Scilab Textbook Companion for Solid Mechanics by S. M. A. Kazimi[1]

Created by
Maloth Thirupathi
Interest
Others
IIT BOMBAY
College Teacher
None
Cross-Checked by
K. V. P. Pradeep

June 21, 2014

[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: Solid Mechanics

Author: S. M. A. Kazimi

Publisher: Tata McGraw-Hill, New Delhi

Edition: 1

Year: 1976

ISBN: 0070964742

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

List of Figures

Chapter 2

ANALYSIS OF STRESS

Scilab code Exa 2.4 Chapter2 Example 4

```
clc
// initialization of variables
clear
tau=[200 100 0
     100 0 0
     0 0 500] // some units
theta=60 // degrees
//calculations
theta=theta*%pi/180
a=[cos(theta) sin(theta) 0
  -sin(theta) cos(theta) 0
   0 0 1]
b=a.'
tau_new=a*tau*b
// Results
printf('The new stress tensor is')
disp(tau_new)
```

Scilab code Exa 2.5 Chapter2 Example 5

```
clc
// initialization of variables
clear
sigma_1=100 //kg*f/cm^2
sigma_2=100 //kg*f/cm^2
sigma_3=-200 //kg*f/cm^2
// calculations
tau_oct=1/3*sqrt((sigma_1-sigma_2)^2+(sigma_2-
    sigma_3)^2+(sigma_3-sigma_1)^2)
// Results
printf('Octahedra shear stress at the point is=%.1f
    kgf/cm^2',tau_oct)
```

Scilab code Exa 2.7 Chapter2 Example 7

```
clc
// initialization of variables
clear
tau=[200 100 0
     100 0 0
     0 0 500] // some units
theta=60 // degrees
//calculations
theta=theta*%pi/180
a=[cos(theta) sin(theta) 0
  -sin(theta) cos(theta) 0
   0 0 1]
b=a.'
tau_new=a*tau*b

// stress invariants :old
I1=tau(1,1)+tau(2,2)+tau(3,3)
I2=tau(1,1)*tau(2,2)+tau(2,2)*tau(3,3)+tau(3,3)*tau
    (1,1)-(tau(1,2)^2+tau(2,3)^2+tau(3,1)^2)
I3=tau(1,1)*tau(2,2)*tau(3,3)+2*tau(1,2)*tau(2,3)*
```

```
tau(3,1)-(tau(1,1)*tau(2,3)^2+tau(2,2)*tau(3,1)
   ^2+tau(3,3)*tau(1,2)^2)

// stress invariants :new
I11=tau_new(1,1)+tau_new(2,2)+tau_new(3,3)
I22=tau_new(1,1)*tau_new(2,2)+tau_new(2,2)*tau_new
   (3,3)+tau_new(3,3)*tau_new(1,1)-(tau_new(1,2)^2+
   tau_new(2,3)^2+tau_new(3,1)^2)
I33=tau_new(1,1)*tau_new(2,2)*tau_new(3,3)+2*tau_new
   (1,2)*tau_new(2,3)*tau_new(3,1)-(tau_new(1,1)*
   tau_new(2,3)^2+tau_new(2,2)*tau_new(3,1)^2+
   tau_new(3,3)*tau_new(1,2)^2)

// Results
printf('The invariants of old stress tensor are I1=
   %0.2f I2=%.e I3=%.e \n and that of the new stress
    tensor are I1=%0.2f I2=%.e I3=%.e',I1,I2,I3,I11,
   I22,I33)
printf('\n Hence the same stress tensor invariants')
```

Scilab code Exa 2.8 Chapter2 Example 8

```
clc
// initialization of variables
clear
sigma_3=0 // kgf/cm^2
tau_oct=1500 // kgf/cm^2
n=2 // given that sigma_1=n*sigma_2
// calculations
sigma_2=1500*3/(sqrt(2*n^2-2*n+2)) // // kgf/cm^2
sigma_1=n*sigma_2 // kgf/cm^2
sigma_0=4500/sqrt(2) // kgf/cm^2
// Results
printf('The necessary stresses sigma_1, sigma_2 for
   biaxial yielding are \n %d kgf/cm^2, %d kgf/cm^2
```

```
and for uniaxial yielding sigma_0 %0.2f kgf/cm^2.
',sigma_1,sigma_2,sigma_0);
```

Scilab code Exa 2.9 Chapter2 Example 9

```
clc
// initialization of variables
clear
// part (a)
tau_xx=300 // kgf/cm^2
tau_yy=0 // kgf/cm^2
tau_xy=600 // kgf/cm^2
// calculations
sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))
   ^2+tau_xy^2)
sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))
   ^2+tau_xy^2)
Beta=atan(2*tau_xy/(tau_xx-tau_yy))
Beta=Beta*180/%pi
// Results
printf('Part (a) \n The magnitude of principal
   stresses are %d,%d kgf/cm^2 and \n the direction
   is given by 2*beta=%.2f degrees',sigma_1,sigma_2,
   Beta)

//part (b)
tau_xx=1000 // kgf/cm^2
tau_yy=150 // kgf/cm^2
tau_xy=450 // kgf/cm^2
// calculations
sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))
   ^2+tau_xy^2)
sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))
   ^2+tau_xy^2)
Beta=atan(2*tau_xy/(tau_xx-tau_yy))
```

```
Beta=Beta*180/%pi
// Results
printf('\n Part (b) \n The magnitude of principal stresses are %d,%d kgf/cm^2 and \n the direction is given by 2*beta=%.2f degrees',sigma_1,sigma_2,Beta)

// part (c)
tau_xx=-850 // kgf/cm^2
tau_yy=350 // kgf/cm^2
tau_xy=700 // kgf/cm^2
// calculations
sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))^2+tau_xy^2)
sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))^2+tau_xy^2)
Beta=atan(2*tau_xy/(tau_xx-tau_yy))
Beta=Beta*180/%pi
// Results
printf('\n Part (c) \n The magnitude of principal stresses are %d,%d kgf/cm^2 and \n the direction is given by 2*beta=%0.2f',sigma_1,sigma_2,-Beta)

// wrong answers were given in textbook for part (b)
```

Scilab code Exa 2.10 Chapter2 Example 10

```
clc
// initialization of variables
clear
tau_xx= -1 // kgf/cm^2
tau_yy= 0 // kgf/cm^2
tau_xy= 7 // kgf/cm^2
// calculations
sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))
```

```
^2+tau_xy^2)
sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))^2+tau_xy^2)
x=sigma_1 // positive one is tension
if(sigma_2>sigma_1)
    x=sigma_2
end
// Results
printf('The diagonal tension is %.3f kgf/cm^2',x)
```

Scilab code Exa 2.11 Chapter2 Example 11

```
clc
// initialization of variables
clear
d=2 // m
l=10 // m
t=1 // cm
p=15 // kgf/cm^2
pitch= 2*%pi //m
//calculations
w=2*%pi*d/2 // m
theta=atan(w/(2*%pi))
sigma_z=p*d*100/(4*t)
sigma_th=p*d*100/(2*t)
sigma_th_new=(sigma_th+sigma_z)/2+(sigma_th-sigma_z)/2*cos(2*theta)
tau_thz=(sigma_z-sigma_th)*sin(2*theta)/2
// results
printf('At the junction, the normal and shear stresses are %d and %d kgf/cm^2 \n respectively, and the rivets must be designed for this',sigma_th_new,-tau_thz)
```

Chapter 3

ANALYSIS OF STRAINS

Scilab code Exa 3.3 Chapter3 Example 3

```
clc
// initialization of variables
clear
epsillon=[0.01 0 0
     0 0.02 0.02
     0 0.02 0.01] // dimensionless
theta=30 // degrees
//calculations
theta=theta*%pi/180
a=[cos(theta) sin(theta) 0
  -sin(theta) cos(theta) 0
   0 0 1]
b=a.'
epsillon_new=a*epsillon*b
// calculation of strain invariants
// for epsillon
J1=epsillon(1,1)+epsillon(2,2)+epsillon(3,3)
J2=epsillon(1,1)*epsillon(2,2)+epsillon(2,2)*
   epsillon(3,3)+epsillon(3,3)*epsillon(1,1)-2*(
   epsillon(1,2)^2+epsillon(2,3)^2+epsillon(3,1)^2)
J3=epsillon(1,1)*epsillon(2,2)*epsillon(3,3)+2*
```

```
    epsillon(1,2)*epsillon(2,3)*epsillon(3,1)-(
    epsillon(1,1)*epsillon(2,3)^2+epsillon(2,2)*
    epsillon(3,1)^2+epsillon(3,3)*epsillon(1,2)^2)

// for epsillon_new
J11=epsillon_new(1,1)+epsillon_new(2,2)+epsillon_new
    (3,3)
J22=epsillon_new(1,1)*epsillon_new(2,2)+epsillon_new
    (2,2)*epsillon_new(3,3)+epsillon_new(3,3)*
    epsillon_new(1,1)-2*(epsillon_new(1,2)^2+
    epsillon_new(2,3)^2+epsillon_new(3,1)^2)
J33=epsillon_new(1,1)*epsillon_new(2,2)*epsillon_new
    (3,3)+2*epsillon_new(1,2)*epsillon_new(2,3)*
    epsillon_new(3,1)-(epsillon_new(1,1)*epsillon_new
    (2,3)^2+epsillon_new(2,2)*epsillon_new(3,1)^2+
    epsillon_new(3,3)*epsillon_new(1,2)^2)

// results
printf('The new strain tensor is');
disp(epsillon_new);
printf('The strain invariants of old strain tensor
    are J1=%0.2f J2=%.e J3=%.e \n and that of the new
     strain tensor are J1=%0.2f J2=%.e J3=%.e',J1,J2,
    J3,J11,J22,J33)
printf('\n Hence the same strain invariants')
```

