Scilab Textbook Companion for Advanced Mechanics of Materials by A. P. Boresi and R. J. Schmidt[1]

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

May 2, 2015

[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: 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

Chapter 1

INTRODUCTION

Scilab code Exa 1.1 Example1

```
clc
// initialization of variables
clear
// part (a)
a=700 // M Pa from figure 1.8
b=100 // M Pafrom figure 1.8
m=1/6 // from figure 1.8
Y=450 // M Pa from figure 1.9
//calculations
sigma_u=a+m*b
// results
printf('\n part (a) \n')
printf(' The ultimate strength is sigma = %.f M Pa',
    sigma_u)
printf('\n and the yield strength is Y = %.f M Pa',Y
    )

// part (b)
c1=62 // from figure 1.8
d1=0.025 // from figure 1.8
c2=27 // from figure 1.10a
```

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

Scilab code Exa 1.2 Example2

```
clc
// initialization of variables
clear
sigma=500 // Stress M Pa
eps=0.0073 // Strain
sigma_A=343 // M Pa from figure 1.9
eps_A=0.00172 // from figure 1.9
// part (a)
E=sigma_A/eps_A

// part (B)
eps_e=sigma/E
eps_p=eps-eps_e
// results
printf(' part (a) \n')
printf(' The modulus of elasticity of the rod is E = %.d G Pa',E/1000)
printf('\n part (b)')
printf('\n the permanent strain is = %.4f',eps_p)
printf('\n and the strain recovered is = %.4f',eps_e)
```

Scilab code Exa 1.3 Example3

```
clc
// initialization of variables
clear
D=25 // kN
L=60 // kN
W=30 //kN
Y=250 // M Pa
safety=5/3 // AISC, 1989
// calculations
Q=(D+L+W)*10^3 // converted to N
A=safety*Q/Y
r=sqrt(A/%pi)+0.5 // additional 0.5 mm is for extra
   safety
d=2*r // diameter
// results
printf('Part (a) \n ')
printf('A rod of %.d mm in diameter, with a cross
   sectional area of %.d mm^2, is adequate',d,%pi*d
   ^2/4)
// 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

```
clc
// initialization of variables
clear
sig_xx=-10 // MPa
sig_yy=30 // MPa
sig_xy=15 // MPa
sig_xz=0 // MPa
sig_yz=0 // MPa
sig_zz=0 //MPa
I1=sig_xx+sig_yy+sig_zz
I2=sig_xx*sig_yy-sig_xy^2+sig_zz*sig_xx-sig_xz^2+
   sig_zz*sig_yy-sig_yz^2
M=[sig_xx sig_xy sig_xz
   sig_xy sig_yy sig_yz
   sig_xz sig_yz sig_zz]
I3=det(M)
p=[1 -I1 I2 -I3]
sigma=roots(p)
printf('I1 = %d  I2 = %d  I3 = %d  ',I1,I2,I3)
```

```
printf('\n Sigma_1 = %d  Sigma_2 = %d  SIgma_3 = %d
   ',sigma(1),sigma(3),sigma(2))
// We have:
// {S_xx-S S_xy S_xz
//  S_xy S_yy-S S_yz
//  S_xz S_yz S_zz-S}*{l m n}=0
// Substituting for Sigma_1
a1=sig_xx-sigma(1)
a2=sig_xy
a3=sig_xz
b1=sig_xy
b2=sig_yy-sigma(1)
b3=sig_yz
c1=sig_xz
c2=sig_yz
c3=sig_zz-sigma(1)
// You can solve it using the matrices but since the
   system is imcoplete we get
n1=0
//bl*l1+b2*m1=0
// This implies m1=-b1/b2*l1
// We also have l1^2+m1^2+n1^2=1
l1=1/sqrt(1+(b1/b2)^2)
m1=-b1/b2*l1
printf('\n N1 = %.4fi + %.4fj',l1,m1)
printf('\n or \n N1 = %.4fi + %.4fj',-l1,-m1)
// Similarly Substituting for Sigma_2
a1=sig_xx-sigma(3)
a2=sig_xy
a3=sig_xz
b1=sig_xy
b2=sig_yy-sigma(3)
b3=sig_yz
c1=sig_xz
c2=sig_yz
c3=sig_zz-sigma(3)
// here, l2 = m2 = 0
l2=0
```

```
m2=0
n2=sqrt(1)
printf('\n N2 = %.4fk',n2)
printf('\n or \n N2 = %.4fk',-n2)
// Similarly Substituting for Sigma_3
a1=sig_xx-sigma(2)
a2=sig_xy
a3=sig_xz
b1=sig_xy
b2=sig_yy-sigma(2)
b3=sig_yz
c1=sig_xz
c2=sig_yz
c3=sig_zz-sigma(2)
// On solving , we get
l3=1/sqrt(1+(b1/b2)^2)
m3=-b1/b2*l3
n3=0
printf('\n N3 = %.4fi + %.4fj',l3,m3)
printf('\n or \n N3 = %.4fi + %.4fj',-l3,-m3)
```

Scilab code Exa 2.2 Example2

```
clc
// initialization of variables
clear
sig_xx=20 // MPa
sig_yy=10 // MPa
sig_xy=30 // MPa
sig_xz=-10 // MPa
sig_yz=80 // MPa
I2=-7800 // (MPa)^2
// part (a)
// Assume sig_zz=k and evaluate determinants to solve for k
```

```
det1=sig_xx*sig_yy-sig_xy^2
//det2=k*sig_xx-sig_xz^2
//det3=k*sig_yy-sig_yz^2
k=(I2-det1+sig_xz^2+sig_yz^2)/(sig_xx+sig_yy)
sig_zz=k
I1=sig_xx+sig_yy+sig_zz
M=[sig_xx sig_xy sig_xz
   sig_xy sig_yy sig_yz
   sig_xz sig_yz sig_zz]
I3=det(M)
// p=poly([1 -I1 I2 -I3], "x")
p=[1 -I1 I2 -I3]
sigma=roots(p)
// results
printf('\n part (a) \n')
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)
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

```
clc
// initialization of variables
clear
sig_xx=120 // MPa
sig_yy=55 // MPa
sig_xy=-55 // MPa
sig_xz=-75 // MPa
sig_yz=33 // MPa
sig_zz=-85 // MPa
// Direction cosines at point A
lA=1/sqrt(3)
```

```
mA=1/sqrt(3)
nA=1/sqrt(3)
// Direction cosines at point B
lB=1/sqrt(2)
mB=1/sqrt(2)
nB=0
// calculations
I1=sig_xx+sig_yy+sig_zz
I2=sig_xx*sig_yy-sig_xy^2+sig_zz*sig_xx-sig_xz^2+
   sig_zz*sig_yy-sig_yz^2
M=[sig_xx sig_xy sig_xz
   sig_xy sig_yy sig_yz
   sig_xz sig_yz sig_zz]
I3=det(M)
p=[1 -I1 I2 -I3]
sig=roots(p)
sig=gsort(sig)
sigma(1)=sig(1)
sigma(3)=sig(2)
sigma(2)=sig(3)
// results
printf('\n The principal stresses are sigma1= %.3f ,
    sigma2=%.3f , sigma3=%.3f M Pa',sigma(1),sigma
   (2),sigma(3))
// Finding about the circles
C11=(sigma(2)+sigma(3))/2
C21=(sigma(1)+sigma(3))/2
C31=(sigma(1)+sigma(2))/2
C12=0
C22=0
C32=0
R1=(sigma(2)-sigma(3))/2
R2=(sigma(1)-sigma(3))/2
R3=(sigma(1)-sigma(2))/2
SnnA=lA^2*sigma(1)+mA^2*sigma(2)+nA^2*sigma(3)
SnsA=sqrt(lA^2*sigma(1)^2+mA^2*sigma(2)^2+nA^2*sigma
   (3)^2-SnnA^2)
SnnB=lB^2*sigma(1)+mB^2*sigma(2)+nB^2*sigma(3)
```

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

Scilab code Exa 2.5 Example5

```
clc
// initialization of variables
clear
sig_xx=80 // MPa
sig_yy=60 // MPa
sig_xy=20 // MPa
sig_xz=40 // MPa
sig_yz=10 // MPa
sig_zz=20 // MPa
// Direction cosines at point A
l=1/sqrt(6)
m=2/sqrt(6)
n=1/sqrt(6)
// calculations
SpX=sig_xx*l+sig_xy*m+sig_xz*n
SpY=sig_xy*l+sig_yy*m+sig_yz*n
SpZ=sig_xz*l+sig_yz*m+sig_zz*n
```

```
// result
printf('part (a)')
printf('\n The stress vector is = %.3f i + %.3f j +
    %.3f k',SpX,SpY,SpZ)
// part b
I1=sig_xx+sig_yy+sig_zz
I2=sig_xx*sig_yy-sig_xy^2+sig_zz*sig_xx-sig_xz^2+
    sig_zz*sig_yy-sig_yz^2
M=[sig_xx sig_xy sig_xz
    sig_xy sig_yy sig_yz
    sig_xz sig_yz sig_zz]
I3=det(M)
p=[1 -I1 I2 -I3]
sigma=roots(p)
tau_max=(sigma(1)-sigma(3))/2
tau_oct=sqrt((sigma(1)-sigma(2))^2+(sigma(1)-sigma
    (3))^2+(sigma(2)-sigma(3))^2)*1/3
n=tau_max/tau_oct
printf('\n part (b)')
printf('\n The principal stresses are sigma1= %.3f ,
     sigma2=%.3f , sigma3=%.3f M Pa',sigma(1),sigma
    (2),sigma(3))
printf('\n and maximum shear stress is = %d M Pa',
    tau_max)
printf('\n part (c)')
printf('\n octahedral shear stress is %.3f MPa ',
    tau_oct)
printf('\n Comparing tau_oct and tau_max, we see
    that \n')
printf(' tau_max = %.3f tau_oct',n)
```

Scilab code Exa 2.7 Example7

```
clc
// initialization of variables
```

```
clear
tau_max=160 //MPa
S_max=0
//S_min=-S_o
S_min=S_max-2*tau_max
S_o=-S_min
printf('part (a)')
printf('\n Sigma_o = %d MPa',S_o)
```

