 printf('\n for theta=90 degrees')
29 printf('\n Maximum tensile stress = %.6f P',S_max1)
30 printf('\n Maximum compressive stress = %.6f P',
    S_max2)
31
32
33 th=%pi
34 // For, maximum tensile stress r=ri
35 r=ri
36 Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
37 Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
38 // Question was asked in terms of P, so let it be
    unity
39 P=1
40 S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/
    Dr)
41 S_max1=S_th
42 // For maximum compressive stress r=ro
43 r=ro
44 Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
45 Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
46 S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/
    Dr)
47 S_max2=S_th
48 printf('\n for theta=180 degrees')
49 printf('\n Maximum tensile stress = %.6f P',S_max1)
50 printf('\n Maximum compressive stress = %.6f P',
    S_max2)

```

```

51
52 //part(c)
53 S_thMax=340 //MPa
54 SF=2.2
55 P=S_thMax/(SF*S_max1)
56 printf('\n part(c)')
57 printf('\n The maximum allowable load is %.2f kN',P
    /1000)

```

---

### Scilab code Exa 9.5 Example5

```

1  clc
2  // initialization of variables
3  clear
4  P=120 //kN
5  b1=120 //mm
6  b2=120 //mm
7  h1=48 //mm
8  h2=24 //mm
9  P=P*10^3
10 A=h1*b1+b2*h2
11 R=(b1*h1*96+b2*h2*180)/A
12 Am=b1*log(b1/72)+h2*log(240/b2)
13 r=72
14 Mx=364*P
15 S_thB=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
16
17 r1=120 //mm
18 t=24 //mm
19 A1=h1*r1
20 Am1=r1*log(r1/r)
21 S_rr=(A*Am1-A1*Am)*Mx/(t*r1*A*(R*Am-A))
22 printf('Circumferential stress is %.1f MPa',S_thB)
23 printf('\n Radial stress is %.1f MPa',S_rr)

```

---

### Scilab code Exa 9.6 Example6

```
1  clc
2  // initialization of variables
3  clear
4  Mo=96 //kN
5  P=120 //kN
6  b1=150 //mm
7  h1=60 //mm
8  b2=120 //mm
9  h2=50 //mm
10 b3=b1
11 h3=40 //mm
12 ro=80 //mm
13 r1=140 //mm
14 r2=260 //mm
15 r3=300 //mm
16 // calculations
17 Mo=Mo*10^6 // N.mm
18 P=P*10^3 // N
19 A=b1*h1+b2*h2+b3*h3
20 Am=b1*log(r1/ro)+h2*log(r2/r1)+b3*log(r3/r2)
21 R=(b1*h1*110+b2*h2*200+b3*h3*280)/A
22 Mx=Mo+P*R
23 r=80 //mm
24 S_th=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
25
26 A1=9000 //mm^2
27 r1=140 //mm
28 t=50 //mm
29 Am1=b1*log(r1/ro)
30 N=120000
31 S_rr=A1*N/(A*t*r1) + (A*Am1-A1*Am)*Mx/(t*r1*A*(R*Am-A))
```

```

32 printf('Circumferential stress is %.2f MPa',S_th)
33 printf('\n Radial stress at B1 is %.2f MPa',S_rr)
34 // to find radial stress at C;
35 A1=b1*h1+b2*h2
36 Am1=b1*log(r1/ro)+h2*log(r2/r1)
37 r1=260 //mm
38 t=50 //mm
39 S_rr=A1*N/(A*t*r1) + (A*Am1-A1*Am)*Mx/(t*r1*A*(R*Am-
    A))
40 printf('\n Radial stress at C1 is %.2f MPa',S_rr)

```

---

#### Scilab code Exa 9.7 Example7

```

1  clc
2  // initialization of variables
3  clear
4  l=15 //m
5  R=10 //m
6  d=0.8 //m
7  b=0.13 //m
8  Po=2400 //N/m
9  P=4800 //N/m
10 // calculations
11 a=R-d/2
12 c=R+d/2
13 A=b*d
14 Am=b*log(c/a)
15 Mx=(Po+P)*l^2/8
16 S_thMax=Mx*(A-a*Am)/(A*a*(R*Am-A))
17 // To find out max radial stress
18 // Nr=d*log(r/a)-(r-a)*log(c/a)
19 // Dr=r*d*(R*log(c/a)-d)
20 // S_rr=Mx/b*Nr/Dr
21 r=a*exp(1-(a/d*log(c/a)))
22 Nr=d*log(r/a)-(r-a)*log(c/a)

```

```

23 Dr=r*d*(R*log(c/a)-d)
24 S_rrMax=Mx/b*Nr/Dr
25 printf('\n part (a)')
26 printf('\n Maximum circumferential stress is %.1f
    MPa',S_thMax/10^6)
27 printf('\n Maximum radial stress is %.3f MPa',
    S_rrMax/10^6)
28 // part (b)
29 Ix=b*d^3/12
30 S_th=Mx*d/(2*Ix)
31 printf('\n part (b)')
32 printf('\n Maximum circumferential stress using
    straight beam formula is %.1f MPa',S_th/10^6)

```

---

#### Scilab code Exa 9.8 Example8

```

1  clc
2  // initialization of variables
3  clear
4  // part(a)
5  Y=280 //MPa
6  A=4000 //mm^2
7  Am=44.99 //mm
8  R=100.0 //mm
9  r=180 //mm
10 r=60 //mm
11 // Mx is not yet known take it as unity
12 Mx=1 //unity
13 r=180
14 S_thMax=Mx*(A-r*Am)/(A*r*(R*Am-A))
15 Mx=Y/(abs(S_thMax))
16 printf('part(a)')
17 printf('\n Mx = %.2f kN.m',Mx/10^6)
18 // part(b)
19 k1=1.143

```

```

20 t_w=20
21 b_p=40
22 alpha=0.651
23 Beta=1.711
24 r=60 //mm
25 b1=2*alpha*b_p+t_w
26 A=b1*t_w+t_w*R
27 R=(b1*t_w*70+t_w*R*130)/A
28 Am=b1*log(80/r)+t_w*log(180/80)
29 // Mx not yet known teke it as unity
30 Mx=1
31 S_thMax=Mx*(A-r*Am)/(A*r*(R*Am-A))
32 r=70 //mm
33 S_thbar=Mx*(A-r*Am)/(A*r*(R*Am-A))
34 S_xx=-Beta*S_thbar
35 //tau_max=Y/2=(S_thMax-S_xx)/2
36 Mx=Y/(S_thMax-S_xx)
37 printf('\n part (b)')
38 printf('\n Mx = %.2 f kN.m',Mx/10^6)

```

---

### Scilab code Exa 9.9 Example9