Scilab code Exa 3.4 Chapter3 Example 4

```
clc
// initialization of variables
clear
epsillon=[0.01 -0.02 0
      -0.02 0.03 -0.01
      0 -0.01 0] // dimensionless
a_xx=0.6
```

```
theta=acos(a_xx) // radians
//calculations
// theta=theta*%pi/180
a=[cos(theta) 0 -sin(theta)
   0       1       0
   sin(theta) 0 cos(theta)]
b=a.'
epsillon_new=a*epsillon*b

// calculation of strain invariants
// for epsillon
J1=epsillon(1,1)+epsillon(2,2)+epsillon(3,3)

J2=epsillon(1,1)*epsillon(2,2)+epsillon(2,2)*
    epsillon(3,3)+epsillon(3,3)*epsillon(1,1)-2*(
    epsillon(1,2)^2+epsillon(2,3)^2+epsillon(3,1)^2)

J3=epsillon(1,1)*epsillon(2,2)*epsillon(3,3)+2*
    epsillon(1,2)*epsillon(2,3)*epsillon(3,1)-(
    epsillon(1,1)*epsillon(2,3)^2+epsillon(2,2)*
    epsillon(3,1)^2+epsillon(3,3)*epsillon(1,2)^2)

// for epsillon_new
J11=epsillon_new(1,1)+epsillon_new(2,2)+epsillon_new
    (3,3)

J22=epsillon_new(1,1)*epsillon_new(2,2)+epsillon_new
    (2,2)*epsillon_new(3,3)+epsillon_new(3,3)*
    epsillon_new(1,1)-2*(epsillon_new(1,2)^2+
    epsillon_new(2,3)^2+epsillon_new(3,1)^2)

J33=epsillon_new(1,1)*epsillon_new(2,2)*epsillon_new
    (3,3)+2*epsillon_new(1,2)*epsillon_new(2,3)*
    epsillon_new(3,1)-(epsillon_new(1,1)*epsillon_new
    (2,3)^2+epsillon_new(2,2)*epsillon_new(3,1)^2+
    epsillon_new(3,3)*epsillon_new(1,2)^2)

// Results
```

```
printf('The new strain tensor is')
disp(epsillon_new)
printf('The strain invariants of old stress tensor
   are J1=%0.2f J2=%.e J3=%.e \n and that of the new
    stress tensor are J1=%0.2f J2=%.e J3=%.e',J1,J2,
   J3,J11,J22,J33)
```

Scilab code Exa 3.5 Chapter3 Example 5

```
clc
// initialization of variables
clear
epsillon_A= 700*10^-6
epsillon_B= 300*10^-6
epsillon_C= 300*10^-6
theta=45 // degrees
theta=theta*%pi/180 // radians
// calculations
epsillon_x=epsillon_A
epsillon_y=epsillon_C
gamma_xy=(epsillon_B-(epsillon_x*cos(theta)^2+
   epsillon_y*sin(theta)^2))/(sin(theta)*cos(theta))
epsillon_1=1/2*(epsillon_x+epsillon_y)+(1/2)*sqrt((
   epsillon_x-epsillon_y)^2+gamma_xy^2)
epsillon_2=1/2*(epsillon_x+epsillon_y)-(1/2)*sqrt((
   epsillon_x-epsillon_y)^2+gamma_xy^2)
phi=0.5*atan(gamma_xy/(epsillon_x-epsillon_y))
phi=phi*180/%pi
//results
printf('The principal strains are %.3e, %.3e',
   epsillon_1,epsillon_2)
printf('\n phi = %.2f degrees',phi)
```

Scilab code Exa 3.6 Chapter3 Example 6

```
clc
// initialization of variables
clear
epsillon_A= 1000*10^-6
epsillon_B= 720*10^-6
epsillon_C= 600*10^-6
th_B=120 // degrees
th_C=240 // degrees
//calculations
th_B=th_B*%pi/180
th_C=th_C*%pi/180
// we need to solve for epsillon_y and gamma_xy
// Ax=B
ep_x=epsillon_A
A=[sin(th_B)^2 sin(th_B)*cos(th_B)
   sin(th_C)^2 sin(th_C)*cos(th_C)]
C=[epsillon_B-ep_x*cos(th_B)^2; epsillon_C-ep_x*cos(th_C)^2]
x=inv(A)*C
ep_y=x(1,1)
gam_xy=x(2,1)
epsillon_x=ep_x
epsillon_y=ep_y
gamma_xy=gam_xy
epsillon_1=1/2*(epsillon_x+epsillon_y)+(1/2)*sqrt((epsillon_x-epsillon_y)^2+gamma_xy^2)
epsillon_2=1/2*(epsillon_x+epsillon_y)-(1/2)*sqrt((epsillon_x-epsillon_y)^2+gamma_xy^2)
// Results
printf('The principal strains are %.3e, %.3e',epsillon_1,epsillon_2)
```

Chapter 4

STRESS STRAIN RELATIONS

Scilab code Exa 4.1 Chapter4 Example 1

```
clc
// initialization of variables
clear
E=2*10^6 // kg/cm^2
G=8*10^5 // kg/cm^2
ep=[0.001 0 -0.002
    0 -0.003 0.0005
    -0.002 0.0005 0]
// calculations
nu=E/(2*G)-1
lambda=E*nu/((1+nu)*(1-2*nu))
mu=G
sigma(1,1)=2*mu*ep(1,1)+lambda*(ep(1,1)+ep(2,2)+ep
   (3,3))
sigma(2,2)=2*mu*ep(2,2)+lambda*(ep(1,1)+ep(2,2)+ep
   (3,3))
sigma(3,3)=2*mu*ep(3,3)+lambda*(ep(1,1)+ep(2,2)+ep
   (3,3))
tau(1,2)=2*mu*ep(1,2)
```

```
tau(1,3)=2*mu*ep(1,3)
tau(2,3)=2*mu*ep(2,3)
tau=[sigma(1,1) tau(1,2) tau(1,3)
     tau(1,2) sigma(2,2) tau(2,3)
     tau(1,3) tau(2,3) sigma(3,3)]
// results
printf('The lames constants are %.e and %.e kg/cm^2',lambda,mu)
printf('\n The stres tensor is')
disp(tau)
```

Scilab code Exa 4.2 Chapter4 Example 2

```
clc
// initialization of variables
clear
sigma_x=1000 //kg/cm^2
sigma_y=-500 //kg/cm^2
sigma_z=0  //kg/cm^2
tau_xy=500 //kg/cm^2
E=2*10^6 // kg/cm^2
nu=0.25
//calculations
ep_x=1/E*(sigma_x-nu*(sigma_y+sigma_z))
ep_y=1/E*(sigma_y-nu*(sigma_x+sigma_z))
ep_z=1/E*(sigma_z-nu*(sigma_y+sigma_x))
J1=ep_x+ep_y+ep_z
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-sigma_y))^2+tau_xy^2)
th=1/2*atan(2*tau_xy/(sigma_x-sigma_y))
th=th*180/%pi
ep_1=1/E*(sigma_1-nu*sigma_2)
ep_2=1/E*(sigma_2-nu*sigma_1)
```

```
ep_3=-1/E*nu*(sigma_1+sigma_2)
//results
printf('The magnitude of principal strain are %.8f, %.8f, %.8f',abs(ep_1),abs(ep_2),abs(ep_3))
printf('\n and the diection is given by theta=%.2f degrees',th)
printf('\n J1 is %.6f',J1)
```

Scilab code Exa 4.3 Chapter4 Example 3

```
clc
// initialization of variables
clear
sigma_x=1400 //kg/cm^2
tau_xy=400// kg/cm^2
ep_z=-3.6*10^-6
nu=1/4
E=2*10^8 //kg/cm^2
// calculations
sigma_y=(-ep_z*E/nu)-sigma_x
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-sigma_y))^2+tau_xy^2)
th=0.5*atan(2*tau_xy/(sigma_x-sigma_y))
th=th*180/%pi
printf('sigma_y is %d kg/cm^2',sigma_y)
printf('\n The principal stresses are %d , %d kg/cm^2',sigma_1,sigma_2)
printf('\n The direction is given by theta = %.2f degrees',th)

// angle was given wrong in the text
```

Scilab code Exa 4.4 Chapter4 Example 4

```
clc
//initialization of variables
clear
C=1000/3 //kg/cm^2
sigma_x=2*C
sigma_y=4*C
tau_xy=4*C
sigma_0=4*C
sigma_1=3+C*sqrt(2)
sigma_2=3-C*sqrt(2)
sigma_3=0
tau_oct=1/3*sqrt((sigma_1-sigma_2)^2+(sigma_2-
   sigma_3)^2+(sigma_3-sigma_1)^2)
tau_max=sigma_1/2
taU=1.885*C
tau_y=2*C
printf('Actual tau is %.3f',taU)
printf('\n tau_max at yield is %.3f',tau_y)
printf('\n Hence yielding doesn not occur according
   to Von-Miles condition \n but it occurs due to
   Tresca condition')
```