Chapter 3

LINEAR STRESS STRAIN TEMPERATURE RELATION

Scilab code Exa 3.1 Example1

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

Scilab code Exa 3.7 Example7

```
clc
// initialization of variables
clear
tR=0.02 // t/R ration
E_A=69 //G Pa
v_A=0.33  // Poisson's ratio
alpha_A=21.6*10^-6 // /degree Celcius (Coefficient of expansion)
E_S=207 // G Pa
v_S=0.280
alpha_S=10.8*10^-6 // /degree Celcius (Coefficient of expansion)
// calculations
// Sig_LA=a*p+b*delT+c*sig_thS
// Sig_LS=v_S*Sig_thS+d*delT
E_S=E_S*10^9
E_A=E_A*10^9
a=1/tR*E_A/E_S
b=-2/3*alpha_S*E_S
c=-E_A/E_S
d=-alpha_S*E_S
// SigthS=e*p+f*delT
// SigthA=g*p+h*delT
e=37.16
f=0.8639*10^6
g=1/tR-e
h=-f
// results
p=689.4 // kPa
delT=100 // degree Celcius
p=p*10^3 // Pa
SigthA=g*p+h*delT
SigthS=e*p+f*delT
Sig_LA=a*p+b*delT+c*SigthS
Sig_LS=v_S*SigthS+d*delT
printf('Thus, for p = %.1f k Pa and delT = %.d degree celcius \n',p/10^3,delT)
printf(' SigthA = %.1f M Pa,    Sig_LA = %.d M Pa \n'
```

```
,SigthA/10^6,Sig_LA/10^6)
printf(' SigthS = %.1f MPa,    Sig_LS = %.d MPa',
   SigthS/10^6,Sig_LS/10^6)
```

Scilab code Exa 3.8 Example8

```
clc
// initialization of variables
clear
// Material constants
Ex=14700 // MPa
Ey=1000 // MPa
Ez=735 // MPa
Gxy=941 // MPa
Gxz=1147 // MPa
Gyz=103 // Mpa
Vxy=0.292
Vxz=0.449
Vyz=0.39
// Stresses at a point
Sxx=7 // Mpa
Syy=2.1 // MPa
Szz=-2.8 //MPa
Sxy=1.4 // MPa
Sxz=0 //MPa
Syz=0 // MPa
// part (a)
th=1/2*atan(2*Sxy/(Sxx-Syy))*180/%pi
I1=Sxx+Syy+Szz
I2=Sxx*Syy-Sxy^2+Szz*Sxx-Sxz^2+Szz*Syy-Syz^2
M=[Sxx Sxy Sxz
   Sxy Syy Syz
   Sxz Syz Szz]
I3=det(M)
p=[1 -I1 I2 -I3]
```

```
S=roots(p)
// results
printf('Part (a) \n')
printf('The maximum principal stress is S1 = %.2f M Pa', S(1))
printf('\n and occurs in direction th = %.1f degrees ',th)
printf('\n and the intermediate principal stress S2 = %.2f M Pa occurs in the direction th = %.1f degrees \n',S(3),th+90)
printf(' The minimum principal stress is S3 = Szz = %.1f M Pa', S(2))
Ex=Ex*10^6
Ey=Ey*10^6
Ez=Ez*10^6
Gxy=Gxy*10^6
Gxz=Gxz*10^6
Gyz=Gyz*10^6
// part (b) is to find strains
Exx=Sxx/Ex-Vxy*Syy/Ey-Vxz*Szz/Ez
Eyy=-Vxy*Sxx/Ex+Syy/Ey-Vyz*Szz/Ez
Ezz=-Vxz*Sxx/Ex-Vyz*Syy/Ey+Szz/Ez
Exy=Sxy/Gxy
Exz=Sxz/Gxz
Eyz=Syz/Gyz
printf('\n Part (b)')
printf('\n The srains are')
printf('\n Exx = %.2e ,   Eyy = %.2e ,   Ezz = %.4e',Exx,Eyy,Ezz)
printf('\n Exy = %.4e ,  Exz = %.2d ,  Eyz = %.2d',Exy,Exz,Eyz)
// Wrong Exx value in the textbook
th=1/2*atan(Exy/(Exx-Eyy))
th=th*180/%pi
th2=th+90
printf('\n part (c)')
printf('\n theta = %.2f  or theta = %.2f degrees',th,th2)
```

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

Chapter 4

INELASTIC MATERIAL BEHAVIOR

Scilab code Exa 4.1 Example1

```
clc
// initialization of variables
clear
P=170 //kN
A=645 // (mm)^2
// part (a)
E=211.4 // G Pa (from figure)
Y=252.6 // M Pa (from figure)
Beta=0.0799 // G Pa (from figure)
Ey=Y/E
// The stress strain law given is
// Sigma= E*eps  for eps< Ey
// Sigma= (1-Beta)*Y + Beta*E*eps  otherwise

// part (b)
th=atan(1.8/2.4)// radians
F=P/(2*cos(th))
F=F*10^3 //N
A=A*10^-6 //m^2
```

```
E=E*10^9 //Pa
Y=Y*10^6 //Pa
L=3.0 //m
Sigma=F/A
if(Sigma<Y)
    eps=Sigma/E
else
   eps=(Sigma-(1-Beta)*Y )/(Beta*E)
end
u=eps*L/cos(th)
u=u*10^3 //mm
// results
printf('part (b)\n')
printf(' Deflection = %.3f mm',u)

// part (c)
P=270 //kN
F=P/(2*cos(th))
F=F*10^3 //N
Sigma=F/A
if(Sigma<Y)
    eps=Sigma/E
else
   eps=(Sigma-(1-Beta)*Y )/(Beta*E)
end
u=eps*L/cos(th)
u=u*10^3 //mm
// results
printf('\n part (c)\n')
printf(' Deflection = %.3f mm for P = %.d kN',u,P)

P=300 //kN
F=P/(2*cos(th))
F=F*10^3 //N
Sigma=F/A
if(Sigma<Y)
    eps=Sigma/E
else
```

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

Scilab code Exa 4.2 Example2

```
clc
// initialization of variables
clear
// Material properties
E=200 //GPa
A=100 //mm^2
Y1=500 //M Pa
Y2=250 // MPa
// calculations
E=E*10^9 // Pa
A=A*10^-6 //m^2
Y1=Y1*10^6 // Pa
Y2=Y2*10^6 //Pa
L_FG=1 //m
L_CD=2 // m
P1=Y2*A
e=P1*L_FG/(E*A)
e_FG=e
e_CD=e
P2=E*A*e_FG/L_FG
P3=E*A*e_CD/L_CD
Py=2*P1+2*P2+P3
//results
printf('part (a) \n')
printf(' Yield Load Py = %.1f kN and the
   displacement is %.2f mm',Py/10^3,e*10^3)
```

```

// part(b)
P4=Y1*A
e=P4*L_FG/(E*A)
P5=E*A*e/L_CD
P=2*P1+2*P4+P5
printf('\n part (b) \n')
printf(' Yield Load P = %.1f kN and the displacement
    is %.2f mm',P/10^3,e*10^3)
// Fully plastic load
P6=Y2*A*2
Pp=2*P1+2*P4+P6
e_CD=P6*L_CD/(E*A)
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

```
clc
// initialization of variables
clear
// Stresses
Sxx=100 // MPa
Syy=-14 // MPa
Sxy=50 // MPa
Y=300 // MPa
// part (a)
Szz=0 // MPa
Syz=0 //MPa
Sxz=0 // MPa
// To calculate principal stresses
I1=Sxx+Syy+Szz
I2=Sxx*Syy-Sxy^2+Szz*Sxx-Sxz^2+Szz*Syy-Syz^2
M=[Sxx Sxy Sxz
   Sxy Syy Syz
```

```
   Sxz Syz Szz]
I3=det(M)
p=[1 -I1 I2 -I3]
Sigma=roots(p)
Smax=Sigma(1)
Smin=Sigma(2)
// Smax=max(Sigma)
// Smin=min(Sigma)
tau_max=Y/2
SF=tau_max*2/(Smax-Smin)
printf('part (a)\n')
printf(' SF = %.2f if the material obeys Tresca criterion',SF)

// part (b)
SF=sqrt(2)*Y/sqrt((Smax^2)+(Smin^2)+(Smin-Smax)^2)
printf('\n part (b)')
printf('\n SF = %.2f if the material obeys von Mises criterion',SF)
```

Chapter 5

APPLICATIONS OF ENERGY METHODS

Scilab code Exa 5.1 Example1

```
clc
// initialization of variables
clear
// part (b)
K1=2 //N/mm (K1=E1A1/L1)
K2=3 //N/mm (K2=E2A2/L2)
b1=400 // mm (b1=h)
h=400 // mm
b2=300 //mm
u=30 //mm
v=40 //mm
// calculations
// Units convertion
K1=K1*10^3
K2=K2*10^3
b1=b1*10^-3
b2=b2*10^-3
h=h*10^-3
u=u*10^-3
```

```
v=v*10^-3
L1=sqrt(b1^2+h^2)
L2=sqrt(b2^2+h^2)
N1=sqrt((b1+u)^2+(h+v)^2)-L1
N2=sqrt((b1+u)^2+(h+v)^2)
N3=sqrt((b2-u)^2+(h+v)^2)-L2
N4=sqrt((b2-u)^2+(h+v)^2)
P=K1*(b1+u)*N1/N2-K2*(b2-u)*N3/N4
Q=K1*(h+v)*N1/N2+K2*(h+v)*N3/N4
// results
printf('part (b)')
printf('\n P = %.1f N',P)
printf('\n Q = %.1f N',Q)
```

Scilab code Exa 5.2 Example2

```
clc
// initialization of variables
clear
// Loads and stresses and dimensions
P=10 //kN
Q=30 //kN
S0=70 //MPa
eps0=0.001
b1=400 //mm
h=400 //mm
b2=300 //mm
A1=300 //mm^2
A2=300 //mm^2
// calculations
// Units convertion
P=P*10^3
Q=Q*10^3
S0=S0*10^6
b1=b1*10^-3
```

```
b2=b2*10^-3
h=h*10^-3
A1=A1*10^-6
A2=A2*10^-6
L1=sqrt(b1^2+h^2)
L2=sqrt(b2^2+h^2)
a=L1*(Q*b2+P*h)/(A1*S0*h*(b1+b2))
b=L2*(Q*b1-P*h)/(A2*S0*h*(b1+b2))
c=L1^2*eps0/(b1+b2)
d=L2^2*eps0/(b1+b2)
u=c*sinh(a)-d*sinh(b)
v=b2/h*c*sinh(a)+b1/h*d*sinh(b)
// results
printf('u = %.4f mm',u*10^3)
printf('\n v = %.4f mm',v*10^3)
```

Scilab code Exa 5.6 Example6

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

Scilab code Exa 5.7 Example7

```
clc
// initialization of variables
clear
// Specifications
T=2 //kN.m
E=72 // G Pa
G=27 // GPa
b=30 //mm
h=40 //mm
d=60 //mm
l1=400 //mm
l2=800 //mm
// calculations
E=E*10^9
G=G*10^9
b=b*10^-3
h=h*10^-3
d=d*10^-3
l1=l1*10^-3
l2=l2*10^-3
T=T*10^3 //N.m
Ix=b*h^3/12
J=%pi*d^4/32
thB= 2*l1^3/3*0.001^2*T/(E*Ix)+T*l2/(G*J)
printf('The rotation of shaft B is th = %.3f rad',
    thB)
// Wrong answer to an extent in the textbook
```

Scilab code Exa 5.9 Example9