```

1 clc
2 // initialization of variables
3 clear
4 E=72 //GPa
5 t=60 //mm
6 M=24 //kN.m
7 // part (a)
8 ro=100 //mm
9 r1=150 //mm
10 A=t*r1
11 Am=t*log((ro+r1)/ro)
12 R=ro+r1/2
13 E=E*10^3

```



```

14 Mx=M*10^6
15 Phi=Am*Mx*%pi/(A*(R*Am-A)*E)
16 printf(' part(a) ')
17 printf('\n Phi = %.5f rad',Phi)
18 // part(b)
19 //Mx=Mx+P*R*sin(th)
20 delta_P=Am*Mx*R*2/(A*(R*Am-A)*E)
21 printf('\n part(b) ')
22 printf('\n deflection = %.3f mm',delta_P)

```

---

### Scilab code Exa 9.10 Example10

```

1 clc
2 // initialization of variables
3 clear
4 P=11.2 //kN
5 E=200 //GPa
6 v=0.3
7 Ix=181.7e+03 //mm^4
8 k1=0.643
9 b1=34.7 //mm
10 h1=10 //mm
11 b2=40 //mm
12 h2=10 //mm
13 t=10 //mm
14 h=50 //mm
15 E=E*10^3
16 A=b1*h1+b2*h2
17 R=(b1*h1*35+b2*h2*60)/A
18 Am=b1*log(40/30)+h1*log(80/40)
19 G=E/(2*(1+v))
20 Aw=t*h
21 P=P*10^3
22 delta_P=2*P*100/(Aw*G) + 2*P/(E*3*Ix)*100^3 + P
    *48.4*%pi/(2*Aw*G) + P*48.4*%pi/(2*A*E) + P

```

```
    *16.9/(A*(48.4*16.9-A)*E)*(100^2*%pi+%pi/2*(48.4)
    ^2+2*100*2*48.4)
23 printf('seperation = %.3f mm',delta_P)
```

---

# Chapter 10

## BEAMS ON ELASTIC FOUNDATIONS

Scilab code Exa 10.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  // part (a)
5  E=200 //GPa
6  d=184 //mm
7  c=99.1 //mm
8  Ix=36.9e+06 //mm4
9  k=14.0 //N/mm2
10 P=170 //kN
11 // calculations
12 E=E*103
13 P=P*103
14 Beta=(k/(4*E*Ix))^(1/4)
15 y_max=P*Beta/(2*k)
16 M_max=P/(4*Beta)
17 S_max=M_max*c/Ix
18 printf(' part (a) ')
19 printf(' \n y_max = %.3 f mm', y_max)
```

```

20 printf( '\n M_max = %.2 f kN.m', M_max/10^6)
21 printf( '\n S_max = %.1 f MPa', S_max)
22 // part (b)
23 z1=1.7 //m
24 z1=z1*10^3 //mm
25 z2=2*z1
26 // A_bz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
27 // C_bz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
28 A_bzo=1
29 C_bzo=1
30 A_bz1=exp(-Beta*z1)*(sin(Beta*z1)+cos(Beta*z1))
31 A_bz2=exp(-Beta*z2)*(sin(Beta*z2)+cos(Beta*z2))
32 C_bz1=exp(-Beta*z1)*(-sin(Beta*z1)+cos(Beta*z1))
33 C_bz2=exp(-Beta*z2)*(-sin(Beta*z2)+cos(Beta*z2))
34 y_end=P*Beta/(2*k)*(A_bzo+A_bz1+A_bz2)
35 M_end=P/(4*Beta)*(C_bzo+C_bz1+C_bz2)
36 y_center=P*Beta/(2*k)*(A_bzo+2*A_bz1)
37 M_center=P/(4*Beta)*(C_bzo+2*C_bz1)
38 y_max=max(y_end,y_center)
39 M_max=max(M_end,M_center)
40 S_max=M_max*c/Ix
41 printf( '\n part(b) ')
42 printf( '\n y_max = %.3 f mm', y_max)
43 printf( '\n M_max = %.2 f kN.m', M_max/10^6)
44 printf( '\n S_max = %.1 f MPa', S_max)

```

---

### Scilab code Exa 10.2 Example2

```

1 clc
2 // initialization of variables
3 clear
4 d=100 //mm
5 Ix=2.45e+06 //mm^4
6 E=72 //GPa
7 L=6.8 //m

```

```

8 K=110 //N/mm
9 l=1.1 //m
10 P=12 //kN
11 // calculations
12 E=E*10^3
13 P=P*10^3
14 l=l*10^3
15 k=K/l
16 L1=7*l
17 Beta=(k/(4*E*Ix))^(1/4)
18 if(l<%pi/(4*Beta))
19 if(L1>3*%pi/(2*Beta))
20     y_max=P*Beta/(2*k)
21     M_max=P/(4*Beta)
22     S_max=M_max*d/(2*Ix)
23 end
24 end
25 printf('y_max = %.3 f mm',y_max)
26 printf('\n M_max = %.2 f kN.m',M_max/10^6)
27 printf('\n S_max = %.1 f MPa',S_max)
28 A_b1=exp(-Beta*l)*(sin(Beta*l)+cos(Beta*l))
29 A_2b1=exp(-Beta*2*l)*(sin(Beta*2*l)+cos(Beta*2*l))
30 A_3b1=exp(-Beta*3*l)*(sin(Beta*3*l)+cos(Beta*3*l))
31     y_C=P*Beta/(2*k)*A_b1
32     y_B=P*Beta/(2*k)*A_2b1
33     y_A=P*Beta/(2*k)*A_3b1
34 printf('\n y_C = %.2 f mm',y_C)
35 printf('\n y_B = %.2 f mm',y_B)
36 printf('\n y_A = %.2 f mm',y_A)

