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

July 31, 2019

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

```
1  clc
2  // initialization of variables
3  clear
4  tau =[200 100 0
5       100 0 0
6       0 0 500] // some units
7  theta =60 // degrees
8  // calculations
9  theta = theta *%pi /180
10 a=[cos (theta) sin (theta) 0
11   -sin (theta) cos (theta) 0
12   0 0 1]
13 b=a.'
14 tau_new =a* tau *b
15 // Results
16 printf ('The new stress tensor is ')
17 disp ( tau_new )
```

Scilab code Exa 2.5 Chapter2 Example 5

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

Scilab code Exa 2.7 Chapter2 Example 7

```
1  clc
2  // initialization of variables
3  clear
4  tau=[200 100 0
5      100 0 0
6      0 0 500] // some units
7  theta=60 // degrees
8  //calculations
9  theta=theta*%pi/180
10 a=[cos(theta) sin(theta) 0
11   -sin(theta) cos(theta) 0
12    0 0 1]
13 b=a.'
14 tau_new=a*tau*b
15
16 // stress invariants :old
17 I1=tau(1,1)+tau(2,2)+tau(3,3)
18 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)
19 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)
20
21  // stress invariants : new
22  I11 = tau_new (1 ,1) + tau_new (2 ,2) + tau_new (3 ,3)
23  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)
24  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)
25
26  // Results
27  printf ('The invariants of old stress tensor are I1=
      %0.2 f I2=%. e I3=%. e \n and that of the new stress
       tensor are I1=%0.2 f I2=%. e I3=%. e ' ,I1 , I2 , I3 , I11 ,
      I22 , I33 )
28  printf ('\n Hence the same stress tensor invariants ')
```

Scilab code Exa 2.8 Chapter2 Example 8

```
1  clc
2  // initialization of variables
3  clear
4  sigma_3 =0 // kgf/cm^2
5  tau_oct =1500 // kgf/cm^2
6  n=2 // given that sigma_1=n*sigma_2
7  // calculations
8  sigma_2 =1500*3/( sqrt (2*n^2 -2*n+2)) // // kgf/cm^2
9  sigma_1 =n*sigma_2 // kgf/cm^2
10 sigma_0 =4500/ sqrt (2) // kgf/cm^2
11 // Results
12 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

```
1  clc
2  // initialization of variables
3  clear
4  // part (a)
5  tau_xx=300 // kgf/cm^2
6  tau_yy=0 // kgf/cm^2
7  tau_xy=600 // kgf/cm^2
8  // calculations
9  sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))
      ^2+tau_xy^2)
10 sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))
      ^2+tau_xy^2)
11 Beta=atan(2*tau_xy/(tau_xx-tau_yy))
12 Beta=Beta*180/%pi
13 // Results
14 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)
15
16 //part (b)
17 tau_xx=1000 // kgf/cm^2
18 tau_yy=150 // kgf/cm^2
19 tau_xy=450 // kgf/cm^2
20 // calculations
21 sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))
      ^2+tau_xy^2)
22 sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))
      ^2+tau_xy^2)
23 Beta=atan(2*tau_xy/(tau_xx-tau_yy))
```

```
24  Beta=Beta*180/%pi
25  // Results
26  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)
27
28  // part (c)
29  tau_xx=-850 // kgf/cm^2
30  tau_yy=350 // kgf/cm^2
31  tau_xy=700 // kgf/cm^2
32  // calculations
33  sigma_1=(tau_xx+tau_yy)/2+sqrt((1/2*(tau_xx-tau_yy))
        ^2+tau_xy^2)
34  sigma_2=(tau_xx+tau_yy)/2-sqrt((1/2*(tau_xx-tau_yy))
        ^2+tau_xy^2)
35  Beta=atan(2*tau_xy/(tau_xx-tau_yy))
36  Beta=Beta*180/%pi
37  // Results
38  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)
39
40  // wrong answers were given in textbook for part (b)
```

Scilab code Exa 2.10 Chapter2 Example 10

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

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

Scilab code Exa 2.11 Chapter2 Example 11