```
clc
// initialization of variables
clear
```

```
// specification
R=65 //mm
E=200 //GPa
G=77.5 //GPa
v=0.29
P=6 //kN
//calculations
R=R*10^-3
E=E*10^9
G=G*10^9
P=P*10^3
A=30^2*10^-6
I=30^4/12*10^-12
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)
printf('part (a)')
printf('\n qp = %.4f mm',q_p1*10^3)
//part (b)
// if Un and Us are neglected
q_p2=(9*%pi/4+2)*P*R^3/(E*I)
e=(q_p1-q_p2)/q_p1*100
printf('\n part (b)')
printf('\n error = %.2f per cent',e)
```

Scilab code Exa 5.10 Example10

```
clc
// initialization of variables
clear
// part (b)
// Specifications
P=150 //N
R=200 //mm
d=20 //mm
E=200 //GPa
```

```
G=77.5 //GPa
//calculations
R=R*10^-3
d=d*10^-3
E=E*10^9
G=G*10^9
r1=R+d/2
r2=R-d/2
A=314*10^-6
I=7850*10^-12 //m^4
Ax=3*%pi/4*P*R/(E*A)
Sh=3*%pi/4*1.33*P*R/(G*A)
M=(7*%pi/4+1)*P*R^3/(E*I)
//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)
qc=Ax+Sh+M
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)
printf('\n which means The concentrations of axial
   loads and shear are negligible')
```

Scilab code Exa 5.12 Example12

```
clc
// initialization of variables
clear
// Material properties and dimensions
E=72 //G Pa
P=10 //kN
Q=5 //kN
Aab=150 //mm^2
Abc=900 //mm^2
Acd=900 //mm^2
Ade=900 //mm^2
```

```
Abd=150 //mm^2
Abe=150 //mm^2
Lab=2 //m
Lbc=2.5 //m
Lbd=1.5 //m
Lbe=2.5 //m
Lcd=2 //m
Lde=2 //m
//calculations
E=E*10^9
P=P*10^3
Q=Q*10^3
Aab=150
Aab=Aab*10^-6
Abc=Abc*10^-6
Acd=Acd*10^-6
Ade=Ade*10^-6
Abd=Abd*10^-6
Abe=Abe*10^-6
M=0
Nab=4/3*(Q+2*P)-5*M/(3*Lbe)
dNab=-5/(3*Lbe)
Nbc=-5/3*(Q+P)
dNbc=0
Nbd=Q
dNbd=0
Nbe=5*P/3-4/3*M/Lbe
dNbe=-4/(3*Lbe)
Ncd=-4*P/3+5/3*M/Lbe
dNcd=5/(3*Lbe)
Nde=Ncd
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)
printf('The rotation of member BE is %.5f rad',thBE)
// Wrong answer in the text
```

Chapter 6

TORSION

Scilab code Exa 6.1 Example1

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

```
.4f rad/m',th)
```

Scilab code Exa 6.2 Example2

```
clc
// initialization of variables
clear
T=113 //Nm
L1=1 //m
L2=1.27 //m
Y=414 //MPa
G=77 //GPa
SF=2
// part (a)
T1=T*2
T2=T
Y=Y*10^6
G=G*10^9
tau_max=0.25*Y
r1=(2*T1/(%pi*tau_max))^(1/3)
d1=2*r1
r2=(2*T2/(%pi*tau_max))^(1/3)
d2=2*r2
inch=25.4 //mm
printf(' part (a) \n')
printf(' d1 = %.2f mm   d2 = %.2f mm',d1*10^3,d2*10^3)
printf('\n Since the dimensons are not standard, we choose d1 = %.1f mm and d2 = %.2f mm',inch,0.75*inch)
// part (b)
d1=inch*10^-3
r1=d1/2
d2=0.75*inch*10^-3
r2=d2/2
```

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

Scilab code Exa 6.3 Example3

```
clc
// initialization of variables
clear
tau_Y=190 //MPa
G=27 //GPa
L=2 //m
Do=60 //mm
Di=40 //mm
SF=2 // Factor of safety
// Angle of twist can't be greater than 0.2 rad
thM=0.2 //rad
Do=Do*10^-3
Di=Di*10^-3
G=G*10^9
tau_Y=tau_Y*10^6
J=%pi/2*((Do/2)^4-(Di/2)^4)
T=tau_Y*J*2/(Do*SF)
printf(' part (a)')
printf('\n Design torque T = %.3f kN.m',T/10^3)

// part (b)
T=G*J*thM/SF
```

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

Scilab code Exa 6.4 Example4

```
clc
// initialization of variables
clear
// Material specifications
G=77.5 //GPa
// Following values of torsion are obtained from
   figure
Toa=-12.5 //kN
Tab=-8.5 //kN
Tbc=1.5 //kN
D1=10 //cm
D2 =5 //cm
D3 =D1 //cm
Loa=500 //mm
Lab=400 //mm
Lbc=300 //mm
//calculations
G=G*10^9
Toa=Toa*10^3
Tab=Tab*10^3
Tbc=Tbc*10^3
D1=D1*10^-2
D2=D2*10^-2
D3=D3*10^-2
Loa=Loa*10^-3
Lab=Lab*10^-3
Lbc=Lbc*10^-3
r1=D1/2
Joa=%pi*r1^4/2
```

```
tauOA=-Toa*D1/(2*Joa)
r2=D2/2
r3=D3/2
Jbc=%pi*r2^4/2
Jab=%pi*r3^4/2
tauBC=Tbc*D2/(2*Jbc)
tau=max(tauOA,tauBC)
printf('The maximum shear stress is = %.2f M Pa in
    segment OA',tau/10^6)
// part (b)
psiA=Toa*Loa/(G*Joa)
psiBA=Tab*Lab/(G*Jab)
psiB=psiA+psiBA
psiCB=Tbc*Lbc/(G*Jbc)
psiC=psiB+psiCB
printf('\n PsiA = %.5f rad  PsiB = %.5f rad  PsiC =
    %.5f rad ',psiA,psiB,psiC)
```

Scilab code Exa 6.5 Example5

```
clc
// initialization of variables
clear
// Shaft specifications
Pi=100 //kW
f1=100 //Hz
f2=10 //Hz
tau_Y=220 //MPa
SF=2.5 // Safety factor
Po=100 //kW
//calculations
Pi=Pi*10^3
tau_Y=tau_Y*10^6
Po=Po*10^3
Tin=Pi/(2*%pi*f1)
```

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

Scilab code Exa 6.7 Example7

```
clc
// initialization of variables
clear
// Flange specifications
T=5000 //Nm
b_f=266 //mm
d=779 //mm
t_w=16.5 //mm
t_f=30 //mm
G=200 // GPa
// calculations
b_f=b_f*10^-3
d=d*10^-3
t_w=t_w*10^-3
t_f=t_f*10^-3
G=G*10^9
//calculations
k1=0.308 // flange (b/h)<10
Jf=2*k1*b_f*t_f^3
k1=0.333 // web (b/h)>10
Jw=k1*(d-2*t_f)*t_w^3
J=Jf+Jw
// part (a)
hmax=0.015
tau_max=2*T*hmax/J
printf('part (a)\n')
printf(' Maximum shear stress is = %.2f MPa',tau_max
```

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

Scilab code Exa 6.8 Example8

```
clc
// initialization of variables
clear
// Rod dimensions and material properties
b1=60 //mm
l1=3 //m
l2=1.5 //m
h1=40 //mm
b2=40 //mm
h2=30 //mm
G=77.5 //GPa
T1=750 //Nm
T2=400 //Nm
//calculations
b1=b1*10^-3
h1=h1*10^-3
b2=b2*10^-3
h2=h2*10^-3
G=G*10^9
// for the left portion of the rod
k1l=0.196
k2l=0.231
// for the right portion of the rod
k1r=0.178
k2r=0.223
T=T1+T2
```

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

Scilab code Exa 6.9 Example9

```
clc
// initialization of variables
clear
Do=22 //mm
Di=18 //mm
Dm=20 //mm
tD=0.1 // t/D
//part (a)
tau=70 //MPa
G=77.5 //GPa
//calculations
Do=Do*10^-3
Di=Di*10^-3
Dm=Dm*10^-3
tau=tau*10^6
G=G*10^9
A=%pi*Dm^2/4
t=Dm*tD
T1=2*A*tau*t
th1=tau*%pi*Dm/(2*G*A)
J=%pi/32*(Do^4-Di^4)
r=Dm/2
```

```
T2=tau*J/r
th2=tau/(G*r)
printf('part (a)\n')
printf(' Using formula_1 T = %.2f Nm theta = %.7f
    rad/mm ',T1,th1*10^-3)
printf('\n Using formula_2 T = %.2f Nm theta = %.7f
    rad/mm ',T2,th2*10^-3)
//part (b)
h=1 //mm
h=h*10^-3
b=10*%pi
b=b*10^-3
T=8*b*h^2*tau/3
th=tau/(2*G*h)
printf('\n part (b)')
printf('\n T = %.3f N.m     theta = %.7f rad/mm ',T,
    th*10^-3)
```

Scilab code Exa 6.10 Example10

```
clc
// initialization of variables
clear
G=26 //GPa
tau_max=40.0 //MPa
t1=4.5 //mm
t3=1.5 //mm
t2=3 //mm
l1=3*60 //mm
l3=60 //mm
r2=30 //mm

//calculations
// 1 indicates coefficient of q1
// 2 indicates coefficient of q2
```

```
l2=r2*%pi
G=G*10^3
A1=l3^2
A2=%pi*r2^2/2
T1=2*A1
T2=2*A2
tha1=l1/t1+l3/t3
tha1=tha1/(2*G*A1)
tha2=-l3/t3
tha2=tha2/(2*G*A1)
thb1=-l3/t3
thb1=thb1/(2*G*A2)
thb2=l2/t2+l3/t3
thb2=thb2/(2*G*A2)
// Since tha=thb
Qr=(thb2-tha2)/(tha1-thb1)
printf('q1/q2 = %.3f ',Qr)
q2=tau_max*t2
q1=Qr*q2
qdif=q1-q2
tau_1=q1/t1
tau_2=q2/t2
tau_3=qdif/t3
T=2*A1*q1+2*A2*q2
th=tha1*q1+tha2*q2
printf('\n T = %.3f kN.m',T/10^6)
printf('\n theta = %.4f rad/m',th*10^3)
```

Chapter 7

BENDING OF STRAIGHT BEAMS

Scilab code Exa 7.1 Example1

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

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

Scilab code Exa 7.2 Example2

```
clc
// initialization of variables
clear
P1=1.5 //kN
P2=4.5 //kN
// part (a)
A=1000 //mm^2
A1=500 //mm^2
A2=500 //mm^2
//calculation
A=A*10^-6
A1=A1*10^-6
A2=A2*10^-6
y1=25*10^-3
y2=55*10^-3
c1=(A1*y1+A2*y2)/A
c2=60*10^-3-c1 // c1+c2=60 mm
y_1=c1-25*10^-3
y_2=c2-5*10^-3
b1=50*10^-3
h1=10*10^-3
h2=50*10^-3
b2=10*10^-3
Ix=1/12*b1*h1^3 + A1*y_1^2 + 1/12*b2*h2^3 + A2*y_2^2
printf('part (a)')
R1=2550 //N
Vy=750 //N
Mx=975 //N.m
S_zzT=Mx*c1/Ix
S_zzC=Mx*(-c2)/Ix
```

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

Scilab code Exa 7.3 Example3