```

---

#### Scilab code Exa 10.4 Example4

```

1 clc
2 // initialization of variables
3 clear

```

```

4 E=10 //GPa
5 h=200 //mm
6 b=100 //mm
7 ko=0.04 //N/mm^3
8 w=35 //N/mm
9 L1=3.61 //m
10 // calculations
11 E=E*10^3
12 L1=L1*10^3
13 k=b*ko
14 Ix=b*h^3/12
15 Beta=(k/(4*E*Ix))^(1/4)
16 ba=2.00 // ba = Beta*a based on the discussion
17 //D_bz=exp(-Beta*z)*sin(Beta*z)
18 D_ba=exp(-ba)*cos(ba)
19 y_max=w/k*(1-D_ba)
20 ba=0.777 //Beta*a
21 bb=4.777 //Beta*b
22 B_ba=exp(-ba)*sin(ba)
23 B_bb=exp(-bb)*sin(bb)
24 M_max=abs(-w*(B_ba-B_bb)/(4*Beta^2))
25 c=h/2
26 S_max=M_max*c/Ix
27 // calculation of MH
28 ba=%pi/4 //Beta*a
29 bb=4-%pi/4 //Beta*b
30 B_ba=exp(-ba)*sin(ba)
31 B_bb=exp(-bb)*sin(bb)
32 M_H=w/(4*Beta^2)*(B_ba+B_bb)
33 printf('y_max = %.3f mm',y_max)
34 printf('\n M_max = %.3f kN.m',M_max/10^6)
35 printf('\n S_max = %.3f MPa',S_max)
36 printf('\n MH = %.3f kN.m',M_H/10^6)

```

---

Scilab code Exa 10.5 Example5

```

1  clc
2  // initialization of variables
3  clear
4  E=200 //GPa
5  h=102 //mm
6  b=68 //mm
7  Ix=2.53e+06 //mm^4
8  L1=4 //m
9  ko=0.35 //N/mm^3
10 P=30.0 //kN
11 // calculations
12 E=E*10^3
13 P=P*10^3
14 L1=L1*10^3
15 k=b*ko
16 Beta=(k/(4*E*Ix))^(1/4)
17 if(L1>3*%pi/(2*Beta))
18     y_max=2*P*Beta/k
19     M_max=-0.3224*P/Beta
20     S_max=abs(M_max*h/(2*Ix))
21 end
22 z=%pi/(4*Beta)
23 printf('y_max = %.2f mm',y_max)
24 printf(' \n M_max = %.2f kN.m',M_max/10^6)
25 printf(' \n S_max = %.1f MPa',S_max)
26 printf(' \n Location of Sigma_max is z = %d mm',z)

```

---

### Scilab code Exa 10.6 Example6

```

1  clc
2  // initialization of variables
3  clear
4  P=30.0 //kN
5  a=500 //mm
6  h=102 //mm

```

```

7 b=68 //mm
8 k=23.8 //N/mm^2
9 Beta=0.001852
10 Ix=2.53e+06 //mm^4
11 // calculations
12 P=P*10^3
13 C_ba=exp(-Beta*a)*(-sin(Beta*a)+cos(Beta*a))
14 D_ba=exp(-Beta*a)*cos(Beta*a)
15 // y = P*Beta/(2*k)*(A_bz+2*D_ba*D_baz+C_ba*C_baz)
16 // Mx = P/(4*Beta)*(C_bz-2*D_ba*B_baz-C_ba*A_baz)
17 A_ba=exp(-Beta*a)*(sin(Beta*a)+cos(Beta*a))
18 B_ba=exp(-Beta*a)*sin(Beta*a)
19 C_ba=exp(-Beta*a)*(-sin(Beta*a)+cos(Beta*a))
20 D_ba=exp(-Beta*a)*cos(Beta*a)
21 z1=424 //mm
22 z=z1-a
23 A_bz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
24 B_bz=exp(-Beta*z)*sin(Beta*z)
25 C_bz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
26 D_bz=exp(-Beta*z)*cos(Beta*z)
27 // to find out X_baz
28 z=a+z
29 A_baz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
30 B_baz=exp(-Beta*z)*sin(Beta*z)
31 C_baz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
32 D_baz=exp(-Beta*z)*cos(Beta*z)
33 y_max = P*Beta/(2*k)*(A_bz+2*D_ba*D_baz+C_ba*C_baz)
34 printf('y_max = %.4f mm',y_max)
35 // For M_max
36 z1=500 //mm
37 z=z1-a
38 A_bz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
39 B_bz=exp(-Beta*z)*sin(Beta*z)
40 C_bz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
41 D_bz=exp(-Beta*z)*cos(Beta*z)
42 // to find out X_baz
43 z=a+z
44 A_baz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))

```



```

45 B_baz=exp(-Beta*z)*sin(Beta*z)
46 C_baz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
47 D_baz=exp(-Beta*z)*cos(Beta*z)
48 M_max = P/(4*Beta)*(C_bz-2*D_ba*B_baz-C_ba*A_baz)
49 printf('\n M_max = %d N.mm',M_max)
50 S_max=M_max*h/(2*Ix)
51 printf('\n Sigma_max = %.1 f MPa',S_max)

```

---

### Scilab code Exa 10.7 Example7

```

1  clc
2  // initialization of variables
3  clear
4  D=30 //m
5  t=10 //m
6  h=20 //mm
7  E=200 //GPa
8  v=0.29
9  rho=900 //kg/m^3
10 // calculations
11 // part (a)
12 E=E*10^3
13 a=D/2*10^3
14 p=t*10^3*9.807*rho*10^-9
15 S_th=p*a/h
16 tau_max=S_th/2
17 printf('part (a)')
18 printf('\n Maximum shear stress= %.2 f MPa',tau_max)
19 // part (b)
20 k=E*h/(a^2)
21 Beta=(3*(1-v^2)/(h^2*a^2))^(1/4)
22 L1=3*pi/(4*Beta) //L1=L/2
23 u=S_th*a/E
24 w=2*k*u/(Beta)
25 M_max=w/(4*Beta)