```
 1  clc
 2  // initialization of variables
 3  clear
 4  d=2 // m
 5  l=10 // m
 6  t=1 // cm
 7  p=15 // kgf/cm^2
 8  pitch= 2*%pi //m
 9  //calculations
10  w=2*%pi*d/2 // m
11  theta=atan(w/(2*%pi))
12  sigma_z=p*d*100/(4*t)
13  sigma_th=p*d*100/(2*t)
14  sigma_th_new=(sigma_th+sigma_z)/2+(sigma_th-sigma_z)
        /2*cos(2*theta)
15  tau_thz=(sigma_z-sigma_th)*sin(2*theta)/2
16  // results
17  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

```
1  clc
2  // initialization of variables
3  clear
4  epsillon =[0.01 0 0
5      0 0.02 0.02
6      0 0.02 0.01] // dimensionless
7  theta =30 // degrees
8  // calculations
9  theta=theta *%pi/180
10 a=[cos(theta) sin(theta) 0
11    -sin(theta) cos(theta) 0
12     0 0 1]
13 b=a.'
14 epsillon_new=a*epsillon*b
15 // calculation of strain invariants
16 // for epsillon
17 J1=epsillon(1,1)+epsillon(2,2)+epsillon(3,3)
18 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)
19 J3=epsillon(1,1)*epsillon(2,2)*epsillon(3,3)+2*
```

```scilab
      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)
20
21 // for epsillon_new
22 J11= epsillon_new (1 ,1)+ epsillon_new (2 ,2)+ epsillon_new
      (3 ,3)
23 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)
24 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)
25
26 // results
27 printf ( 'The new strain tensor is ');
28 disp ( epsillon_new );
29 printf ( 'The strain invariants of old strain tensor
      are J1=%0.2 f J2=%. e J3=%. e \n and that of the new
       strain tensor are J1=%0.2 f J2=%. e J3=%. e ',J1 ,J2 ,
      J3 ,J11 ,J22 ,J33)
30 printf ( '\n Hence the same strain invariants ')
```

Scilab code Exa 3.4 Chapter3 Example 4

```scilab
1 clc
2 // initialization of variables
3 clear
4 epsillon =[0.01 -0.02 0
5      -0.02 0.03 -0.01
6      0 -0.01 0] // dimensionless
7 a_xx =0.6
```

```
 8  theta=acos(a_xx) // radians
 9  //calculations
10  // theta=theta*%pi/180
11  a=[cos(theta) 0 -sin(theta)
12     0          1          0
13     sin(theta) 0 cos(theta)]
14  b=a.'
15  epsillon_new=a*epsillon*b
16
17  // calculation of strain invariants
18  // for epsillon
19  J1=epsillon(1,1)+epsillon(2,2)+epsillon(3,3)
20
21  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)
22
23  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)
24
25  // for epsillon_new
26  J11=epsillon_new(1,1)+epsillon_new(2,2)+epsillon_new
       (3,3)
27
28  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)
29
30  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)
31
32  // Results
```

```
33  printf('The new strain tensor is ')
34  disp(epsillon_new)
35  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

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

Scilab code Exa 3.6 Chapter3 Example 6

```
 1  clc
 2  // initialization of variables
 3  clear
 4  epsillon_A= 1000*10^-6
 5  epsillon_B= 720*10^-6
 6  epsillon_C= 600*10^-6
 7  th_B=120 // degrees
 8  th_C=240 // degrees
 9  //calculations
10  th_B=th_B*%pi/180
11  th_C=th_C*%pi/180
12  // we need to solve for epsillon_y and gamma_xy
13  // Ax=B
14  ep_x=epsillon_A
15  A=[sin(th_B)^2 sin(th_B)*cos(th_B)
16      sin(th_C)^2 sin(th_C)*cos(th_C)]
17  C=[epsillon_B-ep_x*cos(th_B)^2; epsillon_C-ep_x*cos(
        th_C)^2]
18  x=inv(A)*C
19  ep_y=x(1,1)
20  gam_xy=x(2,1)
21  epsillon_x=ep_x
22  epsillon_y=ep_y
23  gamma_xy=gam_xy
24  epsillon_1=1/2*(epsillon_x+epsillon_y)+(1/2)*sqrt((
        epsillon_x-epsillon_y)^2+gamma_xy^2)
25  epsillon_2=1/2*(epsillon_x+epsillon_y)-(1/2)*sqrt((
        epsillon_x-epsillon_y)^2+gamma_xy^2)
26  // Results
27  printf('The principal strains are %.3e, %.3e',
        epsillon_1,epsillon_2)
```

Chapter 4

STRESS STRAIN RELATIONS

Scilab code Exa 4.1 Chapter4 Example 1