```
clc
// initialization of variables
clear
P=12 //kN
Phi=%pi/3
// calculations
L=3 //m
P=12 //kN
A=10000 //mm^2
Ix=39.69*10^6 //mm^4
yo=82 //mm
Iy=30.73*10^6 //mm^4
Ixy=0
P=P*10^3
Ix=Ix*10^-12
Iy=Iy*10^-12
alpha=atan(-Ix/(Iy*tan(Phi)))
M=-L*P
Mx=M*sin(Phi)
yA=-118*10^-3 //m
xA=-70*10^-3 //m
xB=-xA
yB=82*10^-3 //m
S_A=Mx*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
S_B=Mx*(yB-xB*tan(alpha))/(Ix-Ixy*tan(alpha))
printf(' Sigma A = %.1f M Pa \n',S_A/10^6)
printf(' Sigma B = %.1f M Pa',S_B/10^6)
```

Scilab code Exa 7.4 Example4

```
clc
// initialization of variables
clear
P=4 //kN
L=1.2 //m
A=1900 //mm^2
Ix=2.783*10^6 //mm^4
Iy=1.003*10^6 //mm^4
Ixy=-0.973*10^6 //mm^4
P=P*10^3
Ix=Ix*10^-12
Iy=Iy*10^-12
Ixy=Ixy*10^-12
A=1900 //mm^2
xo=19.74 //mm
yo=39.74 //mm
Phi=2*%pi/3
Nr=Ixy-Ix/tan(Phi)
Dr=Iy-Ixy/tan(Phi)
alpha=atan(Nr/Dr)
M=L*P
Mx=M*sin(Phi)
yA=39.74*10^-3 //m
xA=-60.26*10^-3 //m
xB=19.74*10^-3
yB=-80.26*10^-3 //m
S_A=Mx*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
S_B=Mx*(yB-xB*tan(alpha))/(Ix-Ixy*tan(alpha))
printf('part (a)')
printf('\n Sigma A = %.1f M Pa \n',S_A/10^6)
printf(' Sigma B = %.1f M Pa',S_B/10^6)
```

```
// part (b)
th=1/2*atan(-2*Ixy/(Ix-Iy))
th1=0.415 //rad
th2=-1.156 //rad
IX=Ix*(cos(th1))^2+Iy*(sin(th1))^2-2*Ixy*sin(th1)*
    cos(th1)
IY=Ix+Iy-IX
Phi=2*%pi/3-th1
alphA=-IX/(IY*tan(Phi))
alpha=alphA+th1
XA=xA*cos(th1)+yA*sin(th1)
YA=yA*cos(th1)-xA*sin(th1)
XB=xB*cos(th1)+yB*sin(th1)
YB=yB*cos(th1)-xB*sin(th1)
MX=M*sin(Phi)
MY=-M*cos(Phi)
S_A=MX*YA/IX-MY*XA/IY
S_B=MX*YB/IX-MY*XB/IY
printf('\n part (b)')
printf('\n Sigma A = %.1f M Pa \n',S_A/10^6)
printf(' Sigma B = %.1f M Pa',S_B/10^6)
```

Scilab code Exa 7.5 Example5

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

```
ybar=ybar*1e-3
Mxk=-6.1*cos(%pi/6) // Mx=Mxk*P
Myk=-6.1*sin(%pi/6) //My=Myk*P
// Equation to be followed
// S_zz=Mx*y/Ix-My*x/Iy
// At A x=100 mm  y=-92.36 mm
x=100e-3
y=-92.36e-3
S_zzA=Mxk*y/Ix-Myk*x/Iy //Sigma_zz=S_zz*P
// At B x=-100 mm  y=-92.36 mm
x=-100e-3
y=-92.36e-3
S_zzB=Mxk*y/Ix-Myk*x/Iy //Sigma_zz=S_zz*P
// At C x=-3.125 mm  y=207.64 mm
x=-3.125e-3
y=207.64e-3
S_zzC=Mxk*y/Ix-Myk*x/Iy //Sigma_zz=S_zz*P
// To find P
P=2*tau_max/max(S_zzA,S_zzB,S_zzC)
printf('P = %.2f kN',P/10^3)
```

Scilab code Exa 7.6 Example6

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

```
Ix=Ix*10^-12
Iy=Iy*10^-12
alpha=atan(-Ix/(Iy*tan(Phi)))
M=P*L/4
Mx=M*sin(Phi)
yA=-118*10^-3 //m
xA=70*10^-3 //m
xB=-xA
yB=82*10^-3 //m
S_comp=Mx*(yA-xA*tan(alpha))/(Ix-Ixy*tan(alpha))
S_tens=Mx*(yB-xB*tan(alpha))/(Ix-Ixy*tan(alpha))
printf(' Tensile strength = %.1f M Pa \n',S_tens
    /10^6)
printf(' Compressive Strength = %.1f M Pa',S_comp
    /10^6)
v=P*L^3*sin(Phi)/(48*E*Ix)
u=-v*tan(alpha)
delta=sqrt(u^2+v^2)
printf('\n The total deflection is %.2f mm',delta
    *10^3)
```

Scilab code Exa 7.7 Example7

```
clc
// initialization of variables
clear
L=3 //m
Ix=56.43e6 //mm^4
Iy=18.11e6 //mm^4
Ixy=22.72e6 //mm^4
Phi=%pi/3
E=200e9 //Pa
Y=300e6 //Pa
//calculations
Ix=Ix*10^-12
```

```
Iy=Iy*10^-12
Ixy=Ixy*10^-12
yA=-120*10^-3 //m
xA=-91*10^-3 //m
Nr=Ixy-Ix/tan(Phi)
Dr=Iy-Ixy/tan(Phi)
alpha=atan(Nr/Dr)
// M=-L*P To know P we do the following
Mxk=-L*sin(Phi)//Mx=Mxk*P
P=Y*(Ix-Ixy*tan(alpha))/(Mxk*(yA-xA*tan(alpha)))
printf('P = %.2f kN \n',P/10^3)
v=P*L^3*sin(Phi)/(3*E*(Ix-Ixy*tan(alpha)))
u=-v*tan(alpha)
delta=sqrt(u^2+v^2)
printf(' deflection = %.2f mm',delta*10^3)
// Wrong calculation starting from v in Textbook
```

Scilab code Exa 7.8 Example8

```
clc
// initialization of variables
clear
Ix=937e+06 //mm^4
Iy=18.7e+6 //mm^4
Ixy=0
yA=305 //mm
xA=90.5 //mm
Phi=1.5533 //rad
//calculations
Ix=Ix*10^-12
Iy=Iy*10^-12
Ixy=Ixy*10^-12
yA=yA*10^-3 //m
xA=xA*10^-3 //m
alpha=atan(-Ix/(Iy*tan(Phi)))
```

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

Scilab code Exa 7.9 Example9

```
clc
// initialization of variables
clear
Y=280 //MPa
AB=40 //mm
BC=60 //mm
//calculations
Y=Y*10^6
alpha=atan(BC/AB)
C11=20/3 //mm
C12=-10 //mm
C21=-20/3 //mm
C22=10 //mm
Beta=atan((C11-C21)/(C22-C12))
Phi=%pi/2+Beta
d=sqrt((AB/2-C11)^2+(BC/2-C22)^2)
d=d*10^-3 //m
At=1/2*AB*BC/2*10^-6
```

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

Chapter 8

SHEAR CENTER FOR THIN WALL BEAM CROSS SECTIONS

Scilab code Exa 8.1 Example1

```
clc
// initialization of variables
clear
t=4 //mm
// calculations
l1=100 //mm See figure
l2=50 //mm See figure
ybar=125 //mm
t=t*10^-3
ybar=ybar*10^-3
l1=l1*10^-3
l2=l2*10^-3
Ix=2*t*(2*(l1+l2))^3/12-t*(2*l1)^3/12
qAk=l1*t*ybar // qA=qAk*V
qBk=qAk+l1*t*l1/2
qave=qAk+2/3*(qBk-qAk)
F2k=200*qave*10^-3 //F2=F2k*V
```

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

Scilab code Exa 8.2 Chapter8 Example 2

```
clc
// initialization of variables
clear
// Defining the legs
a=50 //mm Top horizontal leg
b=100 //mm Verical leg
c=100 //mm bottom leg
t=4 //mm
Ix=1.734e6 //mm^4
Iy=0.876e6 //mm^4
Ixy=-0.5e6 //mm^4
I=[Ix Ixy
   Ixy Iy]
theta=1/2*atan(-2*Ixy/(Ix-Iy))
Q=[cos(theta) -sin(theta)
   sin(theta) cos(theta)]
I_1=Q*I*Q' // I_1=[IX IXY| IXY IY]
// Finding out the centroidal coordinates
// We have x_bar = Summation(Ai*Xi)/Summation(Ai)
// We take D as reference
Aa=a*t
Ab=b*t
Ac=c*t
A=Aa+Ab+Ac
x_D=((Ac*c/2)+(Aa*a/2))/A
y_D=((Ab*b/2)+(Aa*b))/A
//Finding out B coordinates
```

```
xb=a-x_D
yb=b-y_D
x=[xb;yb]
X=Q'*x //New coordinates of B in transformed system
function y=f(l),
    y=t*l/I_1(1)*(X(2)+1/2*l*sin(theta)),
endfunction
F3=intg(0,a,f) // This is the coefficient of VY
e_X=b*F3
printf('eX = %.2f mm',e_X)
// To find eY
function y1=g(l),
    y1=t*l/I_1(4)*(X(1)-1/2*l*cos(theta)),
endfunction
F3=intg(0,a,g) // This is the coefficient of VX
e_Y=b*F3
printf('\n eY = %.2f mm',e_Y)
XC=Q'*[x_D
       y_D]
XC=XC+[e_X
      -e_Y]
printf('\n In terms of intial coordinates, the shear
    center C is located at \n XC = %.2f mm',XC(1))
printf('\n YC = %.2f mm',XC(2))
xC=Q*XC
printf('\n The x and y coordinates of shear center C
    are \n xC = %.2f mm',xC(1))
printf('\n yC = %.2f mm',xC(2))
```

Scilab code Exa 8.3 Example3

```
clc
// initialization of variables
clear
t1=1 //mm
```

```
t2=2 //mm
oT=9.67 //mm distance between base and the centroid of each T-ection
y2_bar=100+10+1+oT //mm (follwos from the figure)
A1=400 //mm^2
y1_bar=100 //mm
A2=324 //mm^2
Ix=2*A1*y1_bar^2+2*A2*y2_bar^2
q1k=A2*y2_bar //q1=q1k*Vy/Ix
F1k=(oT+t1/2)*q1k // Fi=Fik*Vy/Ix
F2k=60*q1k
F3k=(10+t1/2)*q1k
q2k=q1k+(A1*y1_bar)
F4k=(10+t2/2)*q2k
F5k=200*q2k
V_pk=2*(F1k+2*F3k+F5k)/Ix // V_p=V_pk*Vy
e=(-2*F1k*71-2*F3k*11+F2k*221+F4k*200)/Ix
printf('e = %.2f mm',e)
```

Scilab code Exa 8.4 Example4