```

```

26 Szz_max=M_max*(h/2)/(h^3/12)
27 Sth_max=v*Szz_max
28 tau_max=Szz_max/2
29 u_b=w*(1-v)*a/(2*E*h)
30 printf('\n part (b)')
31 printf('\n Maximum shear stress= %.2f MPa',tau_max)
32 printf('\n u_bottom = %.3f mm',u_b)
33 // part (c)
34 w=u*k/(2*Beta)
35 z=%pi/(4*Beta)
36 B_bz=exp(-Beta*z)*sin(Beta*z)
37 M_max=-w*B_bz/Beta
38 c=6
39 I=h^2
40 Szz_max=(M_max*c/I)
41 S_th1=v*(Szz_max)
42 k=0.3224
43 S_th2=(1-k)*S_th
44 Sigma_th=S_th1+S_th2
45 tau_max=(Sigma_th-Szz_max)/2
46 printf('\n part (c)')
47 printf('\n Maximum shear stress= %.2f MPa',tau_max)

```

---

# Chapter 11

## THE THICK WALL CYLLINDER

Scilab code Exa 11.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  E=200 //GPa
5  v=0.29
6  Di=20 //mm
7  Do=100 //mm
8  a=10 //mm
9  b=50 //mm
10 p1=300 //MPa
11 // calculations
12 // S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
13 // S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
14 r=10
15 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
16 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
17 printf('r = %d mm',r)
18 printf('\n Radial stress = %.1f MPa',S_rr)
19 printf('\n circumferential stress = %.1f MPa',S_th)
```

```

20 r=25
21 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
22 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
23 printf('\n r = %d mm',r)
24 printf('\n Radial stress = %.1f MPa',S_rr)
25 printf('\n circumferential stress = %.1f MPa',S_th)
26 r=50
27 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
28 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
29 printf('\n r = %d mm',r)
30 printf('\n Radial stress = %.1f MPa',S_rr)
31 printf('\n circumferential stress = %.1f MPa',S_th)

```

---

### Scilab code Exa 11.2 Example2

```

1 clc
2 // initialization of variables
3 clear
4 E=72 //GPa
5 v=0.33
6 Di=200 //mm
7 Do=800 //mm
8 a=100 //mm
9 r=a
10 b=Do/2 //mm
11 p1=150 //MPa
12 E=E*10^3
13 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
14 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
15 S_zz=p1*a^2/(b^2-a^2)
16 tau_max=(S_th-S_rr)/2
17 u_a=p1*a/(E*(b^2-a^2))*((1-2*v)*a^2+(1+v)*b^2)
18 printf('Radial stress = %.1f MPa',S_rr)
19 printf('\n circumferential stress = %.1f MPa',S_th)
20 printf('\n Normal stress = %d MPa',S_zz)

```

```

21 printf('\n Maximum shear stress = %d MPa',tau_max)
22 printf('\n u|r=a = %.4 f mm',u_a)

```

---

### Scilab code Exa 11.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  E=200 //GPa
5  a=10 //mm
6  v=0.29
7  ci=25.072 //mm
8  co=25 //mm
9  b=50 //mm
10 rr=0.072 //mm
11 re=0.025 //mm
12 alpha=0.0000117 // per celcius
13 // calculations
14 E=E*10^3
15 p1=300 //MPa
16 term1=co/(E*(b^2-co^2))*((1-v)*co^2+(1+v)*b^2)
17 term2=-ci/(E*(ci^2-a^2))*((-1-v)*ci^2)-(1+v)*a^2)
18 ps=rr/(term1+term2)
19
20 // Inner cylinder p1=0 p2=ps a=10 b=25
21 // outer cylinder p1=ps p2=0 a=25 b=50
22 // S_rr=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
23 // S_th=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
24 // results
25 // residual stresses for inner cylinder
26 p1=0
27 p2=ps
28 r=10

```

```

29 a=10
30 b=25
31 S_rri1=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
32 S_thi1=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
33 printf('\n Inner cylinder ')
34 printf('\n r = %d mm',r)
35 printf('\n S_rr |R = %d MPa,    S_th |R = %.1 f MPa',
    S_rri1,S_thi1)
36 r=25
37 S_rri2=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
38 S_thi2=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
39 printf('\n r = %d mm',r)
40 printf('\n S_rr |R = %.1 f MPa,    S_th |R = %.1 f MPa',
    S_rri2,S_thi2)
41 // residual stresses for outer cylinder
42 p1=ps
43 p2=0
44 a=25
45 b=50
46 r=25
47 S_rro1=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
48 S_tho1=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
49 printf('\n ')
50 printf('\n Outer cylinder ')
51 printf('\n r = %d mm',r)
52 printf('\n S_rr |R = %d MPa,    S_th |R = %.1 f MPa',
    S_rro1,S_tho1)
53 r=50
54 S_rro2=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
55 S_tho2=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)

```

```

56 printf('\n r = %d mm',r)
57 printf('\n S_rr|R = %.1f MPa,    S_th|R = %.1f MPa',
    S_rr02,S_tho2)
58 // AN internal pressure of 300 MPa
59 a=10 //mm
60 b=50 //mm
61 p1=300 //MPa
62 r=10
63 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
64 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
65 S_rr1=S_rr+S_rri1
66 S_th1=S_th+S_thi1
67 printf('\n')
68 printf('\n Inner cylinder')
69 printf('\n r = %d mm',r)
70 printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr1
    ,S_th1)
71 r=25
72 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
73 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
74 S_rr2=S_rr+S_rri2
75 S_th2=S_th+S_thi2
76 printf('\n r = %d mm',r)
77 printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr2
    ,S_th2)
78 // Outer Cyllinder
79 S_rr1=S_rr+S_rr01
80 S_th1=S_th+S_tho1
81 printf('\n')
82 printf('\n Outer cylinder')
83 printf('\n r = %d mm',r)
84 printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr1
    ,S_th1)
85 r=50
86 S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
87 S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
88 S_rr2=S_rr+S_rr02
89 S_th2=S_th+S_tho2

```

```

90 printf('\n r = %d mm',r)
91 printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr2
    ,S_th2)
92 //delT=u/(r*alpha)
93 u=rr+re
94 r=25
95 delT=u/(r*alpha)
96 printf('\n delT = %.1f degree Celcius',delT)

```

---

#### Scilab code Exa 11.4 Example4

```

1 clc
2 // initialization of variables
3 clear
4 SF=1.75
5 p1=300 //MPa
6 S_rr=-SF*p1
7 S_th=SF*325
8 Y=1/sqrt(2)*sqrt((S_th-S_rr)^2+S_rr^2+S_th^2)
9 printf(' Y = %.1f MPa',Y)