```
1  clc
2  // initialization of variables
3  clear
4  E=2*10^6 // kg/cm^2
5  G=8*10^5 // kg/cm^2
6  ep=[0.001 0 -0.002
7       0 -0.003 0.0005
8       -0.002 0.0005 0]
9  // calculations
10 nu=E/(2*G)-1
11 lambda=E*nu/((1+nu)*(1-2*nu))
12 mu=G
13 sigma(1,1)=2*mu*ep(1,1)+lambda*(ep(1,1)+ep(2,2)+ep
       (3,3))
14 sigma(2,2)=2*mu*ep(2,2)+lambda*(ep(1,1)+ep(2,2)+ep
       (3,3))
15 sigma(3,3)=2*mu*ep(3,3)+lambda*(ep(1,1)+ep(2,2)+ep
       (3,3))
16 tau(1,2)=2*mu*ep(1,2)
```

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

Scilab code Exa 4.2 Chapter4 Example 2

```
1   clc
2   // initialization of variables
3   clear
4   sigma_x=1000  //kg/cm^2
5   sigma_y=-500  //kg/cm^2
6   sigma_z=0   //kg/cm^2
7   tau_xy=500  //kg/cm^2
8   E=2*10^6  // kg/cm^2
9   nu=0.25
10  //calculations
11  ep_x=1/E*(sigma_x-nu*(sigma_y+sigma_z))
12  ep_y=1/E*(sigma_y-nu*(sigma_x+sigma_z))
13  ep_z=1/E*(sigma_z-nu*(sigma_y+sigma_x))
14  J1=ep_x+ep_y+ep_z
15  sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
        sigma_y))^2+tau_xy^2)
16  sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
        sigma_y))^2+tau_xy^2)
17  th=1/2*atan(2*tau_xy/(sigma_x-sigma_y))
18  th=th*180/%pi
19  ep_1=1/E*(sigma_1-nu*sigma_2)
20  ep_2=1/E*(sigma_2-nu*sigma_1)
```

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

Scilab code Exa 4.3 Chapter4 Example 3

```
1  clc
2  // initialization of variables
3  clear
4  sigma_x =1400  //kg/cm^2
5  tau_xy =400 //  kg/cm^2
6  ep_z = -3.6*10^ -6
7  nu =1/4
8  E=2*10^8  //kg/cm^2
9  // calculations
10  sigma_y =(-ep_z*E/nu)-sigma_x
11  sigma_1 =(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
         sigma_y))^2+tau_xy^2)
12  sigma_2 =(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
         sigma_y))^2+tau_xy^2)
13  th =0.5*atan(2*tau_xy/(sigma_x-sigma_y))
14  th=th*180/%pi
15  printf('sigma_y is %d kg/cm^2',sigma_y)
16  printf('\n The principal stresses are %d , %d kg/cm
         ^2',sigma_1,sigma_2)
17  printf('\n The direction is given by theta = %.2f
         degrees',th)
18
19  // angle was given wrong in the text
```

Scilab code Exa 4.4 Chapter4 Example 4

```
1  clc
2  // initialization of variables
3  clear
4  C=1000/3  // kg/cm^2
5  sigma_x=2*C
6  sigma_y=4*C
7  tau_xy=4*C
8  sigma_0=4*C
9  sigma_1=3+C*sqrt(2)
10 sigma_2=3-C*sqrt(2)
11 sigma_3=0
12 tau_oct=1/3*sqrt((sigma_1-sigma_2)^2+(sigma_2-
      sigma_3)^2+(sigma_3-sigma_1)^2)
13 tau_max=sigma_1/2
14 taU=1.885*C
15 tau_y=2*C
16 printf('Actual tau is %.3f',taU)
17 printf('\n tau_max at yield is %.3f',tau_y)
18 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

```
1  clc
2  //initialization of variables
3  clear
4  l=20  //cm
5  dL=1  //m
6  dl=0.004  //cm
7  //calculations
8  L=l*dL/dl  //m
9  //results
10 printf('The depth of the clay bed is %d m',L)
```