```
clc
// initialization of variables
clear
b=300 //mm
h=500 //mm
t1=20 //mm
t2=10 //mm
t3=t2
Ix=687.5e+06 //mm^4
q_P=b*t2*h/2
q_Q=q_P+h/2*t1*h/4
q_S=h/2*t3*h/4
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)
```

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

Scilab code Exa 8.5 Example5

```
clc
// initialization of variables
clear
a=500 //mm
b=a
t1=5 //mm
t2=10 //mm
t3=20 //mm
// calculations
Ix=2343.75e+06//mm^4
q_B=b*t2*a/2
q_C=q_B+a/2*t1*a/4
q_S=a/2*t3*a/4
q_G=2*b*t2*a/2
q_H=q_G+a/2*t3*a/4
// th_L = th_R = 0
// Writing the above in following form
//Ab=c ; b={q_A q_F}
A11=a/t1+b/t2+a/t3+b/t2
A12=a/t3
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
A21=A12
A22=a/t3+2*b/t2+a/t3+2*b/t2
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
A=[A11 A12
   A21 A22]
c=[c1
   c2]
b=inv(A)*c
q_A=b(1)/1000 //kN/mm
q_F=b(2)/1000 //kN/mm
q_B=q_B/1000
q_C=q_C/1000
q_S=q_S/1000
q_G=q_G/1000
q_H=q_H/1000
b=a // rewriting to it's initival value
 // To find out e, balance the moments
 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)
e=e/Ix
 printf('e = %.1f mm',e*10^3)
 // Wrong answer in the text
```

Chapter 9

CURVED BEAMS

Scilab code Exa 9.1 Example1

```
clc
// initialization of variables
clear
a=30 //mm
c=80 //mm
b=50 //mm
P=9.5 //kN
d=100 //mm position of P
//calculations
P=P*10^3
A=b^2
A=b*(c-a)
Am=b*log(c/a)
R=(a+c)/2
p=d+R
Mx=p*P
r=a
S_thB=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
r=c
S_thC=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
printf('The maximum tensile stress is (at point B) =
```

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

Scilab code Exa 9.2 Example2

```
clc
// initialization of variables
clear
// part (c)
r_A=1.47 //m
theta=%pi
// S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
    theta))*10^5 //kPa
r=r_A
S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
    theta))*10^5 //kPa
S_A=S_th

r_B=1.53 //m
r=r_B
S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
    theta))*10^5 //kPa
S_B=S_th
printf('part (c)')
printf('\n The tensile stress at A is %.2f MPa',S_A
    /1000)
printf('\n The compressive stress at B is %.2f MPa',
    S_B/1000)

// part (d)
theta=%pi/2
r=r_A
S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
    theta))*10^5 //kPa
```

```
S_A=S_th
r=r_B
S_th=-125*cos(theta)+(14.2857-9.5250*r)/r*(1-cos(
    theta))*10^5 //kPa
S_B=S_th
printf('\n part (d)')
printf('\n The tensile stress at A is %.2f MPa',S_A
    /1000)
printf('\n The compressive stress at B is %.2f MPa',
    S_B/1000)

//part (e)
```

Scilab code Exa 9.3 Example3

```
clc
// initialization of variables
clear
Y=500 //MPa
SF=2.00
A1=1658.76 //mm^2
R1=73.81 //mm
Am1=22.64 //mm
A2=6100 //mm^2
R2=126.62 //mm
Am2=50.57 //mm
A3=115.27 //mm^2
R3=186.01 //mm
Am3=0.62 //mm
A=A1+A2+A3
Am=Am1+Am2+Am3
R=(A1*R1+A2*R2+A3*R3)/A
rB=60 //mm
rC=rB+24+100+5 //follows from figure
//P unknown, so put unity to solve for it later
```

```
P=1
Mx=116.37*P
r=rB
S_thB=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
r=rC
S_thC=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))
S_th=max(abs(S_thB),abs(S_thC))
Pf=Y/S_th
P=Pf/SF
printf('P = %.d N',P)
```

Scilab code Exa 9.4 Example4

```
clc
// initialization of variables
clear
// part (b)
// Following is the formula used in evaluating the circumferential stress
// Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
// Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
// S_th=P*((sin(th)-th*cos(th))/(%pi*(r0-ri)*t))*(1+Nr/Dr)
ri=60 //mm
ro=180 //mm
t=50 //mm
th=%pi/2
// For, maximum tensile stress r=ri
r=ri
Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
// Question was asked in terms of P, so let it be unity
P=1
S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/
```

```
    Dr)
S_max1=S_th
// For maximum compressive stress r=ro
r=ro
Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/Dr)
S_max2=S_th
printf('part (b)')
printf('\n for theta=90 degrees')
printf('\n Maximum tensile stress = %.6f P',S_max1)
printf('\n Maximum compressive stress = %.6f P',S_max2)


th=%pi
// For, maximum tensile stress r=ri
r=ri
Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
// Question was asked in terms of P, so let it be unity
P=1
S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/Dr)
S_max1=S_th
// For maximum compressive stress r=ro
r=ro
Nr=(ro+ri)*(ro-ri-r*log(ro/ri))
Dr=r*((ro+ri)*log(ro/ri)-2*(ro-ri))
S_th=P*((sin(th)-th*cos(th))/(%pi*(ro-ri)*t))*(1+Nr/Dr)
S_max2=S_th
printf('\n for theta=180 degrees')
printf('\n Maximum tensile stress = %.6f P',S_max1)
printf('\n Maximum compressive stress = %.6f P',S_max2)
```

```

//part(c)
S_thMax=340 //MPa
SF=2.2
P=S_thMax/(SF*S_max1)
printf('\n part(c)')
printf('\n The maximum allowable load is %.2f kN',P/1000)
```

Scilab code Exa 9.5 Example5

```
clc
// initialization of variables
clear
P=120 //kN
b1=120 //mm
b2=120 //mm
h1=48 //mm
h2=24 //mm
P=P*10^3
A=h1*b1+b2*h2
R=(b1*h1*96+b2*h2*180)/A
Am=b1*log(b1/72)+h2*log(240/b2)
r=72
Mx=364*P
S_thB=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))

r1=120 //mm
t=24 //mm
A1=h1*r1
Am1=r1*log(r1/r)
S_rr=(A*Am1-A1*Am)*Mx/(t*r1*A*(R*Am-A))
printf('Circumferential stress is %.1f MPa',S_thB)
printf('\n Radial stress is %.1f MPa',S_rr)
```

Scilab code Exa 9.6 Example6

```
clc
// initialization of variables
clear
Mo=96 //kN
P=120 //kN
b1=150 //mm
h1=60 //mm
b2=120 //mm
h2=50 //mm
b3=b1
h3=40 //mm
ro=80 //mm
r1=140 //mm
r2=260 //mm
r3=300 //mm
// calculations
Mo=Mo*10^6 // N.mm
P=P*10^3 // N
A=b1*h1+b2*h2+b3*h3
Am=b1*log(r1/ro)+h2*log(r2/r1)+b3*log(r3/r2)
R=(b1*h1*110+b2*h2*200+b3*h3*280)/A
Mx=Mo+P*R
r=80 //mm
S_th=P/A+(Mx*(A-r*Am))/(A*r*(R*Am-A))

A1=9000 //mm^2
r1=140 //mm
t=50 //mm
Am1=b1*log(r1/ro)
N=120000
S_rr=A1*N/(A*t*r1) + (A*Am1-A1*Am)*Mx/(t*r1*A*(R*Am-
    A))
```

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

Scilab code Exa 9.7 Example7

```
clc
// initialization of variables
clear
l=15 //m
R=10 //m
d=0.8 //m
b=0.13 //m
Po=2400 //N/m
P=4800 //N/m
//calculations
a=R-d/2
c=R+d/2
A=b*d
Am=b*log(c/a)
Mx=(Po+P)*l^2/8
S_thMax=Mx*(A-a*Am)/(A*a*(R*Am-A))
// To find out max radial stress
// Nr=d*log(r/a)-(r-a)*log(c/a)
// Dr=r*d*(R*log(c/a)-d)
// S_rr=Mx/b*Nr/Dr
r=a*exp(1-(a/d*log(c/a)))
Nr=d*log(r/a)-(r-a)*log(c/a)
```

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

Scilab code Exa 9.8 Example8

```
clc
// initialization of variables
clear
// part(a)
Y=280 //MPa
A=4000 //mm^2
Am=44.99 //mm
R=100.0 //mm
r=180 //mm
r=60 //mm
// Mx is not yet known take it as unity
Mx=1 //unity
r=180
S_thMax=Mx*(A-r*Am)/(A*r*(R*Am-A))
Mx=Y/(abs(S_thMax))
printf('part(a)')
printf('\n Mx = %.2f kN.m',Mx/10^6)
// part(b)
k1=1.143
```

```
t_w=20
b_p=40
alpha=0.651
Beta=1.711
r=60 //mm
b1=2*alpha*b_p+t_w
A=b1*t_w+t_w*R
R=(b1*t_w*70+t_w*R*130)/A
Am=b1*log(80/r)+t_w*log(180/80)
// Mx not yet known teke it as unity
Mx=1
S_thMax=Mx*(A-r*Am)/(A*r*(R*Am-A))
r=70 //mm
S_thbar=Mx*(A-r*Am)/(A*r*(R*Am-A))
S_xx=-Beta*S_thbar
//tau_max=Y/2=(S_thMax-S_xx)/2
Mx=Y/(S_thMax-S_xx)
printf('\n part (b)')
printf('\n Mx = %.2f kN.m',Mx/10^6)
```

Scilab code Exa 9.9 Example9

```
clc
// initialization of variables
clear
E=72 //GPa
t=60 //mm
M=24 //kN.m
//part(a)
ro=100 //mm
r1=150 //mm
A=t*r1
Am=t*log((ro+r1)/ro)
R=ro+r1/2
E=E*10^3
```

```
Mx=M*10^6
Phi=Am*Mx*%pi/(A*(R*Am-A)*E)
printf('part(a)')
printf('\n Phi = %.5f rad',Phi)
//part(b)
//Mx=Mx+P*R*sin(th)
delta_P=Am*Mx*R*2/(A*(R*Am-A)*E)
printf('\n part(b)')
printf('\n deflection = %.3f mm',delta_P)
```

Scilab code Exa 9.10 Example10