```

---

#### Scilab code Exa 11.5 Example5

```

1 clc
2 // initialization of variables
3 clear
4 p1=300 //MPa
5 SF=1.75
6 S_rr=SF*p1
7 S_th=SF*325-550.2 //Values are obtained from running
    Ex11_3.sce
8 Y1=1/sqrt(2)*sqrt((S_th-S_rr)^2+S_rr^2+S_th^2)
9 S_rr1=37.5

```



```

10 S_rre=-189.1
11 S_th1=62.5
12 S_the=315.1
13 // ABove values are obtained from running the codes
    Ex11_1 and Ex11_3.sce
14 S_rr=-SF*S_rr1+S_rre
15 S_th=SF*S_th1+S_the
16 Y2=1/sqrt(2)*sqrt((S_th-S_rr)^2+S_rr^2+S_th^2)
17 if(Y2>Y1)
18     Y=Y2
19 end
20 printf(' Y = %.1f MPa',Y)

```

---

#### Scilab code Exa 11.6 Example6

```

1 clc
2 // initialization of variables
3 clear
4 SF=1.8
5 a=20 //mm
6 b=40 //mm
7 Y=450 //MPa
8 // part (a)
9 tau_Y=Y/sqrt(3)
10 Pp=2*tau_Y*log(b/a)
11 S_th=2*tau_Y*(1-log(b/a))
12 S_rr=-Pp
13 S_zz=(S_th+S_rr)/2
14 printf(' part (a) ')
15 printf('\n S_th = %.1f MPa',S_th)
16 printf('\n S_zz = %.1f MPa',S_zz)
17 // part (b)
18 S_thR=S_th-Pp*(b^2+a^2)/(b^2-a^2)
19 S_zzR=S_zz-Pp*(a^2)/(b^2-a^2)
20 S_thR=S_thR/2

```

```

21 S_zzR=S_zzR/2
22 printf('\n part (b)')
23 printf('\n S_th|R = %.1f MPa',S_thR)
24 printf('\n S_zz|R = %.1f MPa',S_zzR)
25 // par (c)
26 // We need to find out p1. To do that let it be
    unity
27 p1=1
28 S_thR=-S_thR
29 S_zzR=-S_zzR
30 S_rr=-SF*p1
31 S_th=SF*p1*(b^2+a^2)/(b^2-a^2)
32 S_zz=SF*p1*a^2/(b^2-a^2)
33 // 2Y^2=(s_th-S_rr)^2+(S_rr-S_zz)^2+(S_zz-S_th)^2
34 // S_th=S_th*p1-S_thR
35 // S_zz=S_zz*p1-S_zzR
36 // a*p1^2+b*p+c=0
37 a=(S_th+SF)^2+(-SF-S_zz)^2+(S_zz-S_th)^2
38 c=S_thR^2+S_zzR^2+(S_thR-S_zzR)^2
39 b=-2*(S_th+SF)*S_thR+2*S_zzR*(-SF-S_zz)+2*(S_zz-S_th
    )*(S_thR-S_zzR)
40 c=c-2*Y^2
41 p11=roots([a b c])
42 p12=roots([a 0 -2*Y^2])
43 p11=p11(1)
44 p12=p12(1)
45 printf('\n Internal working pressure is %.1f MPa,',
    p11)
46 printf('\n Without residual stresses %.1f MPa',p12)

```

---

### Scilab code Exa 11.8 Example8

```

1 clc
2 // initialization of variables
3 clear

```

```

4 a=100 //mm
5 b=300 //mm
6 Y=620 //MPa
7 E=200 //GPa
8 S_zz=0
9 v=0.29
10 rho=7.85e+03 //kg/m^3
11 // part (a)
12 S_thmax=Y
13 Wy=sqrt(4*Y/(rho*((3+v)*b^2+(1-v)*a^2)))
14 printf('part (a)')
15 printf('\n Omega_y =%d rad/s ',Wy*10^6)
16 // part (b)
17 Wp=sqrt(3*Y/(rho*(b^2+a*b+a^2)))
18 ratio=Wp/Wy
19 printf('\n Omega_p = %d rad/s ',Wp*10^6)
20 printf('\n ratio = %.2f',ratio)

```

---

### Scilab code Exa 11.9 Example9

```

1 clc
2 // initialization of variables
3 clear
4 a=100 //mm
5 b=300 //mm
6 v=0.29
7 a=a*10^-3
8 b=b*10^-3
9 printf('r      S_rr |R/Y      S_th |R/Y      (S_th/R-S_rr
      /R)/Y')
10 for i=1:21
11     r=0.09+0.01*i
12 S_rrR=((r-a)/r - 3/(b^2+a*b+a^2)*((r^3-a^3)/(3*r) +
      (3+v)/8*(a^2+b^2-r^2-a^2*b^2/r^2)))
13 S_thR=(1- 3/(8*(b^2+a*b+a^2)) * ((3+v)*(a^2+b^2+a^2*

```

```
    b^2/r^2) - (1+3*v)*r^2))
14 printf('\n %.2f      %.5f      %.5f      %.5f',r,S_rrR,
    S_thR,S_rrR-S_thR)
15 end
```

---

## Chapter 12

# ELASTIC AND INELSTIC STABILITY OF COLUMNS

Scilab code Exa 12.3 Example3

```
1 clc
2 // initialization of variables
3 clear
4 b=25 //mm
5 L=250 //mm
6 E_T=31 //GPa
7 Sigma_T=262 //MPa // From the curve
8 r=b/sqrt(12)
9 Q=%pi^2*E_T/((L/r)^2)
10 // Since this is not close enough, increment E_T
11 E_T=31.6 //GPa
12 Q=%pi^2*E_T/((L/r)^2)
13 P_T=Q*b^2
14 printf('Buckling load is %d kN',P_T)
```

---

Scilab code Exa 12.5 Example5

```
1 clc
2 // initialization of variables
3 clear
4 L=1 //m
5 b=40 //mm
6 h=75 //mm
7 SF=2.5
8 K=1
9 L=L*10^3
10 Iy=b*h^3/12
11 A=b*h
12 ry=sqrt(Iy/A)
13 K_y=K*L/ry
14 rz=b/sqrt(12)
15 K=0.5
16 K_z=K*L/rz
17 S_cr=229 //MPa
18 P_cr=S_cr*A
19 P=P_cr/SF
20 printf('P = %d kN',P/10^3)
```