Scilab code Exa 5.2 Chapter5 Example 2

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

```
6  // calculations
7  db=3000*90/(A*E)
8  dc=db+5000*60/(A*E)
9  dd=dc+4000*30/(A*E)
10 //results
11 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

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

Scilab code Exa 5.4 Chapter5 Example 4

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

24

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

Scilab code Exa 5.6 Chapter5 Example 6

```
1  clc
2  // initialization of variables
3  clear
4  P=10 // tonne
5  E=2*10^6 // kg/cm^2
6  // calculations
7  // We have to solve linear system Ax=B
8  A=[1  1  1  0
9     3  1  -3  0
10    -2  2  0  -E
11    0  -1  2  -E]
12 B=[P*10^3;0;0;0]
13 x=inv(A)*B
14 W1=x(1,1)/1000
15 W2=x(2,1)/1000
16 W3=x(3,1)/1000
17 th=x(4,1)
18 // results
19 printf('The load taken by each rod is %.2f tonne , %
     .1f tonne , %.3f tonne',W1,W2,W3)
20 printf('\n and the slope is theta = %.2e. radians',
     th)
```

Scilab code Exa 5.8 Chapter5 Example 8

```
1  clc
2  // initialization of variables
3  clear
4  b=30  // cm
5  h=30  //cm
6  n=6
7  A=36  //cm^2
8  ss_s=1500  //kg/cm^2
9  ss_c=60  //kg/cm^2
10 Er=15  // Elasticity ratio
11 // calculations
12 L=A*Er*ss_c+(b*h-A)*ss_c
13 // results
14 printf('The safe load is %d.kg',L)
```

Scilab code Exa 5.9 Chapter5 Example 9

```
1  clc
2  // initiaization of variables
3  clear
4  gs_b=10  //cm
5  gs_h=10  //cm
6  d_b=2  //cm
7  d_h=2  //cm
8  As= 1  //cm^2
9  s=10000  //kg/cm^2
10 // part (a)
11 Es=2*10^6  //kg/cm^2
12 Ec=2*10^5  //kg/cm^2
13 // calculations
14 e=s/Es
15 Ac=gs_b*gs_h-(d_b*d_h)
16 e_c=e*Es*As/(Ec*Ac+Es*As)
17 s_c=Ec*e_c
18 e_s=e-e_c
```

```
19  s_s=Es*e_s
20  // results
21  printf('part (a) \n The stress in steel and concrete
        are respectively %d , %.2e kg/cm^2',s_s,s_c)
22  // part(b)
23  P=8000  //kg
24  // calculations
25  e_c=(e*Es*As-P)/(Ec*Ac+Es*As)
26  e_s=e-e_c
27  s_c=Ec*e_c
28  s_s=Es*e_s
29  // results
30  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

```
 1  clc
 2  // initialization
 3  clear
 4  d=10  //cm
 5  id=9.99  //cm
 6  t=3  //mm
 7  E=1.0*10^6  //kg/cm^2
 8  a=2.02*10^-5 // degree/celcius
 9  // part(a)
10  Tr=10  //degree C
11  T=(d-id)/id*1/a
12  printf('part(a) \n The sleeve must be heated to %.1f
        degree C or more for this purpose',T+Tr)
13
14  //part(b)
15  s_th=a*T*E
16  p=s_th*t*2/(d*10)
```

```
17  printf ('\n part(b) \n The pressure developed between
         the rod and sleeve is %d kg/cm^2',p)
18
19  // part(c)
20  f =0.2
21  o =10 // overlap: cm
22  A=%pi*d*o
23  F=f*p*A
24  printf ('\n part (c) \n The axial force required is
         %d kg',F)
25
26  //part (d)
27  // linked to part c
28  T2=20 //degree C
29  a2=1.17*10^-5 //  /degree C
30  Ts=(a-a2)*(T2-Tr)*E
31  Ts=s_th-Ts
32  p2=p*Ts/s_th
33  F2=F*Ts/s_th
34  printf ('\n part(d)\n The pressure developed between
         the rod and sleeve is %.1f kg/cm^2',p2)
35  printf ('\n The axial force required is %d kg',F2)
36  //part(e)
37  T3=Tr+(s_th/((a-a2)*10^6))
38  printf ('\n part(e) \n The temperature at which the
         sleeve comes off easily is %.1f C',T3)
39
40  // calculations in the text: rounding off errors
```