Chapter 5

UNIAXIAL DEFORMATIONS

Scilab code Exa 5.1 Chapter5 Example 1

```
clc
//initialization of variables
clear
l=20 //cm
dL=1 //m
dl=0.004 //cm
//calculations
L=l*dL/dl //m
//results
printf('The depth of the clay bed is %d m',L)
```

Scilab code Exa 5.2 Chapter5 Example 2

```
clc
//initialization of variables
clear
A=1 //unit area
E=2*10^6 //kg/cm^2
```

```
// calculations
db=3000*90/(A*E)
dc=db+5000*60/(A*E)
dd=dc+4000*30/(A*E)
//results
printf('The extension of the rod in part AB is %.2e
    cm in part BC is %.2e cm \n and in part CD is %.2
    e cm',db,dc,dd)
```

Scilab code Exa 5.3 Chapter5 Example 3

```
clc
//initialization of variables
clear
A=3 //cm^2
L=18 //m
E= 2*10^6 //kg/cm^2
r=7833 //kg/m^3
//calculations
e=r*(L*100)^2/(2*E*10^6)
// results
printf('The elongation is %.5f cm',e)
```

Scilab code Exa 5.4 Chapter5 Example 4

```
clc
//initialization of variables
clear
// linked to 5_3
P=3 //tonne
E=2*10^6 //kg/cm^2
d_0= 1 //cm
d_1=2.8 //cm
```

```
// calculations
e=4*P*1000*d_l*10^3/(d_l^2*%pi*E*(1-((d_l-d_0)/d_l))
   )
//results
printf('The total elongation is %.2f cm',e)
```

Scilab code Exa 5.6 Chapter5 Example 6

```
clc
//initialization of variables
clear
P=10 //tonne
E=2*10^6 //kg/cm^2
// calculations
// We have to solve linear system Ax=B
A=[1 1 1 0
   3 1 -3 0
   -2 2 0 -E
   0 -1 2 -E]
B=[P*10^3;0;0;0]
x=inv(A)*B
W1=x(1,1)/1000
W2=x(2,1)/1000
W3=x(3,1)/1000
th=x(4,1)
//results
printf('The load taken by each rod is %.2f tonne, %
   .1f tonne, %.3f tonne',W1,W2,W3)
printf('\n and the slope is theta = %.2e. radians',
   th)
```

Scilab code Exa 5.8 Chapter5 Example 8

```
clc
// initialization of variables
clear
b=30 // cm
h=30 //cm
n=6
A=36 //cm^2
ss_s=1500 //kg/cm^2
ss_c=60 //kg/cm^2
Er=15 // Elasticity ratio
// calculations
L=A*Er*ss_c+(b*h-A)*ss_c
// results
printf('The safe load is %d.kg',L)
```

Scilab code Exa 5.9 Chapter5 Example 9

```
clc
// initiaization of variables
clear
gs_b=10 //cm
gs_h=10 //cm
d_b=2 //cm
d_h=2 //cm
As= 1 //cm^2
s=10000 //kg/cm^2
// part (a)
Es=2*10^6 //kg/cm^2
Ec=2*10^5 //kg/cm^2
// calculations
e=s/Es
Ac=gs_b*gs_h-(d_b*d_h)
e_c=e*Es*As/(Ec*Ac+Es*As)
s_c=Ec*e_c
e_s=e-e_c
```

```
s_s=Es*e_s
// results
printf('part (a) \n The stress in steel and concrete
    are respectively %d , %.2e kg/cm^2',s_s,s_c)
// part(b)
P=8000 //kg
// calculations
e_c=(e*Es*As-P)/(Ec*Ac+Es*As)
e_s=e-e_c
s_c=Ec*e_c
s_s=Es*e_s
// results
printf('\n part (b) \n The stress in steel and
   concrete are respectively %.1f , %.2f kg/cm^2',
   s_s,s_c)
```

Scilab code Exa 5.10 Chapter5 Example 10

```
clc
// initialization
clear
d=10 //cm
id=9.99 //cm
t=3 //mm
E=1.0*10^6 //kg/cm^2
a=2.02*10^-5 // degree/celcius
// part(a)
Tr=10 //degree C
T=(d-id)/id*1/a
printf('part(a) \n The sleeve must be heated to %.1f
    degree C or more for this purpose',T+Tr)

//part(b)
s_th=a*T*E
p=s_th*t*2/(d*10)
```

```
printf('\n part(b) \n The pressure developed between
    the rod and sleeve is %d kg/cm^2',p)

// part(c)
f=0.2
o=10 // overlap: cm
A=%pi*d*o
F=f*p*A
printf('\n part (c) \n The axial force required is
   %d kg',F)

//part (d)
// linked to part c
T2=20 //degree C
a2=1.17*10^-5 //  /degree C
Ts=(a-a2)*(T2-Tr)*E
Ts=s_th-Ts
p2=p*Ts/s_th
F2=F*Ts/s_th
printf('\n part(d)\n The pressure developed between
   the rod and sleeve is %.1f kg/cm^2',p2)
printf('\n The axial force required is %d kg',F2)
//part(e)
T3=Tr+(s_th/((a-a2)*10^6))
printf('\n part(e) \n The temperature at which the
   sleeve comes off easily is %.1f C',T3)

// calculations in the text: rounding off errors
```

Scilab code Exa 5.11 Chapter5 Example 11

```
clc
//initialization of variables
clear
T1=37.8 // degre C
```

```
t=0.355 //mm
T2=93.3 // degree C
L=2 //cm
m=1
n=1.53
a=1.86*10^-5
//calculations
R=2*t*(3*(1+m)^2+(1+m*n)*(m^2+(m*n)^-1))
R=R/(6*a*(T2-T1)*(1+m^2))  // mm
R=R/10
def=L^2/(8*R)
// results
printf('The radius of curvature is %.1f cm',R)
printf('\n The deflection is %.6f cm',def)
```

Scilab code Exa 5.12 Chapter5 Example 12

```
clc
// initialization of variables
clear
L=5 //cm
D=1.8 //cm
l=2.5 //cm
d=1.5 //cm
F=1 //tonne
E=2.1*10^6 //kg/cm^2
// calculations
s1=F*1000*4/(D^2*%pi)
s2=F*1000*4/(d^2*%pi)
U1=1/2*s1^2/E
U1=U1*L*D^2*%pi/4
U2=1/2*s2^2/E
U2=U2*l*d^2*%pi/4
U=U1+U2
// results
```

```
printf('The energy stored in the bolt is %.3f kg-cm'
    ,U)
```

Scilab code Exa 5.13 Chapter5 Example 13

```
clc
// initialization of variables
clear
t=16 //mm
Pt=1500 //kg/cm^2
Ps=1025 //kg/cm^2
Pb=2360 //kg/cm^2

//part (a)
p=6 //cm
r=24 //mm
d=r/10+0.15
Ft=t*(p-d)*Pt/10
Fs=%pi*d^2*Ps/4
Fb=d*t*Pb
x=min(Ft,Fs,Fb)
effA=x*100/(p*t/10*Pt)

//part (b)
p=9 //cm
r=30 //mm
d=r/10+0.2
Ft=t*(p-d)*Pt/10
Fs=%pi*d^2*Ps/4
Fb=d*t*Pb
x=min(Ft,Fs,Fb)
effB=x*100/(p*t/10*Pt)

// results
printf('The efficiencies corresponding to cases a
```

```
    and b are %.1f, %.1f',effA,effB)
printf('\n Hence part b is better than part a')
```

Chapter 6

TORSION INCLUDING NON CIRCULAR SECTIONS

Scilab code Exa 6.3 Chapter6 Example 3

```
clc
// initialization o variables
clear
p=5 //cm
D=10 //cm
d=2 //mm
T= 10 //kgm
ss= 785 //kg/cm^2
// calculations
P= 2*T/(%pi*D^2)
P=P*5*100
// results
printf('Force per rivet is %.1f kg',P)
printf('\n The diameter of rivet , using a
   permissible stress of %d kg/cm^2 = 0.227 cm',ss)
```

Scilab code Exa 6.4 Chapter6 Example 4

```
clc
// initialization of variables
clear
D=5 //cm
Y=3500 //kg/cm^2

//part (a)
Ta=350 //kg-m
tau=Y/2
Ip=Ta*D*100/(2*tau)
d1=Ip*32/%pi
d1=(D^4-d1)^(1/4)

//part (b)
Tb= 700 //kg-m
Ip=Tb*D*100/(2*tau)
d2=Ip*32/%pi
d2=(D^4-d2)
T=tau*%pi*(D^4)*2/(32*D)
// results
printf('The maximum diameter corresponding to the
   case a is %.2f cm',d1)
printf('\n Since the daimeter for the case (b) is
   coming out to be negative, \n The maximum torque
   transmitted is %.d kg-m',T/100)
```

Scilab code Exa 6.5 Chapter6 Example 5

```
clc
// initialization of variables
clear
A=3 //cm^2
E= 2*10^6 //kg/cm^2
```

```
nu= 0.25
l= 60 //m
L=150 //cm
d=0.5 //cm
dd=10 //cm
D=180 //cm
//calculations
K=(l*100/(A*E))+(L*D/2*D*32*2*(1+nu)/(E*%pi*dd^4*2))
P=d/K
// results
printf('The weight of the students that entered the
    length is %d kg',P)
```