---

# Chapter 13

## FLAT PLATES

Scilab code Exa 13.2 Example2

```
1  clc
2  // initialization of variables
3  clear
4  d=3.6 //m
5  w=2.7 //m
6  ha=3.0 //m
7  b=0.9 //m
8  a=1.2 //m
9  v=0.29
10 E=200 //GPa
11 p=ha*9.8
12 //part (a)
13 S_w=124 //MPa
14 b_a=b/a
15 M=0.04*p*b^2*10^3
16 h=sqrt(6*M/S_w)
17 printf('part (a)')
18 printf('\n h = %.2 f mm',h)
19 // part (b)
20 C=0.032/(1+b_a^4)
21 p=p*10^3
```

```

22 E=E*10^9
23 b=b*10^3
24 w_max=C*(1-v^2)*p*b^4/(E*h^3)
25 printf('\n part (b)')
26 printf('\n w_max = %.2 f mm',w_max)

```

---

### Scilab code Exa 13.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  E=200 //GPa
5  v=0.29
6  Y=315 //MPa
7  h=10 //mm
8  D=200 //mm
9  SF=2.0
10 // part (a)
11 a=D/2
12 E=E*10^3
13 Py=1 // Since unknown
14 S_maxk=3/4*Py*a^2/h^2
15 Py=Y/S_maxk
16 w_max=3/16*(1-v^2)*Py*a^4/(E*h^3)
17 printf('Py = %.2 f MPa',Py)
18 printf('\n W_max = %.3 f mm',w_max)
19 // part (b)
20 Pw=Py/SF
21 printf('\n part (b)')
22 printf('\n Pw = %.2 f MPa',Pw)

```

---

### Scilab code Exa 13.4 Example4



```
1 clc
2 // initialization of variables
3 clear
4 D=500 //mm
5 h=5 //mm
6 Sigma=288 //MPa
7 E=72 //GPa
8 SF=2
9 //part (a)
10 a=D/2
11 E=E*10^3
12 f=Sigma*a^2/(E*h^2)
13 // w_max/h has to be 2.4 since f=10
14 Pr=50
15 p=Pr*E*h^4/a^4
16 p=p/2
17 printf('part (a)')
18 printf('\n Allowable pressure = %d kPa',p*10^3)
19 // part (b)
20 q=p*a^4/(E*h^4)
21 // Corresponding w_max/h = 1.8
22 w_max=1.8*h
23 printf('\n part (b)')
24 printf('\n W_max = %.2 f mm',w_max)
```

---

# Chapter 14

## STRESS CONCENTRATIONS

Scilab code Exa 14.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  // part (a)
5  ab_r=100
6  Sigma_1=-20 //MPa
7  Sigma_2=-75 //MPa
8  alphao=0.01 //rad
9  //calculations
10 A=(Sigma_1+Sigma_2)/(Sigma_1-Sigma_2)
11 th=1/2*acos((A*sinh(2*alphao)-1/2*(sinh(2*alphao)+
    cosh(2*alphao)))/A)
12 printf('pat (a)')
13 printf('\n theta = %.4f rad',th)
14 //part (b)
15 S_bb=-((Sigma_1-Sigma_2)^2/(2*(Sigma_1+Sigma_2))*(1+
    cosh(2*alphao)/sinh(2*alphao)))
16 printf('\n part (b)')
17 printf('\n Maximum tensile stress = %d MPa',S_bb)
18 //part (c)
19 Beta=exp(2*alphao)*cosh(2*alphao)-2*A^2*(sinh(2*
```

```

    alphao))^2
20 Beta=1/2*acos(Beta/(exp(2*alphao)))
21 printf('\n part (c)')
22 printf('\n Beta = %.4f rad',Beta)

```

---

#### Scilab code Exa 14.2 Example2

```

1  clc
2  // initialization of variables
3  clear
4  S_u=420 //MPa
5  SF=4.00
6  D=110 //mm
7  d=50.0 //mm
8  w=20 //mm
9  rho=10.0 //mm
10 SF=4.0
11 // calculations
12 t=(D-d)/2
13 tr=t/rho
14 rd=rho/d
15 S_cs=1+2*sqrt(tr)
16 A=w*d
17 Pf=S_u*A/1.83
18 P=Pf/SF
19 printf('P = %.1f kN',P/10^3)

```

---

#### Scilab code Exa 14.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  //part(a)

```

```
5 H=200 //mm
6 h=100 //mm
7 rho=10 //mm
8 Sigma_u=250 //MPa
9 P=1.5 //kN
10 L=1.4 //m
11 b=40 //mm
12 P=P*10^3
13 L=L*10^3
14 Hr=H/h
15 rh=rho/h
16 S_cc=1.77
17 c=h/2
18 I=b*h^3/12
19 S_max=S_cc*P*L*c/I
20 printf('part (a)')
21 printf('\n Flexural design stress = %.1f MPa',S_max)
22 //part (b)
23 SF=Sigma_u*I/(S_cc*P*L*c)
24 printf('\n part (b)')
25 printf('\n SF =%.2f ',SF)
```

---

# Chapter 15

## FRACTURE MECHANICS

Scilab code Exa 15.2 Example2

```
1  clc
2  // initialization of variables
3  clear
4  d=250 //mm
5  c=30  //mm
6  t=25  //mm
7  // part (a)
8  a=5   //mm
9  lambda=a/(2*c)
10 f11=1.22 //from the tble
11 f21=1.02
12 //We don't know P yet so say P=1
13 P=1
14 Sfl=P/(t*2*c)*f11+3*280*P*f21/(2*t*c^2)
15 K_IC=59*sqrt(1000)
16 P=K_IC/(Sfl*sqrt(a*pi))
17 printf('part (a)')
18 printf('\n P = %.1f kN',P/10^3)
19 // part (b)
20 a=10 //mm
21 lambda=a/(2*c)
```