Scilab code Exa 5.11 Chapter5 Example 11

```
1  clc
2  //initialization of variables
3  clear
4  T1=37.8 // degre C
```

28

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

Scilab code Exa 5.12 Chapter5 Example 12

```
 1  clc
 2  // initialization of variables
 3  clear
 4  L=5  //cm
 5  D=1.8  //cm
 6  l=2.5  //cm
 7  d=1.5  //cm
 8  F=1  //tonne
 9  E=2.1*10^6  //kg/cm^2
10  // calculations
11  s1=F*1000*4/(D^2*%pi)
12  s2=F*1000*4/(d^2*%pi)
13  U1=1/2*s1^2/E
14  U1=U1*L*D^2*%pi/4
15  U2=1/2*s2^2/E
16  U2=U2*l*d^2*%pi/4
17  U=U1+U2
18  // results
```

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

Scilab code Exa 5.13 Chapter5 Example 13

```
1  clc
2  // initialization of variables
3  clear
4  t=16  //mm
5  Pt=1500  //kg/cm^2
6  Ps=1025  //kg/cm^2
7  Pb=2360  //kg/cm^2
8
9  //part (a)
10 p=6  //cm
11 r=24  //mm
12 d=r/10+0.15
13 Ft=t*(p-d)*Pt/10
14 Fs=%pi*d^2*Ps/4
15 Fb=d*t*Pb
16 x=min(Ft,Fs,Fb)
17 effA=x*100/(p*t/10*Pt)
18
19 //part (b)
20 p=9  //cm
21 r=30  //mm
22 d=r/10+0.2
23 Ft=t*(p-d)*Pt/10
24 Fs=%pi*d^2*Ps/4
25 Fb=d*t*Pb
26 x=min(Ft,Fs,Fb)
27 effB=x*100/(p*t/10*Pt)
28
29 // results
30 printf('The efficiencies corresponding to cases a
```

```
        and b are %.1 f , %.1 f ' , effA , effB )
31  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

```
1  clc
2  // initialization o variables
3  clear
4  p=5  //cm
5  D=10  //cm
6  d=2  //mm
7  T= 10  //kgm
8  ss= 785  //kg/cm^2
9  // calculations
10 P= 2*T/(%pi*D^2)
11 P=P*5*100
12 // results
13 printf('Force per rivet is %.1f kg',P)
14 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

```
1  clc
2  // initialization of variables
3  clear
4  D=5  //cm
5  Y=3500  //kg/cm^2
6
7  //part (a)
8  Ta=350  //kg-m
9  tau=Y/2
10 Ip=Ta*D*100/(2*tau)
11 d1=Ip*32/%pi
12 d1=(D^4-d1)^(1/4)
13
14 //part (b)
15 Tb= 700  //kg-m
16 Ip=Tb*D*100/(2*tau)
17 d2=Ip*32/%pi
18 d2=(D^4-d2)
19 T=tau*%pi*(D^4)*2/(32*D)
20 // results
21 printf('The maximum diameter corresponding to the
       case a is %.2f cm',d1)
22 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

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

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

Scilab code Exa 6.6 Chapter6 Example 6

```
 1  clc
 2  // initialization of variables
 3  clear
 4  // linked to 6_5
 5  A=3  //cm^2
 6  E= 2*10^6  //kg/cm^2
 7  nu= 0.25
 8  l= 60  //m
 9  L=150  //cm
10  d=0.5  //cm
11  dd=10  //cm
12  D=180  //cm
13  // calculations
14  K=(l*100/(A*E))+(L*D/2*D*32*2*(1+nu)/(E*%pi*dd^4*2))
15  P=d/K
16  Ts=P/A
17  fs=dd*D*P*32/(%pi*4*dd^4)
18
19  // results
20  printf('The tensile stress is %.1f kg/cm^2',Ts)
```

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

Scilab code Exa 6.7 Chapter6 Example 7