Scilab code Exa 6.6 Chapter6 Example 6

```
clc
// initialization of variables
clear
// linked to 6_5
A=3 //cm^2
E= 2*10^6 //kg/cm^2
nu= 0.25
l= 60 //m
L=150 //cm
d=0.5 //cm
dd=10 //cm
D=180 //cm
//calculations
K=(l*100/(A*E))+(L*D/2*D*32*2*(1+nu)/(E*%pi*dd^4*2))
P=d/K
Ts=P/A
fs=dd*D*P*32/(%pi*4*dd^4)

// results
printf('The tensile stress is %.1f kg/cm^2',Ts)
```

```
printf('\n Maximum shear stress is %.1f kg/cm^2',fs)
```

Scilab code Exa 6.7 Chapter6 Example 7

```
clc
// initialization of variables
clear
F=500 //kg
k=25 //kg/cm
dd=15 //cm
ss=3500 //kg/cm^2
L=2 //m
G=8*10^5 //kg/cm^2
// calculations
x=sqrt(%pi*G/(25*L*32*100))
d=x*16*(F+dd*k)/(ss*%pi)
x2=x*d^2
// results
printf('d=%.2f cm',d)
printf('\n x=%.2f cm',x2)

// Text: not exact
```

Scilab code Exa 6.11 Chapter6 Example 11

```
clc
//initialization of variables
clear
d=5 //cm
rpm1=300 //rpm
rpm2=30000 //rpm
s=1000 //kg/cm^2
//calcuations
```

```
T=(d/2)*%pi*10^2*s/32
hp1= 2*%pi*rpm1*T/4500
hp2=hp1*100
// results
printf('Horse power at 300 rpm and 30000 rpm are
    respecively %d, %d h.p.',hp1/10,hp2/10)

// wrong/approximate answers in the text
```

Scilab code Exa 6.12 Chapter6 Example 12

```
clc
// initialization of variables
clear
hp=300 //h.p.
N1=30 //rpm
N2=30000 //rpm
fs=600 //kg/cm^2
// calculations
T1=4500*hp*100/(2*%pi*N1)
T2=T1/1000
D1=16*T1/(%pi*fs)
D1=D1^(1/3)
D2=16*T2/(%pi*fs)
D2=D2^(1/3)
// results
printf('Diameters required are %.1f, %.1f cm',D1,D2)

// wrong calculations in the text
```

Scilab code Exa 6.13 Chapter6 Example 13

```
clc
```

```
// initialization of variables
clear
d=10 //cm
t =1 //mm
T= 100 //kg-m
L=5 //m
G=8*10^5 //kg/cm^2

//calculations
r=d/2
fs=T*r*100/(r^2*2*%pi*L*t*10^-1)
U=fs^2/(2*G)
U1=U*(%pi*L*100)
// results
printf('Energy per unit volume = %.3f kg-cm/cm^3',U)
printf('\n Total strain energy= %d kg-cm',U1)
```

Scilab code Exa 6.14 Chapter6 Example 14

```
clc
//initialization of variables
clear
D=10 //cm
d= 1 //cm
n=20
P=60 //kg
G=8*10^5 //kg/cm^2
//calculations
n=n-0.75*2
delta=P*n*%pi*D^3*32/(4*%pi*G)
// results
printf('The deflection is %.1f cm',delta)
```

Scilab code Exa 6.15 Chapter6 Example 15

```
clc
//initialization of variables
clear
// linked to 6_14
D=10 //cm
d= 1 //cm
n=20
P=60 //kg
G=8*10^5 //kg/cm^2
// calculations
m=D/d
fs=8*P*D/(d^3*%pi)
fs1=fs*(1+0.615/m+3/(4*m-4))
// results
printf('The shear stress with and without correction
    facor are \n respectively %d, %d kg/cm^2',fs,fs1
  )
```

Scilab code Exa 6.16 Chapter6 Example 16

```
clc
//initialization of variables
clear

// circle
D=1 //unit diameter
Ip=D^4/32
Zp=D^3/16

//Square
s=sqrt(%pi/4)*D
Is=0.886*D^4/32
Zs=0.7383*D^3/16
```

```
//Rectangle
a=sqrt(%pi/2)*D
b=sqrt(%pi/8)*D
Ir=0.719*D^4/32
Zr=0.616*D^3/16

// Trianle
t=sqrt(%pi/sqrt(3))*D
It=0.725*D^4/32
Zt=0.622*D^3/16

//Ellipse
A=D/sqrt(2)
B=D/sqrt(8)
Ie=A^3*B^3/(A^2+B^2)
Ze=A*B^2/2

//Normalization
Is=Is/Ip
Ie=Ie/Ip
It=It/Ip
Ir=Ir/Ip

Zs=Zs/Zp
Ze=Ze/Zp
Zt=Zt/Zp
Zr=Zr/Zp
Ip=1
Zp=1
//results
printf('Z:: Circle:Square:Ellipse:Triangle:Rectangle = %.3f : %.3f : %.3f : %.3f : %.3f',Zp,Zs,Ze,Zt,Zr)
printf('\n I:: Circle:Square:Ellipse:Triangle:Rectangle = %.3f : %.3f : %.3f : %.3f : %.3f',Ip,Is,Ie,It,Ir)
```

Scilab code Exa 6.17 Chapter6 Example 17

```
clc
//initialization of variables
clear

yp=2450 //kg/cm^2
d=0.4 //cm
ys=4200 //kg/cm^2
sa=1.6 //mm
sb=7 //mm
// calculations
sa=sa/10
sb=sb/10
T1=yp*%pi*d^3/16
T2=ys*0.303*sa^2*sb
// results
printf('The maximum torque that can be transitted by
    the screw-driver is %.1f kg-cm',T2)
```

Scilab code Exa 6.18 Chapter6 Example 18

```
clc
//initialization of variables
clear
b=5 //cm
h=10 //cm
tL=3 //mm
tl=1.5 //mm
T=100 //kg-cm
// calculations
tl=tl/10
```

```
fs=T*100/(2*b*h*tl)
// results
printf('The maximum stress is %.1f kg/cm^2',fs)
```

Scilab code Exa 6.19 Chapter6 Example 19

```
clc
//initialization of variables
clear
b=5 //cm
h=10 //cm
tL=3 //mm
tl=1.5 //mm
T=100 //kg-cm
// calculations
D=2*(b+h)/%pi
AR=b*h
AC=%pi*D^2/4
r=AC/AR
// results
printf('The ratio is 1:%.2f',r)
```

Scilab code Exa 6.20 Chapter6 Example 20

```
clc
//initialization of variables
clear
G=8*10^5 //kg/cm^2
//part (a)
T =20 //kg-m
t1=0.9 //cm
t2=0.5 //cm
b1=6.8 //cm
```

```
b2=14.2 //cm
I0=1/3*(2*b1*t1^3+b2*t2^3)
Zt=I0/max(t1,t2)
fs=T*100/Zt
Phi=T*100/(G*I0)
printf('part (a)')
printf('\n The maximum shear stress and twist rate are respectively \n %d kg/cm^2, %.2e radians/cm ',fs,Phi)

//part (b)
t1=1 //cm
t2=1 //cm
b1=10 //cm
b2=9 //cm
I0=1/3*(b1*t1^3+b2*t2^3)
Zt=I0/max(t1,t2)
fs=T*100/Zt
Phi=T*100/(G*I0)
printf('\n part (b)')
printf('\n The maximum shear stress and twist rate are respectively \n %d kg/cm^2, %.2e radians/cm ',fs,Phi)

//part (c)
t1=0.76 //cm
t2=0.48 //cm
b1=8 //cm
b2=14.04 //cm
I0=1/3*(2*b1*t1^3+b2*t2^3)
Zt=I0/max(t1,t2)
fs=T*100/Zt
Phi=T*100/(G*I0)
printf('\n part (c)')
printf('\n The maximum shear stress and twist rate are respectively \n %d kg/cm^2, %.2e radians/cm ',fs,Phi)

```

```
//part(d)
t=1 //cm
b=19 //cm
I0=1/3*t^3*b
Zt=I0/t
fs=T*100/Zt
Phi=T*100/(G*I0)
printf('\n part (d)')
printf('\n The maximum shear stress and twist rate
   are respectively \n %d kg/cm^2, %.2e radians/cm '
   ,fs,Phi)

// Twist rate: answers differ by a scale of 10.
   wrong answers in the text
```

Scilab code Exa 6.21 Chapter6 Example 21

```
clc
// initialization of variables
clear
D=5 //cm
d=2 //cm
t_y=3000 //kg/cm^2
// calculations
R=D/2
r=d/2
Tep=2*%pi*R^3*t_y/3-%pi*r^3*t_y/6
t_er=2*Tep/(%pi*R^3)
t_er1=t_er*r/R
prs=t_y-t_er1
nrs=t_er-t_y
// results
printf('Maximum +ve residual stress occurs at %d cm
   radius and is equal to \n %d kg/cm^2',r,prs)
printf('\n Maximum -ve residual stress occurs at %d
```

```
cm radius and is equal to \n %d kg/cm^2',R,-nrs)
```

Chapter 7

BEAMS AND BENDING

Scilab code Exa 7.2 Chapter7 Example 2

```
clc
//initialization of variables
clear
s=3 //m
n=60
p=50 //kg
// calculations
W=n*p
Rc=W*2/s
Rb=W-Rc
dx = 0.001;
x = 0:dx:s
n = s/dx +1;
for i = 1:n
    Sx(i) = -Rb + Rc*x(i)^2/6;
    Mx(i) = Rb*x(i) - Rc*x(i)^3 /18;
end
//Results
figure(1);plot(x,Sx);title("Shear force diagram");
   xlabel("X (in m)");ylabel("Shear force (in kg)");
figure(2);plot(x,Mx);title("Bending Moment diagram")
```