```
clc
// initialization of variables
clear
P=11.2 //kN
E=200 //GPa
v=0.3
Ix=181.7e+03 //mm^4
k1=0.643
b1=34.7 //mm
h1=10 //mm
b2=40 //mm
h2=10 //mm
t=10 //mm
h=50 //mm
E=E*10^3
A=b1*h1+b2*h2
R=(b1*h1*35+b2*h2*60)/A
Am=b1*log(40/30)+h1*log(80/40)
G=E/(2*(1+v))
Aw=t*h
P=P*10^3
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)
printf('seperation = %.3f mm',delta_P)
```

Chapter 10

BEAMS ON ELASTIC FOUNDATIONS

Scilab code Exa 10.1 Example1

```
clc
// initialization of variables
clear
//part(a)
E=200 //GPa
d=184 //mm
c=99.1 //mm
Ix=36.9e+06//mm^4
k=14.0 //N/mm^2
P=170 //kN
//calculations
E=E*10^3
P=P*10^3
Beta=(k/(4*E*Ix))^(1/4)
y_max=P*Beta/(2*k)
M_max=P/(4*Beta)
S_max=M_max*c/Ix
printf('part (a)')
printf('\n y_max = %.3f mm',y_max)
```

```
printf('\n M_max = %.2f kN.m',M_max/10^6)
printf('\n S_max = %.1f MPa',S_max)
// part (b)
z1=1.7//m
z1=z1*10^3 //mm
z2=2*z1
// A_bz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
// C_bz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
A_bzo=1
C_bzo=1
A_bz1=exp(-Beta*z1)*(sin(Beta*z1)+cos(Beta*z1))
A_bz2=exp(-Beta*z2)*(sin(Beta*z2)+cos(Beta*z2))
C_bz1=exp(-Beta*z1)*(-sin(Beta*z1)+cos(Beta*z1))
C_bz2=exp(-Beta*z2)*(-sin(Beta*z2)+cos(Beta*z2))
y_end=P*Beta/(2*k)*(A_bzo+A_bz1+A_bz2)
M_end=P/(4*Beta)*(C_bzo+C_bz1+C_bz2)
y_center=P*Beta/(2*k)*(A_bzo+2*A_bz1)
M_center=P/(4*Beta)*(C_bzo+2*C_bz1)
y_max=max(y_end,y_center)
M_max=max(M_end,M_center)
S_max=M_max*c/Ix
printf('\n part(b)')
printf('\n y_max = %.3f mm',y_max)
printf('\n M_max = %.2f kN.m',M_max/10^6)
printf('\n S_max = %.1f MPa',S_max)
```

Scilab code Exa 10.2 Example2

```
clc
// initialization of variables
clear
d=100 //mm
Ix=2.45e+06 //mm^4
E=72 //GPa
L=6.8 //m
```

```
K=110 //N/mm
l=1.1 //m
P=12 //kN
//calculations
E=E*10^3
P=P*10^3
l=l*10^3
k=K/l
L1=7*l
Beta=(k/(4*E*Ix))^(1/4)
if(l<%pi/(4*Beta))
if(L1>3*%pi/(2*Beta))
    y_max=P*Beta/(2*k)
    M_max=P/(4*Beta)
    S_max=M_max*d/(2*Ix)
end
end
printf('y_max = %.3f mm',y_max)
printf('\n M_max = %.2f kN.m',M_max/10^6)
printf('\n S_max = %.1f MPa',S_max)
A_bl=exp(-Beta*l)*(sin(Beta*l)+cos(Beta*l))
A_2bl=exp(-Beta*2*l)*(sin(Beta*2*l)+cos(Beta*2*l))
A_3bl=exp(-Beta*3*l)*(sin(Beta*3*l)+cos(Beta*3*l))
    y_C=P*Beta/(2*k)*A_bl
    y_B=P*Beta/(2*k)*A_2bl
    y_A=P*Beta/(2*k)*A_3bl
printf('\n y_C = %.2f mm',y_C)
printf('\n y_B = %.2f mm',y_B)
printf('\n y_A = %.2f mm',y_A)
```

Scilab code Exa 10.4 Example4

```
clc
// initialization of variables
clear
```

```
E=10 //GPa
h=200 //mm
b=100 //mm
ko=0.04 //N/mm^3
w=35 //N/mm
L1=3.61 //m
//calculations
E=E*10^3
L1=L1*10^3
k=b*ko
Ix=b*h^3/12
Beta=(k/(4*E*Ix))^(1/4)
ba=2.00 // ba = Beta*a  based on the discussion
//D_bz=exp(-Beta*z)*sin(Beta*z)
D_ba=exp(-ba)*cos(ba)
y_max=w/k*(1-D_ba)
ba=0.777 //Beta*a
bb=4.777 //Beta*b
B_ba=exp(-ba)*sin(ba)
B_bb=exp(-bb)*sin(bb)
M_max=abs(-w*(B_ba-B_bb)/(4*Beta^2))
c=h/2
S_max=M_max*c/Ix
// calculation of M_H
ba=%pi/4 //Beta*a
bb=4-%pi/4 //Beta*b
B_ba=exp(-ba)*sin(ba)
B_bb=exp(-bb)*sin(bb)
M_H=w/(4*Beta^2)*(B_ba+B_bb)
printf('y_max = %.3f mm',y_max)
printf('\n M_max = %.3f kN.m',M_max/10^6)
printf('\n S_max = %.3f MPa',S_max)
printf('\n M_H = %.3f kN.m',M_H/10^6)
```

Scilab code Exa 10.5 Example5

```
clc
// initialization of variables
clear
E=200 //GPa
h=102 //mm
b=68 //mm
Ix=2.53e+06 //mm^4
L1=4 //m
ko=0.35 //N/mm^3
P=30.0 //kN
//calculations
E=E*10^3
P=P*10^3
L1=L1*10^3
k=b*ko
Beta=(k/(4*E*Ix))^(1/4)
if(L1>3*%pi/(2*Beta))
    y_max=2*P*Beta/k
   M_max=-0.3224*P/Beta
   S_max=abs(M_max*h/(2*Ix))
end
z=%pi/(4*Beta)
printf('y_max = %.2f mm',y_max)
printf('\n M_max = %.2f kN.m',M_max/10^6)
printf('\n S_max = %.1f MPa',S_max)
printf('\n Location of Sigma_max is z = %d mm',z)
```

Scilab code Exa 10.6 Example6

```
clc
// initialization of variables
clear
P=30.0 //kN
a=500 //mm
h=102 //mm
```

```
b=68 //mm
k=23.8 //N/mm^2
Beta=0.001852
Ix=2.53e+06 //mm^4
//calculations
P=P*10^3
C_ba=exp(-Beta*a)*(-sin(Beta*a)+cos(Beta*a))
D_ba=exp(-Beta*a)*cos(Beta*a)
// y = P*Beta/(2*k)*(A_bz+2*D_ba*D_baz+C_ba*C_baz))
// Mx = P/(4*Beta)*(C_bz-2*D_ba*B_baz-C_ba*A_baz)
A_ba=exp(-Beta*a)*(sin(Beta*a)+cos(Beta*a))
B_ba=exp(-Beta*a)*sin(Beta*a)
C_ba=exp(-Beta*a)*(-sin(Beta*a)+cos(Beta*a))
D_ba=exp(-Beta*a)*cos(Beta*a)
z1=424 //mm
z=z1-a
A_bz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
B_bz=exp(-Beta*z)*sin(Beta*z)
C_bz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
D_bz=exp(-Beta*z)*cos(Beta*z)
// to find out X_baz
z=a+z
A_baz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
B_baz=exp(-Beta*z)*sin(Beta*z)
C_baz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
D_baz=exp(-Beta*z)*cos(Beta*z)
y_max = P*Beta/(2*k)*(A_bz+2*D_ba*D_baz+C_ba*C_baz)
printf('y_max = %.4f mm',y_max)
// For M_max
z1=500 //mm
z=z1-a
A_bz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
B_bz=exp(-Beta*z)*sin(Beta*z)
C_bz=exp(-Beta*z)*(-sin(Beta*z)+cos(Beta*z))
D_bz=exp(-Beta*z)*cos(Beta*z)
// to find out X_baz
z=a+z
A_baz=exp(-Beta*z)*(sin(Beta*z)+cos(Beta*z))
```

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

Scilab code Exa 10.7 Example7

```
clc
// initialization of variables
clear
D=30 //m
t=10 //m
h=20 //mm
E=200 //GPa
v=0.29
rho=900 //kg/m^3
//calculations
//part (a)
E=E*10^3
a=D/2*10^3
p=t*10^3*9.807*rho*10^-9
S_th=p*a/h
tau_max=S_th/2
printf('part (a)')
printf('\n Maximum shear stress= %.2f MPa',tau_max)
// part (b)
k=E*h/(a^2)
Beta=(3*(1-v^2)/(h^2*a^2))^(1/4)
L1=3*%pi/(4*Beta) //L1=L/2
u=S_th*a/E
w=2*k*u/(Beta)
M_max=w/(4*Beta)
```

```
Szz_max=M_max*(h/2)/(h^3/12)
Sth_max=v*Szz_max
tau_max=Szz_max/2
u_b=w*(1-v)*a/(2*E*h)
printf('\n part (b)')
printf('\n Maximum shear stress= %.2f MPa',tau_max)
printf('\n u_bottom = %.3f mm',u_b)
// part (c)
w=u*k/(2*Beta)
z=%pi/(4*Beta)
B_bz=exp(-Beta*z)*sin(Beta*z)
M_max=-w*B_bz/Beta
c=6
I=h^2
Szz_max=(M_max*c/I)
S_th1=v*(Szz_max)
k=0.3224
S_th2=(1-k)*S_th
Sigma_th=S_th1+S_th2
tau_max=(Sigma_th-Szz_max)/2
printf('\n part (c)')
printf('\n Maximum shear stress= %.2f MPa',tau_max)
```

Chapter 11

THE THICK WALL CYLLINDER

Scilab code Exa 11.1 Example1

```
clc
// initialization of variables
clear
E=200 //GPa
v=0.29
Di=20 //mm
Do=100 //mm
a=10 //mm
b=50 //mm
p1=300 //MPa
//calculations
// S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
// S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
r=10
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
printf('r = %d mm',r)
printf('\n Radial stress = %.1f MPa',S_rr)
printf('\n circumferential stress = %.1f MPa',S_th)
```

```
r=25
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
printf('\n r = %d mm',r)
printf('\n Radial stress = %.1f MPa',S_rr)
printf('\n circumferential stress = %.1f MPa',S_th)
r=50
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
printf('\n r = %d mm',r)
printf('\n Radial stress = %.1f MPa',S_rr)
printf('\n circumferential stress = %.1f MPa',S_th)
```

Scilab code Exa 11.2 Example2

```
clc
// initialization of variables
clear
E=72 //GPa
v=0.33
Di=200 //mm
Do=800 //mm
a=100 //mm
r=a
b=Do/2 //mm
p1=150 //MPa
E=E*10^3
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
S_zz=p1*a^2/(b^2-a^2)
tau_max=(S_th-S_rr)/2
u_a=p1*a/(E*(b^2-a^2))*((1-2*v)*a^2+(1+v)*b^2)
printf('Radial stress = %.1f MPa',S_rr)
printf('\n circumferential stress = %.1f MPa',S_th)
printf('\n Normal stress = %d MPa',S_zz)
```

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

Scilab code Exa 11.3 Example3