```
1  clc
2  // initialization of variables
3  clear
4  F=500 //kg
5  k=25 //kg/cm
6  dd=15 //cm
7  ss=3500 //kg/cm^2
8  L=2 //m
9  G=8*10^5 //kg/cm^2
10 // calculations
11 x=sqrt(%pi*G/(25*L*32*100))
12 d=x*16*(F+dd*k)/(ss*%pi)
13 x2=x*d^2
14 // results
15 printf('d=%.2f cm',d)
16 printf('\n x=%.2f cm',x2)
17
18 // Text: not exact
```

Scilab code Exa 6.11 Chapter6 Example 11

```
1  clc
2  //initialization of variables
3  clear
4  d=5 //cm
5  rpm1=300 //rpm
6  rpm2=30000 //rpm
7  s=1000 //kg/cm^2
8  //calcuations
```

```
9  T=(d/2)*%pi*10^2*s/32
10 hp1= 2*%pi*rpm1*T/4500
11 hp2=hp1*100
12 // results
13 printf('Horse power at 300 rpm and 30000 rpm are
       respecively %d, %d h.p.',hp1/10,hp2/10)
14
15 // wrong/approximate answers in the text
```

Scilab code Exa 6.12 Chapter6 Example 12

```
1  clc
2  // initialization of variables
3  clear
4  hp=300 //h.p.
5  N1=30 //rpm
6  N2=30000 //rpm
7  fs=600 //kg/cm^2
8  // calculations
9  T1=4500*hp*100/(2*%pi*N1)
10 T2=T1/1000
11 D1=16*T1/(%pi*fs)
12 D1=D1^(1/3)
13 D2=16*T2/(%pi*fs)
14 D2=D2^(1/3)
15 // results
16 printf('Diameters required are %.1f, %.1f cm',D1,D2)
17
18 // wrong calculations in the text
```

Scilab code Exa 6.13 Chapter6 Example 13

```
1  clc
```

```
 2 // initialization of variables
 3 clear
 4 d=10 //cm
 5 t =1 //mm
 6 T= 100 //kg−m
 7 L=5 //m
 8 G=8*10^5 //kg/cm^2
 9
10 //calculations
11 r=d/2
12 fs=T*r*100/(r^2*2*%pi*L*t*10^-1)
13 U=fs^2/(2*G)
14 U1=U*(%pi*L*100)
15 // results
16 printf('Energy per unit volume = %.3f kg−cm/cm^3',U)
17 printf('\n Total strain energy= %d kg−cm',U1)
```

Scilab code Exa 6.14 Chapter6 Example 14

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

Scilab code Exa 6.15 Chapter6 Example 15

```
1  clc
2  //initialization of variables
3  clear
4  // linked to 6_14
5  D=10  //cm
6  d= 1  //cm
7  n=20
8  P=60  //kg
9  G=8*10^5  //kg/cm^2
10 // calculations
11 m=D/d
12 fs=8*P*D/(d^3*%pi)
13 fs1=fs*(1+0.615/m+3/(4*m-4))
14 // results
15 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

```
1  clc
2  //initialization of variables
3  clear
4
5  // circle
6  D=1 //unit diameter
7  Ip=D^4/32
8  Zp=D^3/16
9
10 //Square
11 s=sqrt(%pi/4)*D
12 Is=0.886*D^4/32
13 Zs=0.7383*D^3/16
```

```scilab
14
15  //Rectangle
16  a=sqrt(%pi/2)*D
17  b=sqrt(%pi/8)*D
18  Ir=0.719*D^4/32
19  Zr=0.616*D^3/16
20
21  //  Trianle
22  t=sqrt(%pi/sqrt(3))*D
23  It=0.725*D^4/32
24  Zt=0.622*D^3/16
25
26  //Ellipse
27  A=D/sqrt(2)
28  B=D/sqrt(8)
29  Ie=A^3*B^3/(A^2+B^2)
30  Ze=A*B^2/2
31
32  //Normalization
33  Is=Is/Ip
34  Ie=Ie/Ip
35  It=It/Ip
36  Ir=Ir/Ip
37
38  Zs=Zs/Zp
39  Ze=Ze/Zp
40  Zt=Zt/Zp
41  Zr=Zr/Zp
42  Ip=1
43  Zp=1
44  //results
45  printf('Z:: Circle:Square:Ellipse:Triangle:Rectangle
        = %.3f : %.3f : %.3f : %.3f : %.3f',Zp,Zs,Ze,Zt,
      Zr)
46  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