```

22 f11=1.33 //from the tble
23 f21=1.05
24 // We don't know P yet so say P=1
25 P=1
26 Sfl=P/(t*2*c)*f11+3*280*P*f21/(2*t*c^2)
27 K_IC=59*sqrt(1000)
28 P=K_IC/(Sfl*sqrt(a*pi))
29 printf('\n part (b)')
30 printf('\n P = %.1f kN',P/10^3)

```

---

### Scilab code Exa 15.3 Example3

```

1 clc
2 // initialization of variables
3 clear
4 a=100/2 //mm
5 Y=1500 //MPa
6 t=6 //mm
7 w=800 //mmm
8 c=200 //mm
9 a_c=a/c
10 f1=1.045
11 w=w*10^-3
12 t=t*10^-3
13 a=a*10^-3
14 A=w*t
15 Sigma=1/A
16 K_I=Sigma*sqrt(pi*a)*f1
17 printf('part (a)')
18 printf('\n K_I = %.2f MPa sqrt(m)',K_I)

```

---

### Scilab code Exa 15.4 Example4

```
1 clc
2 // initialization of variables
3 clear
4 S_u=1300 //MPa
5 K_C=69 // MPa sqrt(m)
6 SF=2.2
7 // calculations
8 S_c=S_u/2.2
9 a=1/%pi*(K_C/S_c)^2
10 printf('a = %.2f mm',a*10^3)
```

---

#### Scilab code Exa 15.5 Example5

```
1 clc
2 // initialization of variables
3 clear
4 // For 30 mm crack
5 a=30/2 // mm crack
6 S_30 =600 //MPa
7 a=a*10^-3
8 C=S_30*sqrt(a)
9 // For 120 mm crack
10 a=120/2
11 a=a*10^-3
12 S_120=C/sqrt(a)
13 printf('Sigma_120 = %d MPa',S_120)
```

---

## Chapter 16

# FATIGUE PROGRESSIVE FRACTURE

Scilab code Exa 16.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  Y=345 //MPa
5  S_u=586 //MPa
6  d=20 //mm
7  R=800 //mm
8  //part (a)
9  SF=1.8
10 N=1e+07
11 S_am=290 //MPa
12 // P_max not yet known. take it as unity until an
    equation to be solved is encountered
13 P_max=1
14 c=d/2
15 M=SF/2*P_max*R //M=T
16 I=%pi*c^4/4
17 Sigma=M*c/I
18 J=%pi*c^4/2
```



```

19 tau=M*c/J
20 S_max=315 //MPa
21 // P_max^2*(3*(tau/S_max)^2+(Sigmaa/S_max)^2)=1
22 P_max=sqrt(1/((3*(tau/S_max)^2)+(Sigma/S_max)^2))
23 P_min=-5/6*P_max
24 printf('part (a)')
25 printf('\n P_max = %d N',P_max)
26 printf('\n P_min = %d N',P_min)

```

---

### Scilab code Exa 16.2 Example2

```

1 clc
2 // initialization of variables
3 clear
4 b=10 //mm
5 M=1
6 t=50 //mm
7 rho=5 //mm
8 h=25 //mm
9 c=60 //mm
10 SF=4.0
11 //part (a)
12 S_cc=2.8
13 q=0.94
14 S_ce=1+q*(S_cc-1)
15 // M is not known. take it as unity
16 S_n=3*M*t/(2*h*(c^3-t^3))
17 S_e=S_ce*S_n
18 printf('part (a)')
19 printf('\n Effective stress = %.1e M',S_e)
20 //part (b)
21 S_max=172 //MPa
22 S_w=S_max/SF
23 M=S_w/S_e
24 printf('\n part (b)')

```

```
25 printf( '\n M =%.1 f N.m' ,M/10^3)
```

---

### Scilab code Exa 16.3 Example3

```
1 clc
2 // initialization of variables
3 clear
4 rho=0.75 //mm
5 S_n=32.97e-06 // M
6 S_cc=6.1
7 q=0.69
8 S_ce=1+q*(S_cc-1)
9 // M is not known. take it as unity
10 M=1
11 S_e=S_ce*S_n
12 printf( 'part (a) ')
13 printf( '\n Effective stress = %.1e M' ,S_e)
14 // part (b)
15 S_w=43 //MPa
16 M=S_w/S_e
17 printf( '\n part (b) ')
18 printf( '\n M =%.1 f N.m' ,M/10^3)
```

---

### Scilab code Exa 16.4 Example4

```
1 clc
2 // initialization of variables
3 clear
4 E=72 //Gpa
5 v=0.33
6 S_u=470 //MPa
7 Y=330 //MPa
8 S_an=190 //MPa
```

```

 9 N=1e+06 // cycles
10 T=10 //mm
11 D=59 //mm
12 d=50 //mm
13 t=3 //mm
14 rho=t
15 P_min=20 //kN
16 q=0.95
17 // calculations
18 P_min=P_min*10^3
19 S_cc=1.90
20 S_ce=1+q*(S_cc-1)
21 A=T*d
22 S_nMin=P_min/A
23 S_nam=S_an/S_ce
24 // (S_na/S_nam)+(S_nm/S_u)^2=1
25 // S_nm^2=S_nMin^2+S_na^2+2*S_na*S_nMin
26 c=S_nMin^2-S_u^2
27 a=1
28 b=2*S_nMin+S_u^2/S_nam
29 S_na=roots([a b c])
30 S_na=S_na(2)
31 // Solving gives S_na
32 S_nm=S_nMin+S_na
33 S_nMax=S_nMin+2*S_na
34 P_max=A*S_nMax
35 S_max=S_nm+S_ce*S_na
36 S_min=S_nm-S_ce*S_na
37 printf('P_max = %.1f kN',P_max/10^3)
38 printf('\n S_max = %.1f MPa',S_max)
39 printf('\n S_min = %.1f MPa',S_min)