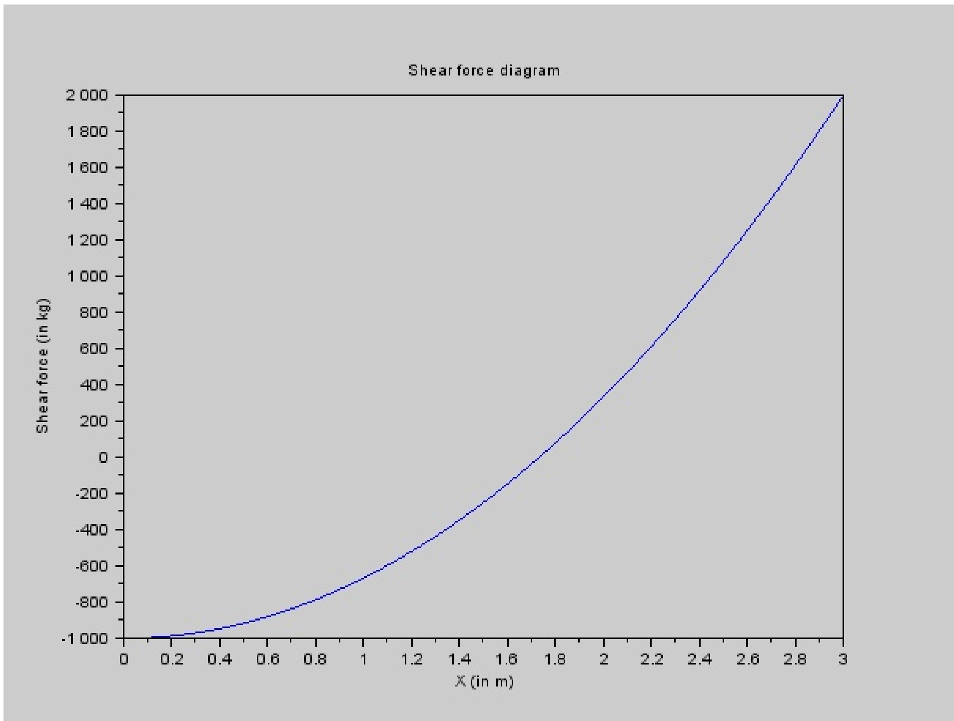

Figure 7.1: Chapter7 Example 2

```
;xlabel("X (in m)");ylabel("Bending Moment (in kg
-m)");
```

Scilab code Exa 7.8 Chapter7 Example 8

```
clc
//initialization of new variables
clear
b=10075 //mm
```

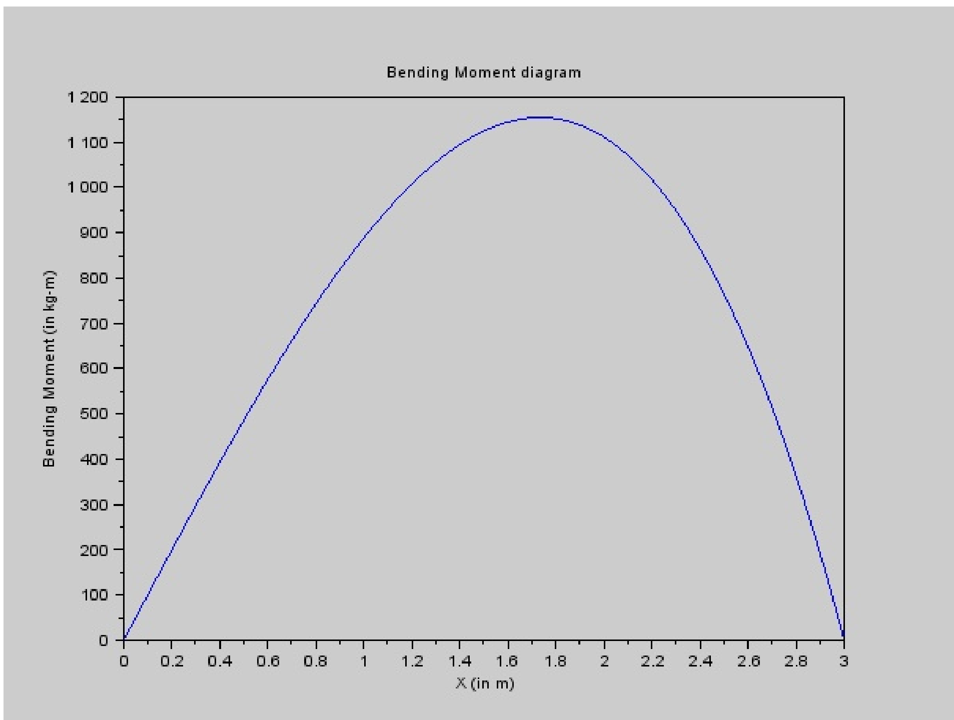

Figure 7.2: Chapter7 Example 2

```
h=10 //mm
p1=7.5
p2=9
//part (a)
ybar=1*p1*0.5+1*p2*5.5
ybar=ybar/16.5
xbar=1*p1*0.5+1*p1*4.75
xbar=xbar/16.5
printf('part (a)')
printf('\n Centroid coordinates (x,y) = (%.2f, %.2f) cm',xbar,ybar)

//part (b)
Ixx=p1*1^3/12+p1*1*(3.23-0.5)^2+1*p2^3/12+p2*1*(5.5-3.23)^2
Iyy=1*p1^3/12+p1*1*(3.75-2.43)^2+p2*1^3/12+p2*1*(2.43-0.5)^2
Ixy=p1*1.32*2.73+9*(-1.93)*(-2.27)
printf('\n part (b)')
printf('\n Moment of Areas: \n Ixx = %.1f cm^4 \n Iyy = %.1f cm^4 \n Ixy=%.1f cm^4',Ixx,Iyy,Ixy)

//part (c)
alpha=0.5*atan(2*Ixy/(Iyy-Ixx))
alpha=alpha*180/%pi
printf('\n part (c)')
printf('\n Direction of principal axes:')
printf('\n alpha = %.2f degrees',alpha)

//part (d)
Iuu=(Ixx+Iyy)/2+sqrt((Iyy-Ixx)^2/4+Ixy^2)
Ivv=(Ixx+Iyy)/2-sqrt((Iyy-Ixx)^2/4+Ixy^2)
printf('\n part (d)')
printf('\n Iuu = %.2f cm^4 \n Ivv = %.2f cm^4',Iuu,Ivv)
```

Scilab code Exa 7.10 Chapter7 Example 10

```
clc
//initialization of variables
clear
Ys=17000 //kg/cm^2
E=2*10^6 //kg/cm^2
d1=1 //mm
d=1 //cm
//calculations: 1 cm
R=E*d/(2*Ys)
M=Ys*%pi*d^3/32
// results
printf('%d cm daimeter wire:',d)
printf('\n Minimum radius = %.2f cm',R)
printf('\n Bending Moment = %.2f kg-cm',M)
// calculations: 1 mm
R1=R/(d1*10)
M1=M/(d1*1000)
// results
printf('\n %d mm daimeter wire:',d1)
printf('\n Minimum radius = %.2f cm',R1)
printf('\n Bending Moment = %.2f kg-cm',M1)
```

Scilab code Exa 7.11 Chapter7 Example 11

```
clc
//initialization of variables
clear
t=0.5 //cm
s=2 //m
p1=7.5 //cm
```

```
p2=10 //cm
d=p2/2
str=1650 //kg/cm^2
// calculations
// ab
IxX=p1*t^3/12+t*p1*d^2
// bc
alpha=atan(3/4)
Ixx=t*(p1+d)^3/12
Iyy=0
Ixy=0
Iuu=Ixx*cos(alpha)^2+Iyy*sin(alpha)^2-Ixy*sin(2*
   alpha)
Ixx=Iuu+IxX
IXX=Ixx*100/(2*p1)
Z=IXX/(d+t/2)
w=str*Z*8/(s^2*100)
w=w/1000
//Results
printf('w = %.1f tonne/m',w)
```

Scilab code Exa 7.12 Chapter7 Example 12

```
clc
//initialization of variables
clear
wb=10 //cm
wh=20 //cm
sb=0.5 //cm
sh=10 //cm
s=6 //m
fs=1650 //kg/cm^2
fw=150 //kg/cm^2
Es=2*10^6 //kg/cm^2
Ew=12*10^4 //kg/cm^2
```

```
//calculations
// Method 1
as=2*fs/(21*Es)
aw=2*fw/(20*Ew)
a=min(as,aw)
ss=a*Ew*wh/2
//Moment resistance of steel portion
F=(fs+1573)/2*sb*sh
k=sb/3*(fs+2*1573)/(fs+1573)
Ms=2*F*(10.5-k)
//Moment resistance of wooden portion
F=ss*wb*wb/2
Mw=2*(F*(wb-wb/3))
M=Ms+Mw
//Total udl supported
W=M*8/(s*100)

//Results
printf('Using method 1')
printf('\n W = %d kg',W)

//Method 2
nE=Es/Ew
nf=fs/fw
Is=2*(0+sb*sh*10.25^2)
Iw=0.6*wh^3/12
I=Is+Iw
W=fs*I*8/(s*100*10.5)

//Results
printf('\n Using method 2')
printf('\n W = %d kg',W)
```