```
clc
// initialization of variables
clear
E=200 //GPa
a=10 //mm
v=0.29
ci=25.072 //mm
co=25 //mm
b=50 //mm
rr=0.072 //mm
re=0.025 //mm
alpha=0.0000117 // per celcius
//calculations
E=E*10^3
p1=300 //MPa
term1=co/(E*(b^2-co^2))*((1-v)*co^2+(1+v)*b^2)
term2=-ci/(E*(ci^2-a^2))*((-(1-v)*ci^2)-(1+v)*a^2)
ps=rr/(term1+term2)

// Inner cylinder p1=0 p2=ps a=10 b=25
// outer cylinder p1=ps p2=0 a=25 b=50
// S_rr=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
// S_th=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b
    ^2-a^2))*(p1-p2)
// results
// residual stresses for inner cylinder
p1=0
p2=ps
r=10
```

```
a=10
b=25
S_rri1=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
S_thi1=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
printf('\n Inner cylinder')
printf('\n r = %d mm',r)
printf('\n S_rr|R = %d MPa,    S_th|R = %.1f MPa',S_rri1,S_thi1)
r=25
S_rri2=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
S_thi2=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
printf('\n r = %d mm',r)
printf('\n S_rr|R = %.1f MPa,    S_th|R = %.1f MPa',S_rri2,S_thi2)
// residual stresses for outer cylinder
p1=ps
p2=0
a=25
b=50
r=25
S_rro1=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
S_tho1=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
printf('\n')
printf('\n Outer cylinder')
printf('\n r = %d mm',r)
printf('\n S_rr|R = %d MPa,    S_th|R = %.1f MPa',S_rro1,S_tho1)
r=50
S_rro2=(p1*a^2-p2*b^2)/(b^2-a^2)-(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
S_tho2=(p1*a^2-p2*b^2)/(b^2-a^2)+(a^2*b^2)/(r^2*(b^2-a^2))*(p1-p2)
```

```
printf('\n r = %d mm',r)
printf('\n S_rr|R = %.1f MPa,    S_th|R = %.1f MPa',S_rro2,S_tho2)
// AN internal pressure of 300 MPa
a=10 //mm
b=50 //mm
p1=300 //MPa
r=10
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
S_rr1=S_rr+S_rri1
S_th1=S_th+S_thi1
printf('\n')
printf('\n Inner cylinder')
printf('\n r = %d mm',r)
printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr1,S_th1)
r=25
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
S_rr2=S_rr+S_rri2
S_th2=S_th+S_thi2
printf('\n r = %d mm',r)
printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr2,S_th2)
// Outer Cyllinder
S_rr1=S_rr+S_rro1
S_th1=S_th+S_tho1
printf('\n')
printf('\n Outer cylinder')
printf('\n r = %d mm',r)
printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr1,S_th1)
r=50
S_rr=p1*(a^2*(r^2-b^2))/(r^2*(b^2-a^2))
S_th=p1*(a^2*(r^2+b^2))/(r^2*(b^2-a^2))
S_rr2=S_rr+S_rro2
S_th2=S_th+S_tho2
```

```
printf('\n r = %d mm',r)
printf('\n S_rr = %.1f MPa,    S_th = %.1f MPa',S_rr2
    ,S_th2)
//delT=u/(r*alpha)
u=rr+re
r=25
delT=u/(r*alpha)
printf('\n delT = %.1f degree Celcius',delT)
```

Scilab code Exa 11.4 Example4

```
clc
// initialization of variables
clear
SF=1.75
p1=300 //MPa
S_rr=-SF*p1
S_th=SF*325
Y=1/sqrt(2)*sqrt((S_th-S_rr)^2+S_rr^2+S_th^2)
printf(' Y = %.1f MPa',Y)
```

Scilab code Exa 11.5 Example5

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

```
S_rre=-189.1
S_th1=62.5
S_the=315.1
// ABove values are obtained from running the codes
    Ex11_1 and Ex11_3.sce
S_rr=-SF*S_rr1+S_rre
S_th=SF*S_th1+S_the
Y2=1/sqrt(2)*sqrt((S_th-S_rr)^2+S_rr^2+S_th^2)
if(Y2>Y1)
    Y=Y2
end
printf(' Y = %.1f MPa',Y)
```

Scilab code Exa 11.6 Example6

```
clc
// initialization of variables
clear
SF=1.8
a=20 //mm
b=40 //mm
Y=450 //MPa
//part (a)
tau_Y=Y/sqrt(3)
Pp=2*tau_Y*log(b/a)
S_th=2*tau_Y*(1-log(b/a))
S_rr=-Pp
S_zz=(S_th+S_rr)/2
printf('part (a)')
printf('\n S_th = %.1f MPa',S_th)
printf('\n S_zz = %.1f MPa',S_zz)
// part (b)
S_thR=S_th-Pp*(b^2+a^2)/(b^2-a^2)
S_zzR=S_zz-Pp*(a^2)/(b^2-a^2)
S_thR=S_thR/2
```

```
S_zzR=S_zzR/2
printf('\n part (b)')
printf('\n S_th|R = %.1f MPa',S_thR)
printf('\n S_zz|R = %.1f MPa',S_zzR)
// par (c)
// We need to find out p1. To do that let it be unity
p1=1
S_thR=-S_thR
S_zzR=-S_zzR
S_rr=-SF*p1
S_th=SF*p1*(b^2+a^2)/(b^2-a^2)
S_zz=SF*p1*a^2/(b^2-a^2)
// 2Y^2=(s_th-S_rr)^2+(S_rr-S_zz)^2+(S_zz-S_th)^2
// S_th=S_th*p1-S_thR
// S_zz=S_zz*p1-S_zzR
// a*p1^2+b*p+c=0
a=(S_th+SF)^2+(-SF-S_zz)^2+(S_zz-S_th)^2
c=S_thR^2+S_zzR^2+(S_thR-S_zzR)^2
b=-2*(S_th+SF)*S_thR+2*S_zzR*(-SF-S_zz)+2*(S_zz-S_th)*(S_thR-S_zzR)
c=c-2*Y^2
p11=roots([a b c])
p12=roots([a 0 -2*Y^2])
p11=p11(1)
p12=p12(1)
printf('\n Internal working pressure is %.1f MPa,',p11)
printf('\n Without residual stresses %.1f MPa',p12)
```

Scilab code Exa 11.8 Example8

```
clc
// initialization of variables
clear
```

```
a=100 //mm
b=300 //mm
Y=620 //MPa
E=200 //GPa
S_zz=0
v=0.29
rho=7.85e+03 //kg/m^3
// part (a)
S_thmax=Y
Wy=sqrt(4*Y/(rho*((3+v)*b^2+(1-v)*a^2)))
printf('part (a)')
printf('\n Omega_y =%d rad/s',Wy*10^6)
// part (b)
Wp=sqrt(3*Y/(rho*(b^2+a*b+a^2)))
ratio=Wp/Wy
printf('\n Omega_p = %d rad/s',Wp*10^6)
printf('\n ratio = %.2f',ratio)
```

Scilab code Exa 11.9 Example9

```
clc
// initialization of variables
clear
a=100 //mm
b=300 //mm
v=0.29
a=a*10^-3
b=b*10^-3
printf('r        S_rr|R/Y        S_th|R/Y        (S_th/R-S_rr/R)/Y')
for i=1:21
    r=0.09+0.01*i
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)))
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))
printf('\n %.2f       %.5f        %.5f        %.5f',r,S_rrR,
   S_thR,S_rrR-S_thR)
end
```

Chapter 12

ELASTIC AND INELSTIC STABILITY OF COLUMNS

Scilab code Exa 12.3 Example3

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

Scilab code Exa 12.5 Example5

```
clc
// initialization of variables
clear
L=1 //m
b=40 //mm
h=75 //mm
SF=2.5
K=1
L=L*10^3
Iy=b*h^3/12
A=b*h
ry=sqrt(Iy/A)
K_y=K*L/ry
rz=b/sqrt(12)
K=0.5
K_z=K*L/rz
S_cr=229 //MPa
P_cr=S_cr*A
P=P_cr/SF
printf('P = %d kN',P/10^3)
```

Chapter 13

FLAT PLATES

Scilab code Exa 13.2 Example2

```
clc
// initialization of variables
clear
d=3.6 //m
w=2.7 //m
ha=3.0 //m
b=0.9 //m
a=1.2 //m
v=0.29
E=200 //GPa
p=ha*9.8
//part (a)
S_w=124 //MPa
b_a=b/a
M=0.04*p*b^2*10^3
h=sqrt(6*M/S_w)
printf('part (a)')
printf('\n h = %.2f mm',h)
// part (b)
C=0.032/(1+b_a^4)
p=p*10^3
```

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

Scilab code Exa 13.3 Example3

```
clc
// initialization of variables
clear
E=200 //GPa
v=0.29
Y=315 //MPa
h=10 //mm
D=200 //mm
SF=2.0
//part (a)
a=D/2
E=E*10^3
Py=1 // Since unknown
S_maxk=3/4*Py*a^2/h^2
Py=Y/S_maxk
w_max=3/16*(1-v^2)*Py*a^4/(E*h^3)
printf('Py = %.2f MPa',Py)
printf('\n W_max = %.3f mm',w_max)
// part (b)
Pw=Py/SF
printf('\n part (b)')
printf('\n Pw = %.2f MPa',Pw)
```

Scilab code Exa 13.4 Example4

```
clc
// initialization of variables
clear
D=500 //mm
h=5 //mm
Sigma=288 //MPa
E=72 //GPa
SF=2
//part (a)
a=D/2
E=E*10^3
f=Sigma*a^2/(E*h^2)
// w_max/h has to be 2.4 since f=10
Pr=50
p=Pr*E*h^4/a^4
p=p/2
printf('part (a)')
printf('\n Allowable pressure = %d kPa',p*10^3)
// part (b)
q=p*a^4/(E*h^4)
// Corresponding w_max/h = 1.8
w_max=1.8*h
printf('\n part (b)')
printf('\n W_max = %.2f mm',w_max)
```

Chapter 14

STRESS CONCENTRATIONS

Scilab code Exa 14.1 Example1

```
clc
// initialization of variables
clear
// part (a)
ab_r=100
Sigma_1=-20 //MPa
Sigma_2=-75 //MPa
alphao=0.01 //rad
//calculations
A=(Sigma_1+Sigma_2)/(Sigma_1-Sigma_2)
th=1/2*acos((A*sinh(2*alphao)-1/2*(sinh(2*alphao)+
    cosh(2*alphao)))/A)
printf('pat (a)')
printf('\n theta = %.4f rad',th)
//part (b)
S_bb=-(Sigma_1-Sigma_2)^2/(2*(Sigma_1+Sigma_2))*(1+
    cosh(2*alphao)/sinh(2*alphao))
printf('\n part (b)')
printf('\n Maximum tensile stress = %d MPa',S_bb)
//part (c)
Beta=exp(2*alphao)*cosh(2*alphao)-2*A^2*(sinh(2*
```

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

Scilab code Exa 14.2 Example2