```

---

### Scilab code Exa 16.5 Example5

```
1 clc
```

```

2 // initialization of variables
3 clear
4 // Equation given: E_l =E_p + E_e
5 // E_p = 0.58*(2N)^-0.57
6 // E_e=0.0062*(2N)^-0.09
7 // Part (a)
8 function [f]=func(N)
9     f = 0.58*(2*N)^(-0.57)+0.0062*(2*N)^(-0.09)
        -0.01;
10 endfunction
11
12 Nc=6390
13 N=Nc
14 E_p = 0.58*(2*N)^-0.57
15 E_e = 0.0062*(2*N)^-0.09
16 E_l=E_p+E_e
17 printf('Part (a)')
18 printf('\n Total strain = %.5f ',E_l)
19 //part (b)
20 N=1/2*10^6
21 E_p = 0.58*(2*N)^-0.57
22 E_e = 0.0062*(2*N)^-0.09
23 E_l=E_p+E_e
24 printf('\n Part (b)')
25 printf('\n Total strain = %.5f ',E_l)
26 // part (c)
27 E_l=0.01
28 // In order to solve for N We have to solve a non-
        linear equation
29
30 N = 1;//initial guess
31 f = 1;//initial guess
32 while(abs(f)>0.000001),
33     f = func(N);
34     if f>0 then
35         N = N+1;
36     elseif f<0 then
37         N = N-1;

```

```
38     end
39 end
40 printf('\n N = %d cycles.',N);
```

---

# Chapter 17

## CONTACT STRESSES

Scilab code Exa 17.1 Example1

```
1  clc
2  // initialization of variables
3  clear
4  E1=200 //GPa
5  E2=200 //Gpa
6  v1=0.29
7  v2=0.29
8  R1=60 //mm
9  R11=130 //mm
10 R2=80 //mm
11 R22=200 //mm
12 th=%pi/3
13 P=4.5 //kN
14 P=P*10^3
15 E=E1*10^3
16 B=1/4*(1/R1+1/R2+1/R11+1/R22)+1/4*((1/R1+1/R2-1/R11
    -1/R22)^2 - 4*(1/R1-1/R11)*(1/R2-1/R22)*(sin(th)
    ^2))^(1/2)
17 A=1/4*(1/R1+1/R2+1/R11+1/R22)-1/4*((1/R1+1/R2-1/R11
    -1/R22)^2 - 4*(1/R1-1/R11)*(1/R2-1/R22)*(sin(th)
    ^2))^(1/2)
```

```

18 Del=2*(1-v1^2)/(E*(A+B))
19 BAr=B/A
20 Cb=0.77
21 Cs=0.724
22 Ct=0.24
23 Cg=0.22
24 Cz=0.53
25 Cd=2.10
26 b=Cb*(P*Del)^(1/3)
27 br=b/Del
28 S_max=-Cs*br
29 tau_max=Ct*br
30 tau_oct=Cg*br
31 Zs=Cz*b
32 delta=Cd*P/%pi*((A+B)/br)
33 printf('Sigma_max = %d MPa',S_max)
34 printf('\n tau_max = %d MPa',tau_max)
35 printf('\n tau_oct_max = %d MPa',tau_oct)
36 printf('\n Zs = %.2 f mm',Zs)
37 printf('\n delta = %.3 f mm',delta)
38 // S_max doesn't match due to round off error

```

---

### Scilab code Exa 17.2 Example2

```

1 clc
2 // initialization of variables
3 clear
4 E=200 //GPa
5 v=0.29
6 Y=1600 //MPa
7 Po=4.2 //kN
8 Omega=3000 //rpm
9 th=%pi/3
10 P=1.75 //kN
11 R1=4.76 //mm

```

```

12 R11=R1
13 R2=-4.86 //mm
14 R22=18.24 //mm
15 //part (a)
16 E=E*10^3
17 Po=Po*10^3
18 P=P*10^3
19 B=1/4*(1/R1+1/R2+1/R11+1/R22)+1/4*((1/R1+1/R2-1/R11
      -1/R22)^2 - 4*(1/R1-1/R11)*(1/R2-1/R22)*(sin(th)
      ^2))^(1/2)
20 A=1/4*(1/R1+1/R2+1/R11+1/R22)-1/4*((1/R1+1/R2-1/R11
      -1/R22)^2 - 4*(1/R1-1/R11)*(1/R2-1/R22)*(sin(th)
      ^2))^(1/2)
21 Del=2*(1-v^2)/(E*(A+B))
22 BAr=B/A
23 Cb=0.32
24 k=0.075
25 Cs=1.00
26 Ct=0.3
27 Cg=0.27
28 Cz=0.78
29 b=Cb*(P*Del)^(1/3)
30 a=b/k
31 br=b/Del
32 S_max=-Cs*br
33 tau_max=Ct*br
34 tau_oct=Cg*br
35 Zs=Cz*b
36 tauo=0.486*b/(2*Del)
37 Zo=0.41*b
38 printf('b = %.4 f mm',b)
39 printf('\n a = %.3 f mm',a)
40 printf('\n b/Delta = %d MPa',br)
41 printf('\n Sigma_max = %d MPa',S_max)
42 printf('\n tau_max = %d MPa',tau_max)
43 printf('\n tau_oct_max = %d MPa',tau_oct)
44 printf('\n Zs = %.3 f mm',Zs)
45 printf('\n Tau_0 = %d MPa',tauo)

```



```

46 printf('\n Zo = %.3 f mm',Zo)
47
48 // part (b)
49 tau_oY=sqrt(2)*Y/3
50 Py = 1/Del*(tau_oY/(Cg*Cb)*Del)^3
51 printf('\n part (b)')
52 printf('\n P_Y = %d N',Py)
53 SF=Py/P
54 printf('\n SF = %.2 f ',SF)

```

---

### Scilab code Exa 17.3 Example3

```

1  clc
2  // initialization of variables
3  clear
4  E=200 //GPa
5  v=0.29
6  R=40 //mm
7  h=20 //mm
8  P=24.1 //kN
9  S_max=1445 //MPa
10 tau_max=433 //MPa
11 tau_octM=361 //MPa
12 // calculations
13 E=E*10^3
14 P=P*10^3
15 Del=2*R*(1-v^2)/E
16 b=sqrt(2*P*Del/(h*pi))
17 br=b/Del
18 S_maxT=2*b/(9*Del)
19 S_maxC=-1.13*br
20 tau_max=0.31*br
21 tau_octM=0.255*br
22 printf('Sigma_max (tension) = %d MPa',S_maxT)
23 printf('\n Sigma_max (compression) = %d MPa',S_maxC)

```

```
24 printf( '\n tau_max = %d MPa', tau_max)
25 printf( '\n tau_oct_max = %d MPa', tau_octM)
```

---