Scilab code Exa 7.13 Chapter7 Example 13

```
clc
//initialization of variables
clear
p=6 //mm
Ixx=2375 //cm^4
Es=2*10^6 //kg/cm^2
EAl=0.667*10^6 //kg/cm^2
d1=10.6 //cm
d2=10 //cm
// calculations
I1=2*(0+p/10*10*10.3^2)
I2=Ixx*EAl/Es
I=I1+I2
n=I/I2
// results
printf('stiffness ratio = %.2f ',n)
n1=Es*d1/(d2*EAl)
printf('\n Stress ration = %.2f ',n1)
```

Scilab code Exa 7.14 Chapter7 Example 14

```
clc
//initilization of new variables
clear
wt=0.8 //cm
ft=1.4 //cm
w=10 //cm
y=20 //cm
// Sigma_y: yield stress is not given explicitly
k1=wt*(40-2*ft)/2
Zp=(14*19.3+k1*9.3)*2
If=2*(w*ft^3/12+w*ft*19.3^2)
Iw=wt*(40-2*ft)^3/12
I=Iw+If
Z=I/y
```

```
sf=Zp/Z
//Results
printf('shape factor = %.2f ',sf)
```

Scilab code Exa 7.15 Chapter7 Example 15

```
clc
//initilization of new variables
clear
wt=0.8 //cm
ft=1.4 //cm
w=10 //cm
y=20 //cm
T=750 //T==750*sigma_y
// calculations
MpF=ft*w*(40-2*ft)
c1=((40-2*ft)/2)^2-(T-MpF)/wt
c=sqrt(3*c1)
// results
printf('Elastic core of %.1f cm depth is present',2*
    c)
```

Scilab code Exa 7.17 Chapter7 Example 17

```
clc
//initialization of new variables
clear
P=2000 //kg
a=4 //cm
b=1 //cm
d=7 //cm
r=3 //cm
// calculations
```

```
A=(a+b)/2*d
xbar=(a+b*2)*d/(r*(a+b))
rbar=r+xbar
I=b*d^3/12+r*d^3/12
Ixx=I-A*2.8^2
e=Ixx/(rbar*A)
f1=P*5.8*(xbar-0.62)/(A*0.62*r)
f2=P*5.8*(-d+2.18)/(A*0.62*(5.18+d-2.18))
str=P/A
Str_i=f1+str
Str_o=-f2-str
//Results
printf('stress at the inner side of the hook = %.1f
   kg/cm^2 (tensile)',Str_i)
printf('\n stress at the outer side of the hook = %
   .1f kg/cm^2 (compressive)',Str_o)
// approximations involved in the text
```

Scilab code Exa 7.20 Chapter7 Example 20

```
clc
//initialization of new variables
clear
t=1 //cm
a=40 //cm
A=236
// calculations
ybar=a*t*0.5+(50-1)*4*0.5/(a*t+(50-1)*4)
y1bar=1.25*a-ybar
IAA=a*t^3/3+(50-1)^3*4/12+(50-1)*4*25.5^2
Io=IAA-A*ybar^2
//part (1)
r=y1bar/ybar
// results
printf('Ratio of maximum bending stress in the stem
```

```
    and flange')
printf('\n Ratio = %.2f',r)
//part(2)
// calculations
r=(2/3*388*29.56)-(2/3*160*20.44)-(228*20.44)
r=r/(2*2/3*388*29.56)
// results
printf('\n Ratio of S.F in flange to total S.F')
printf('\n Ratio = %.2f percent',r*100)
// part (3)
// calculations
r=359*200/Io
// results
printf('\n Ratio of maximum shear stress in the
    flange to average sher stress in the stem')
printf('\n Ratio = %.2f ',r)
//part (4)
// calculations
s=10 //m
r=r/0.922
sigma=1650 //kg/cm^2
shear=945 //kg/cm^2
wsh=2*200*shear/(r*s)
wsi=8*Io*sigma/(s^2*10*29.56)
w=min(wsh,wsi)
// results
printf('\n Maximum u.d.l. = %d kg/m ',w)

//wrong moment of Inertia (Io) in the text and hence
    part (3) and part (4) are wrong
```

Scilab code Exa 7.21 Chapter7 Example 21

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

```
clear
a=30 //cm
t=2.5 //cm
S=15 //cm
s=5 //Tonne
// calculations
I=a*a^3-25*25^3
I=I/12
tau_zx=s*1000*27.5*t*25/(4*35000*t)
FA=S*t*tau_zx
tau_xy=s*1000*a*t*27.5/(4*35000*t)
FB=tau_xy*t*S
//Results
printf('case A \n F = %d kg',FA)
printf('\n case B \n F= %d kg',FB)
```

Scilab code Exa 7.23 Chapter7 Example 23

```
clc
//initialization of variables
clear
h=40 //cm
b=10 //cm
t1=1.4 //cm
t2=0.8 //cm
Ixx=13989.5 //cm^4
//calculations
e=b^2*h^2*t1/(4*Ixx)*(1-t1/h-t1/b+t1^2/(b*h))*(1-t1/h)
//Results
printf('Shear center: \n e = %.2f cm',e)
```

Scilab code Exa 7.33 Chapter7 Example 33

```
clc
//initialization of new variables
clear
L=50 //cm
k=15 //cm
I=200 //cm^4
II=40 //cm^4
d=30 //cm
Pd=40 //cm
E=0.6667*10^6 //kg/cm^2
//calculations
delta=(100*10/2*16.33+L*d*35+L*k/2*25+d*k/2*45)
delta1=delta/E
//Results
printf('deflection = %.2f mm',delta1*10^1)
```

Chapter 8

STABILITY OF EQUILIBRIUM COLUMNS

Scilab code Exa 8.1 Chapter8 Example 1

```
clc
// initialization of variables
clear
L=5 //m
D=20 //cm
t=1 //cm
E=2*10^6 //kg/cm^2
I=2502 //cm^4
L=5*100 //cm
// calculations
P=E*I/(4*L^2)
// results
printf('The maximal axial load taken is %.1f Tonne',
   P/100)
printf('\n for both ends pinned, P=%.1f Tonne',P
   *4/100)
printf('\n for both ends fixed, P=%.1f Tonne',P
   *16/100)
printf('\n for one end fixed, one pinned, P=%.1f
```

```
    Tonne',P*4*2.13/100)

// Evaluation of critical load (P) in the text is
    wrong
```

Scilab code Exa 8.2 Chapter8 Example 2

```
clc
//initialization of variables
clear
E=2*10^6 //kg/cm^2
sigma_y=2600 //kg/cm^2
I=2502 //cm^4
L=500 //cm
A=59.7 //cm^2
L_tcr=L/sqrt(I/A)

printf('The actual critical length ratio is %.1f',
    L_tcr)
//case (b)
L_cr=sqrt(E*%pi^2/sigma_y)
printf('\n case (b)')
printf('\n The critical length ratio is %.1f',L_cr)

//case (a)
L_cr=sqrt(E*%pi^2/(4*sigma_y))
printf('\n case (a)')
printf('\n The critical length ratio is %.1f',L_cr)

//case (c)
L_cr=sqrt(4*E*%pi^2/sigma_y)
printf('\n case (c)')
printf('\n The critical length ratio is %.1f',L_cr)

// case (d)
```

```
L_cr=sqrt(2.05*E*%pi^2/sigma_y)
// Results
printf('\n case (d)')
printf('\n The critical length ratio is %.1f',L_cr)
printf('\n Only in case (a) actual ratio is more
    than critical ratio and material \n remains
    elastic  For cases (b), (c) and (d) critical
    length ratio is \n much higher and hence the
    material yelds before crippling loads are reached
    ')
```

Scilab code Exa 8.3 Chapter8 Example 3

```
clc
//initialzation of variables
clear
h=3.5 //m
A=22.4 //cm^2
r=7.08 //cm
E=2*10^6 //kg/cm^2
Q=1/2
// calculations
h=h*100
Q1=(Q*h/r)^2
s_cr=E*%pi^2/Q1
// results
printf('The critical stress is %d kg/cm^2',s_cr)
printf('\n This is much higher than yield stress for
     the material, \n so the column will fail by
    yielding')

// rounding off errors in the text
```

Scilab code Exa 8.4 Chapter8 Example 4

```
clc
//initialization of variables
clear
r_min=1.17 //cm
A=17.21 //cm^2
Q=1/2
h=3.5 //m
E=2*10^6 //kg/cm^2
h=h*100
// calculations
Q1=(Q*h/r_min)^2
s_cr=E*%pi^2/Q1
P_cr=s_cr*A
// results
printf('The crippling load is %d kg',P_cr)

// wrong calculations given in the text
```

Scilab code Exa 8.5 Chapter8 Example 5

```
clc
//initialization of variables
clear
L=2.5 //m
A=6.02 //cm^2
Q1=105
s=796.5 //kg/cm^2
// calculations
P=2*A*s
printf('The safe load is %d kg',P)
// Results
// wrong calculations in the text
```

Scilab code Exa 8.6 Chapter8 Example 6

```
clc
//initialization of variables
clear
h=3.5 //m
r_xx=7.08 //cm
A=24.38 //cm^2
Q=0.5
Q1=Q*h*100/r_xx

//Permissible load by secent formula
P=1231.28*2*A
printf('Permissible load by secent formula: %d kg',P
   )

//Permissible load by Rankine-Gordon formula
P=1260/(1+(24.75^2/18000))*2*A
printf('\n Permissible load by Rankine-Gordon
   formula: %d kg',P)

//Permissible load by parabolic formula
P=(1050-0.0233*Q1^2)*2*A
printf('\n Permissible load by parabolic formula: %d
    kg',P)

//Permissible load by straight-line formula
P=(1120-Q1*4.8)*2*A
printf('\n Permissible load by parabolic formula: %d
    kg',P)

// Rounding off errors in the text
```