```
clc
// initialization of variables
clear
S_u=420 //MPa
SF=4.00
D=110 //mm
d=50.0 //mm
w=20 //mm
rho=10.0 //mm
SF=4.0
//calculations
t=(D-d)/2
tr=t/rho
rd=rho/d
S_cs=1+2*sqrt(tr)
A=w*d
Pf=S_u*A/1.83
P=Pf/SF
printf('P = %.1f kN',P/10^3)
```

Scilab code Exa 14.3 Example3

```
clc
// initialization of variables
clear
//part(a)
```

```
H=200 //mm
h=100 //mm
rho=10 //mm
Sigma_u=250 //MPa
P=1.5 //kN
L=1.4 //m
b=40 //mm
P=P*10^3
L=L*10^3
Hr=H/h
rh=rho/h
S_cc=1.77
c=h/2
I=b*h^3/12
S_max=S_cc*P*L*c/I
printf('part (a)')
printf('\n Flexural design stress = %.1f MPa',S_max)
//part (b)
SF=Sigma_u*I/(S_cc*P*L*c)
printf('\n part (b)')
printf('\n SF =%.2f ',SF)
```

Chapter 15

FRACTURE MECHANICS

Scilab code Exa 15.2 Example2

```
clc
// initialization of variables
clear
d=250 //mm
c=30 //mm
t=25 //mm
// part (a)
a=5 //mm
lambda=a/(2*c)
f1l=1.22 //from the tble
f2l=1.02
//We don't know P yet so say P=1
P=1
Sfl=P/(t*2*c)*f1l+3*280*P*f2l/(2*t*c^2)
K_IC=59*sqrt(1000)
P=K_IC/(Sfl*sqrt(a*%pi))
printf('part (a)')
printf('\n P = %.1f kN',P/10^3)
// part (b)
a=10 //mm
lambda=a/(2*c)
```

```
f1l=1.33 //from the tble
f2l=1.05
// We don't know P yet so say P=1
P=1
Sfl=P/(t*2*c)*f1l+3*280*P*f2l/(2*t*c^2)
K_IC=59*sqrt(1000)
P=K_IC/(Sfl*sqrt(a*%pi))
printf('\n part (b)')
printf('\n P = %.1f kN',P/10^3)
```

Scilab code Exa 15.3 Example3

```
clc
// initialization of variables
clear
a=100/2 //mm
Y=1500 //MPa
t=6 //mm
w=800 //mmm
c=200 //mm
a_c=a/c
fl=1.045
w=w*10^-3
t=t*10^-3
a=a*10^-3
A=w*t
Sigma=1/A
K_I=Sigma*sqrt(%pi*a)*fl
printf('part (a)')
printf('\n K_I = %.2f MPa sqrt(m)',K_I)
```

Scilab code Exa 15.4 Example4

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

Scilab code Exa 15.5 Example5

```
clc
// initialization of variables
clear
// For 30 mm crack
a=30/2 // mm  crack
S_30 =600 //MPa
a=a*10^-3
C=S_30*sqrt(a)
// For 120 mm crack
a=120/2
a=a*10^-3
S_120=C/sqrt(a)
printf('Sigma_120 = %d MPa',S_120)
```

Chapter 16

FATIGUE PROGRESSIVE FRACTURE

Scilab code Exa 16.1 Example1

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

```
tau=M*c/J
S_max=315 //MPa
// P_max^2*(3*(tau/S_max)^2+(Sigmaa/S_max)^2)=1
P_max=sqrt(1/((3*(tau/S_max)^2)+(Sigma/S_max)^2))
P_min=-5/6*P_max
printf('part(a)')
printf('\n P_max = %d N',P_max)
printf('\n P_min = %d N',P_min)
```

Scilab code Exa 16.2 Example2

```
clc
// initialization of variables
clear
b=10 //mm
M=1
t=50 //mm
rho=5 //mm
h=25 //mm
c=60 //mm
SF=4.0
//part (a)
S_cc=2.8
q=0.94
S_ce=1+q*(S_cc-1)
// M is not known. take it as unity
S_n=3*M*t/(2*h*(c^3-t^3))
S_e=S_ce*S_n
printf('part (a)')
printf('\n Effective stress = %.1e M',S_e)
//part (b)
S_max=172 //MPa
S_w=S_max/SF
M=S_w/S_e
printf('\n part (b)')
```

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

Scilab code Exa 16.3 Example3

```
clc
// initialization of variables
clear
rho=0.75 //mm
S_n=32.97e-06 // M
S_cc=6.1
q=0.69
S_ce=1+q*(S_cc-1)
// M is not known. take it as unity
M=1
S_e=S_ce*S_n
printf('part (a)')
printf('\n Effective stress = %.1e M',S_e)
// part (b)
S_w=43 //MPa
M=S_w/S_e
printf('\n part (b)')
printf('\n M =%.1f N.m',M/10^3)
```

Scilab code Exa 16.4 Example4

```
clc
// initialization of variables
clear
E=72 //Gpa
v=0.33
S_u=470 //MPa
Y=330 //MPa
S_an=190 //MPa
```

```
N=1e+06 //cycles
T=10 //mm
D=59 //mm
d=50 //mm
t=3 //mm
rho=t
P_min=20 //kN
q=0.95
//calculations
P_min=P_min*10^3
S_cc=1.90
S_ce=1+q*(S_cc-1)
A=T*d
S_nMin=P_min/A
S_nam=S_an/S_ce
//(S_na/S_nam)+(S_nm/S_u)^2=1
// S_nm^2=S_nMin^2+S_na^2+2*S_na*S_nMin
c=S_nMin^2-S_u^2
a=1
b=2*S_nMin+S_u^2/S_nam
S_na=roots([a b c])
S_na=S_na(2)
// Solving gives S_na
S_nm=S_nMin+S_na
S_nMax=S_nMin+2*S_na
P_max=A*S_nMax
S_max=S_nm+S_ce*S_na
S_min=S_nm-S_ce*S_na
printf('P_max = %.1f kN',P_max/10^3)
printf('\n S_max = %.1f MPa',S_max)
printf('\n S_min = %.1f MPa',S_min)
```

Scilab code Exa 16.5 Example5

```
clc
```

```
// initialization of variables
clear
// Equation given: E_l =E_p + E_e
// E_p = 0.58*(2N)^-0.57
// E_e=0.0062*(2N)^-0.09
// Part (a)
function [f]=func(N)
    f = 0.58*(2*N)^(-0.57)+0.0062*(2*N)^(-0.09)-0.01;
endfunction

Nc=6390
N=Nc
E_p = 0.58*(2*N)^-0.57
E_e = 0.0062*(2*N)^-0.09
E_l=E_p+E_e
printf('Part (a)')
printf('\n Total strain = %.5f ',E_l)
//part (b)
N=1/2*10^6
E_p = 0.58*(2*N)^-0.57
E_e = 0.0062*(2*N)^-0.09
E_l=E_p+E_e
printf('\n Part (b)')
printf('\n Total strain = %.5f ',E_l)
// part (c)
E_l=0.01
// In order to solve for N We have to solve a non-linear equation

N = 1;//initial guess
f = 1;//initial guess
while(abs(f)>0.000001),
    f = func(N);
    if f>0 then
        N = N+1;
    elseif f<0 then
        N = N-1;
```

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

Chapter 17

CONTACT STRESSES

Scilab code Exa 17.1 Example1

```
clc
// initialization of variables
clear
E1=200 //GPa
E2=200 //Gpa
v1=0.29
v2=0.29
R1=60 //mm
R11=130 //mm
R2=80 //mm
R22=200 //mm
th=%pi/3
P=4.5 //kN
P=P*10^3
E=E1*10^3
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)
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)
```

```
Del=2*(1-v1^2)/(E*(A+B))
BAr=B/A
Cb=0.77
Cs=0.724
Ct=0.24
Cg=0.22
Cz=0.53
Cd=2.10
b=Cb*(P*Del)^(1/3)
br=b/Del
S_max=-Cs*br
tau_max=Ct*br
tau_oct=Cg*br
Zs=Cz*b
delta=Cd*P/%pi*((A+B)/br)
printf('Sigma_max = %d MPa',S_max)
printf('\n tau_max = %d MPa',tau_max)
printf('\n tau_oct_max = %d MPa',tau_oct)
printf('\n Zs = %.2f mm',Zs)
printf('\n delta = %.3f mm',delta)
// S_max doesn't match due to round off error
```

Scilab code Exa 17.2 Example2

```
clc
// initialization of variables
clear
E=200 //GPa
v=0.29
Y=1600 //MPa
Po=4.2 //kN
Omega=3000 //rpm
th=%pi/3
P=1.75 //kN
R1=4.76 //mm
```

```
R11=R1
R2=-4.86 //mm
R22=18.24 //mm
//part (a)
E=E*10^3
Po=Po*10^3
P=P*10^3
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)
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)
Del=2*(1-v^2)/(E*(A+B))
BAr=B/A
Cb=0.32
k=0.075
Cs=1.00
Ct=0.3
Cg=0.27
Cz=0.78
b=Cb*(P*Del)^(1/3)
a=b/k
br=b/Del
S_max=-Cs*br
tau_max=Ct*br
tau_oct=Cg*br
Zs=Cz*b
tauo=0.486*b/(2*Del)
Zo=0.41*b
printf('b = %.4f mm',b)
printf('\n a = %.3f mm',a)
printf('\n b/Delta = %d MPa',br)
printf('\n Sigma_max = %d MPa',S_max)
printf('\n tau_max = %d MPa',tau_max)
printf('\n tau_oct_max = %d MPa',tau_oct)
printf('\n Zs = %.3f mm',Zs)
printf('\n Tau_0 = %d MPa',tauo)
```

```
printf('\n Zo = %.3f mm',Zo)

// part (b)
tau_oY=sqrt(2)*Y/3
Py = 1/Del*(tau_oY/(Cg*Cb)*Del)^3
printf('\n part (b)')
printf('\n P_Y = %d N',Py)
SF=Py/P
printf('\n SF = %.2f ',SF)
```

Scilab code Exa 17.3 Example3

```
clc
// initialization of variables
clear
E=200 //GPa
v=0.29
R=40 //mm
h=20 //mm
P=24.1 //kN
S_max=1445 //MPa
tau_max=433 //MPa
tau_octM=361 //MPa
//calculations
E=E*10^3
P=P*10^3
Del=2*R*(1-v^2)/E
b=sqrt(2*P*Del/(h*%pi))
br=b/Del
S_maxT=2*b/(9*Del)
S_maxC=-1.13*br
tau_max=0.31*br
tau_octM=0.255*br
printf('Sigma_max (tension) = %d MPa',S_maxT)
printf('\n Sigma_max (compression) = %d MPa',S_maxC)
```

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