Chapter 9

COMBINED STRESSES

Scilab code Exa 9.1 Chapter9 Example 1

```
clc
//initialization of variables
clear
//case (a)
A=72.9 //cm^2
Iy=633 //cm^4
Ix=1199 //cm^4
t=24/(5*Ix)+13.5/(5*Iy)
r=1/(A*t)
printf('case (a) \n r = %.3f cm',r)
// case (b)
t=24/(5*Ix)-13.5/(5*Iy)
r=1/(A*t)
printf('\n case (b) \n r = %.1f cm',r)
//case (c)
t=-24/(5*Ix)+13.5/(5*Iy)
r=1/(A*t)
printf('\n case (a) \n r = %.1f cm',r)
printf('\n So the load is to be placed on the leg OD
    , at a distance of %.1f cm from O',r )
```

Scilab code Exa 9.3 Chapter9 Example 3

```
clc
//initialization of variables
clear
b=14 //cm
d=20 //cm
rx=8.46 //cm
ry=2.99 //cm
// calculations
ex=2*rx^2/d
ey=2*ry^2/b
h=2*ex
w=2*ey
// results
printf('for steel height=%.3f cm and width=%.3f cm',
   h,w)
// ISHB 225
b=22.5 //cm
d=22.5 //cm
rx=9.8 //cm
ry=4.96 //cm
// calculations
ex=2*rx^2/d
ey=2*ry^2/b
h=2*ex
w=2*ey
// results
printf('\n for an ISHB height=%.3f cm and width=%.3f
    cm',h,w)
```

Scilab code Exa 9.4 Chapter9 Example 4

```
clc
//initialization of variables
clear
t=280 //kg/cm^2
c=840 //kg/cm^2
xbar=7.5 //cm from AB
A=210 //cm^2
// calculations
e=50+xbar //cm
Iyy=7433 //cm^2
k=(1/210+e*xbar/Iyy)
P=t/k
k1=(-1/210+e*(xbar+5)/Iyy)
P1=c/k1
P_safe=min(P1,P)
// results
printf('The safe load is %d kg',P_safe)
```

Scilab code Exa 9.5 Chapter9 Example 5

```
clc
//initialization of the variables
clear
s=1.6 //m
s1=4 //m
pi=28 //degrees
w=16 //kg/m^2
p=100 //kg/m^2
pl=20 //cm
pb=10 //cm
r=500 //kg/m^3
// calculations
pi=pi*%pi/180 //radians
W=w*s+(r*pl*pb/(100*100))
P=p*s
```

```
L=P+W*cos(pi)
Mx=L*s1^2*100/8
sigma_1=Mx*6/(pb*pl^2)
My=W*sin(pi)*s1^2*100/8
sigma_2=My*6/(pl*pb^2)
sigma=sigma_1+sigma_2
// results
printf('Due to bending in the noth the planes, D
   experiences maximum \n compression of %.2f kg/cm
   ^2 and B has maximum tension of %.2f kg/cm^2',
   sigma,sigma)
```

Scilab code Exa 9.6 Chapter9 Example 6

```
clc
//initialization of the problems
clear
s=1.6 //m
s1=4 //m
pi=28 //degrees
w=16 //kg/m^2
p=100 //kg/m^2
pl=20 //cm
pb=10 //cm
r=500 //kg/m^3
Zx=54.8 //cm^3
Zy=3.9 //cm^3
// calculations
pi=pi*%pi/180 //radians
W=w*s+8.1
P=p*s
L=P+W*cos(pi)
Mx=L*s1^2*100/8
sigma_1=Mx/Zx
My=W*sin(pi)*s1^2*100/8
```

```
sigma_2=My/Zy
sigma=sigma_1+sigma_2
// results
printf('Maximum stresses are %d kg/cm^2, tension or
    compression',sigma)
```

Scilab code Exa 9.7 Chapter9 Example 7

```
clc
//initialization of variables
clear
s=1.6 //m
s1=4 //m
pi=28 //degrees
w=16 //kg/m^2
p=100 //kg/m^2
pl=20 //cm
pb=10 //cm
r=500 //kg/m^3
sg=5 //cm
E=12*10^4
pi=pi*%pi/180 //radians
// calculations
W=w*s+(r*pl*pb/(100*100))
P=p*s
L=P+W*cos(pi)
Mx=L*s1^2*100/8
sigma_1=Mx*6/(pb*pl^2)
My=W*sin(pi)*s1^2*100/8
sigma_2=My*6/(pl*pb^2)
st=sigma_1*sg/10
Ts=st-sigma_2
ez=Ts/E
// results
printf('The strain gauge, aligned to the z axis will
```

```
give compression strain of %.1e',ez)
```

Scilab code Exa 9.8 Chapter9 Example 8

```
clc
//initialization of variables
clear
P=3 //tonne/m
s=6 //m
l=50 //cm
b=20 //cm
k=0.5 //m
//calculations
R=P*s/2
sf=R-k*P
bm=R*k-P*k^2/2
tau_xy=1.5*sf*1000/(l*b)
tau_max=tau_xy
str=bm*s*10^5/(b*l*l)

// consider the line a-a

sigma_x=str*12.5/25
sigma_y=0
tau_xy=tau_xy*(1-(12.5/25)^2)

sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)

printf('For the line a-a the bending stress and
    shearing stress are \n respectively %.2f kg/cm^2,
     %.2f kg/cm^2 ',sigma_x,tau_xy)
printf('\n The principal stresses are %.2f kg/cm^2 (
```

```
tension) %.2f kg/cm^2 (compression) ',sigma_1,sigma_2)

//consider the line c-c
printf('\n For the line c-c the bending stress and shearing stress are \n respectively %.2f kg/cm^2, %.2f kg/cm^2 ',sigma_x,tau_xy)
printf('\n The principal stresses are %.2f kg/cm^2 (compression) %.2f kg/cm^2 (tension) ',sigma_2,sigma_1)

//for the line b-b
tau_xy=tau_max
sigma_x=0
sigma_y=0
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-sigma_y))^2+tau_xy^2)
// results
printf('\n For the line b-b the bending stress and shearing stress are \n respectively %.2f kg/cm^2, %.2f kg/cm^2 ',sigma_x,tau_xy)
printf('\n The principal stresses are %.2f kg/cm^2 (tension) %.2f kg/cm^2 (compression) ',sigma_1,sigma_2)
```

Scilab code Exa 9.9 Chapter9 Example 9

```
clc
//initialization of variables
clear
P=3 //tonne/m
s=6 //m
l=50 //cm
```

```
b=20 //cm
k=0.5 //m
//calculations
R=P*s/2
sf=R-k*P
bm=R*k-P*k^2/2
tau_xy=1.5*sf*1000/(l*b)  //max shear stress
tau_max=tau_xy
str=bm*s*10^5/(b*l*l)  //max bending stress

// consider the line a-a

sigma_x=str*12.5/25
sigma_y=0
tau_xy=tau_xy*(1-(12.5/25)^2)

sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)

theta=1/2*atan(2*tau_xy/(sigma_x-sigma_y))
sigma_p=sigma_1/cos(theta)
P=sigma_p*2*l*b/(3*1000)
printf('A prestressing force of %.2f Tonne must be
    applied to balance the tension at a-a',P)

//At bottom point D or C
pre_str=P*2*1000/(l*b)
net=str-pre_str
printf('\n At bottom point D or C')
printf('\n Net tension = %.2f kg/cm^2 ',net)

//consider the line b-b
pre_str=P
sigma_x=pre_str
sigma_y=0
tau_xy=tau_max
```

```
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
   sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
   sigma_y))^2+tau_xy^2)
printf('\n At section b-b')
printf('\n pre-stress=%.2f kg/cm^2',pre_str)
printf('\n principal stresses are %.2f, %.2f kg/cm^2
   ',sigma_1,sigma_2)

//for the line c-c
sigma_x=str*12.5/25
sigma_y=0
tau_xy=tau_xy*(1-(12.5/25)^2)
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
   sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
   sigma_y))^2+tau_xy^2)
pre_str=pre_str/2
net=sigma_1+pre_str
sigma_x=net
sigma_y=0
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
   sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
   sigma_y))^2+tau_xy^2)
// results
printf('\n At section c-c')
printf('\n the direct stress is %.2f kg/cm^2',net)
printf('\n pre-stress = %.2f kg/cm^2',pre_str)
printf('\n principal stresses are %.2f, %.2f kg/cm^2
   ',sigma_1,sigma_2)

// wrong calculations in the thext for some parts
```

Scilab code Exa 9.10 Chapter9 Example 10

```
clc
//initialization of variables
clear
b=2 //cm
h=2 //cm
T=2000 //kg-cm
V=250 //kg
M=2000 //kg-cm
// calculations
Mmax=M*6/(b*h*b)
Vmax=3*V/(2*b*h)
Zt=0.208*b^2*h
Tmax=T/(Zt)

sigma=Mmax
printf('points A,B,')
printf('\n sigma=%d kg/cm^2 (tension)',sigma)
printf('\n points C,D,')
printf('\n sigma=%d kg/cm^2 (cmpression)',sigma)
tau=Vmax+Tmax
printf('\n point E')
printf('\n tau=%.2f kg/cm^2 shear',tau)
tau=Vmax-Tmax
printf('\n tau=%.2f kg/cm^2 shear',tau)
// at G
sigma_x=sigma
sigma_y=0
tau_xy=Tmax
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
// results
printf('\n at point G')
printf('\n sigma_1 = %d kg/cm^2 (tension)',sigma_1)
printf('\n sigma_2 = %d kg/cm^2 (compression)',
    sigma_2)
```

```
// Question was asked only to find out at A,B,C,D,E,
   F and G
```

Scilab code Exa 9.11 Chapter9 Example 11

```
clc
//initialization of variables
clear
w=10 //cm
s=2.8 //m
P=1 //tonne
Ft=1.4 //cm
Wt=0.8 //cm
Ix=13989.5 //cm^4
Z=699.5 //cm^3
// calculations
BM= 2.8
T=P*1000*8.21
SF=P*1000
BS=BM*10^5/(Z)
sigmaXA=BS*18.6/20
K=w*Ft*19.3+18.6*Wt*9.3
tau_xy_C=SF/(Ix*Wt)*K
tau_xy_A=tau_xy_C*(w*Ft*19.3)/K
tau_xy_B=tau_xy_A*0.5*Wt/w
sigmaXB=sigmaXA*19.3/20

tau_max=3*Ft*8210/(w*Ft^3+37.2*Wt^3)
tau_A=3*Wt*8210/(w*Ft^3+37.2*Wt^3)

//For point A
Shear=tau_xy_A-tau_A
sigma_x=sigmaXA
sigma_y=0
tau_xy=Shear
```

```
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)

printf('For point A')
printf('\n Total shear= %.1f kg/cm^2 ',Shear)
printf('\n Bending stress = %d kg/cm^2 (Compr.)',
    sigma_x)
printf('\n Principal stresses are %d (tension), %d (
    comp.) kg/cm^2 ',sigma_1,sigma_2)

//For point B
printf('\n FOr point B')
printf('\n Bending shear stress is %.2f k/cm^2',
    tau_xy_B)
sigmaXB=BS*19.3/20
sigma_x=sigmaXB
sigma_y=0
tau_xy=tau_max
sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
    sigma_y))^2+tau_xy^2)
printf('\n Principal stresses are %d (tension), %d (
    comp.) kg/cm^2 ',sigma_1,sigma_2)

// Answers in the text are approximations
```

Scilab code Exa 9.12 Chapter9 Example 12

```
clc
//initialization of variables
clear
b=10 //cm
```

```
h=10 //cm
P=5 //tonne
e=1 //cm
E=12*10^4 //kg/cm^2
str=130 // kg/cm^2
n=3
L=2 //m
// calculations
L=L*100 //cm
Pcr=%pi^2*E*b*h^3/(12*L^2)
Pcr=Pcr/1000
Smax=-P*1000/(b*h)-(P*1000*1*5*12/10^4)*1/(1-(n*P/
   Pcr))
// results
printf('permissible stress = %d kg/cm^2',str)
printf('\n develoed stress = %.1f kg/cm^2',Smax)
printf('\n Since it is below the permissible stress,
    the design is safe')
```

Scilab code Exa 9.13 Chapter9 Example 13

```
clc
//initializatio of variables
clear
// linked to 9.13
b=10 //cm
h=10 //cm
P=5 //tonne
e=1 //cm
E=12*10^4 //kg/cm^2
str=130 // kg/cm^2
n=3
L=2 //m
// calculations
L=L*100 //cm
```

```
Pcr=%pi^2*E*b*h^3/(12*L^2)
Pcr=Pcr/1000
Smax=-P*1000/(b*h)-(P*1000*1*5*12/10^4)*1/(1-(n*P/
    Pcr))
Smax=abs(Smax)

rr=b*h^3/(12*100)
Smax_se=P*1000/(b*h)*(1+e*5/rr*sec(%pi/2*sqrt(n*P/
    Pcr)))
Perror=(Smax-Smax_se)/Smax
Perror=Perror*100
Perror=abs(Perror)
// results
printf('Using secent formula, stress obtained is %d
    kg/cm^2',Smax_se)
printf('\n hence, the percentage error %.2f',Perror)
// approximate answees in the text
```

Scilab code Exa 9.14 Chapter9 Example 14

```
clc
//initialization of variables
clear
P=400 //kg/m
L=10 //m
F=10 //tonne
n=3
Ixx=5943.1 //cm^4
A=52.03 //cm^2
rx=10.69 //cm
E=2*10^6 //kg/cm^2
// calculations
Pcr=%pi^2*E*Ixx/((L*100)^2)
Pcr=Pcr/1000
e=P*L^2/(8*F*1000)
```

```
g=e*12.5*100/rx^2
Smax=F*1000/A*(1+g*1/(1-n*(F/Pcr)))
// results
printf('The maximum stress developed is %d kg/cm^2',Smax)

// approximate calculations involved in the text book
```

Scilab code Exa 9.15 Chapter9 Example 15

```
clc
//initialization of variables
clear
// linked to 9_14
// calculations
P=400 //kg/m
L=10 //m
F=10 //tonne
n=3
Ixx=5943.1 //cm^4
A=52.03 //cm^2
rx=10.69 //cm
E=2*10^6 //kg/cm^2
Pcr=%pi^2*E*Ixx/((L*100)^2)
Pcr=Pcr/1000
e=P*L^2/(8*F*1000)
g=e*12.5*100/rx^2
Smax=F*1000/A*(1+g*1/(1+n*(F/Pcr)))
// results
printf('The maximum stress developed is %d kg/cm^2',Smax)

// approximate answer in the text
```

Chapter 10

INTRODUCTION TO ENERGY METHODS

Scilab code Exa 10.3 Chapter10 Example 3

```
clc
//initialization of variables
clear
L=6000 //cm
L1=150 //cm
T=90 //W
Ip=%pi*10^4/32
E=2*10^6 //kg/cm^2
G=E/2.5
A=3 //cm^2
delta=0.5
//calculations
U=L/(2*E*A)+(T*T*L1/(2*G*Ip))
// U=0.5*W*delta
W=0.25/U
//results
printf('W = %.1f kg',W)
```

Scilab code Exa 10.4 Chapter10 Example 4

```
clc
//initialization of variabes
clear
cA=10 //cm^2
wA=5 //cm^2
P=1 //tonne
E=2*10^6 //kg/cm^2
P=P*1000 //kg
// calculations
U_up=P^2*200/(2*E*cA)*1/sqrt(3)*(2+4+6+8+10+12)
U_do=P^2*200/(2*E*cA)*1/sqrt(3)*(1+3+5+7+9+11+13/2)
U_web=P^2*200/(2*E*wA)*1/sqrt(3)*(2*13)
U=U_up+U_do+U_web
delta=U*2/(P)
// results
printf('deflection = %.3f cm',delta)
```

Scilab code Exa 10.7 Chapter10 Example 7

```
clc
//initialization of variables
clear
L=1 //m
w=10 //kg
h=50 //cm
A=1 //cm^2
E=2*10^6 //kg/cm^2
Ar=1 //cm^2
Ec=3*10^4 //kg/cm^2
// For steel
```

```
del=w*L*100/(A*E)
P=w*(1+sqrt(1+(2*h/del)))
printf('Stress in steeel = %d kg/cm^2 ',P)

// for cloth laminate
del=w*L*100/(A*Ec)
P=w*(1+sqrt(1+(2*h/del)))
printf('\n Stress in cloth laminate = %.1f kg/cm^2 '
    ,P)
```

Scilab code Exa 10.8 Chapter10 Example 8

```
clc
//initialization of variables
clear
w=64 //kg
H=60 //cm
b=40 //cm
h=5 //cm
E=0.12*10^6 //kg/cm^2
Es=2*10^6 //kg/cm^2
// for part (a) and (b)
I=b*h^3/12
del=4*w*120^3/(E*I)
P=w*(1+sqrt(1+(2*H/del)))
str=P*240*6/(b*h^2)
printf('part (a) and (b)')
printf('\n Maximum stress in wood = %d kg/cm^2',str)
printf('\n Max. force on divers feet = %d kg',P)

//for part (c)
Ixx=I*E/Es
Zxx=19.4 //cm^2
Ixx=72.7 //cm^4
del=4*w*120^3/(Es*Ixx)
```

```
P=w*(1+sqrt(1+(2*H/del)))
str=P*240/Zxx
// results
printf('\n part (c)')
printf('\n Maximum stress in steel = %d kg/cm^2',str)
printf('\n Max. force on divers feet = %d kg',P)
printf('\n Hence wood is better than steel')

// wrong calculations in some parts
```

Scilab code Exa 10.11 Chapter10 Example 11

```
clc
//initialization of variables
clear
A=100 //cm^2
E=2*10^6 //kg/cm^2
// calculations
del=1093.5*10^6/(E*A)
// 1093.5 from the table
// results
printf('Central deflection = %.2f mm',del)
```

Scilab code Exa 10.12 Chapter10 Example 12

```
clc
//initialization of variables
clear
T=30 //degree celcius
alpha=0.0000117 // per degree celcius
//AB
L=6 //m
```

```
dl=T*alpha*L
df=0.375 //kg
tot=dl*df
//BC
dl=T*alpha*L
df=0.375 //kg
tot=tot+dl*df
//CD
dl=T*alpha*L
df=0.75 //kg
tot=tot+dl*df
tot=tot*100*2
// results
printf('The deflection is %.4f cm',tot)
```