```
1  clc
2  //initialization of variables
3  clear
4
5  yp=2450 //kg/cm^2
6  d=0.4 //cm
7  ys=4200 //kg/cm^2
8  sa=1.6 //mm
9  sb=7 //mm
10 // calculations
11 sa=sa/10
12 sb=sb/10
13 T1=yp*%pi*d^3/16
14 T2=ys*0.303*sa^2*sb
15 // results
16 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

```
1  clc
2  //initialization of variables
3  clear
4  b=5 //cm
5  h=10 //cm
6  tL=3 //mm
7  tl=1.5 //mm
8  T=100 //kg-cm
9  // calculations
10 tl=tl/10
```

40

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

Scilab code Exa 6.19 Chapter6 Example 19

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

Scilab code Exa 6.20 Chapter6 Example 20

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

```
10  b2=14.2  //cm
11  I0=1/3*(2*b1*t1^3+b2*t2^3)
12  Zt=I0/max(t1,t2)
13  fs=T*100/Zt
14  Phi=T*100/(G*I0)
15  printf('part (a)')
16  printf('\n The maximum shear stress and twist rate
        are respectively \n %d kg/cm^2, %.2e radians/cm '
        ,fs,Phi)
17
18  //part (b)
19  t1=1  //cm
20  t2=1  //cm
21  b1=10  //cm
22  b2=9  //cm
23  I0=1/3*(b1*t1^3+b2*t2^3)
24  Zt=I0/max(t1,t2)
25  fs=T*100/Zt
26  Phi=T*100/(G*I0)
27  printf('\n part (b)')
28  printf('\n The maximum shear stress and twist rate
        are respectively \n %d kg/cm^2, %.2e radians/cm '
        ,fs,Phi)
29
30  //part (c)
31  t1=0.76  //cm
32  t2=0.48  //cm
33  b1=8  //cm
34  b2=14.04  //cm
35  I0=1/3*(2*b1*t1^3+b2*t2^3)
36  Zt=I0/max(t1,t2)
37  fs=T*100/Zt
38  Phi=T*100/(G*I0)
39  printf('\n part (c)')
40  printf('\n The maximum shear stress and twist rate
        are respectively \n %d kg/cm^2, %.2e radians/cm '
        ,fs,Phi)
41
```

```
42  // part(d)
43  t=1  //cm
44  b=19  //cm
45  I0=1/3*t^3*b
46  Zt=I0/t
47  fs=T*100/Zt
48  Phi=T*100/(G*I0)
49  printf('\n part (d)')
50  printf('\n The maximum shear stress and twist rate
        are respectively \n %d kg/cm^2, %.2e radians/cm '
        ,fs,Phi)
51
52  // Twist rate: answers differ by a scale of 10.
        wrong answers in the text
```

Scilab code Exa 6.21 Chapter6 Example 21

```
1   clc
2   // initialization of variables
3   clear
4   D=5  //cm
5   d=2  //cm
6   t_y=3000  //kg/cm^2
7   // calculations
8   R=D/2
9   r=d/2
10  Tep=2*%pi*R^3*t_y/3-%pi*r^3*t_y/6
11  t_er=2*Tep/(%pi*R^3)
12  t_er1=t_er*r/R
13  prs=t_y-t_er1
14  nrs=t_er-t_y
15  // results
16  printf('Maximum +ve residual stress occurs at %d cm
        radius and is equal to \n %d kg/cm^2',r,prs)
17  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

```scilab
1  clc
2  //initialization of variables
3  clear
4  s=3 //m
5  n=60
6  p=50 //kg
7  // calculations
8  W=n*p
9  Rc=W*2/s
10 Rb=W-Rc
11 dx = 0.001;
12 x = 0:dx:s
13 n = s/dx +1;
14 for i = 1:n
15     Sx(i) = -Rb + Rc*x(i)^2/6;
16     Mx(i) = Rb*x(i) - Rc*x(i)^3 /18;
17 end
18 //Results
19 figure(1);plot(x,Sx);title("Shear force diagram");
       xlabel("X (in m)");ylabel("Shear force (in kg)");
20 figure(2);plot(x,Mx);title("Bending Moment diagram")
```
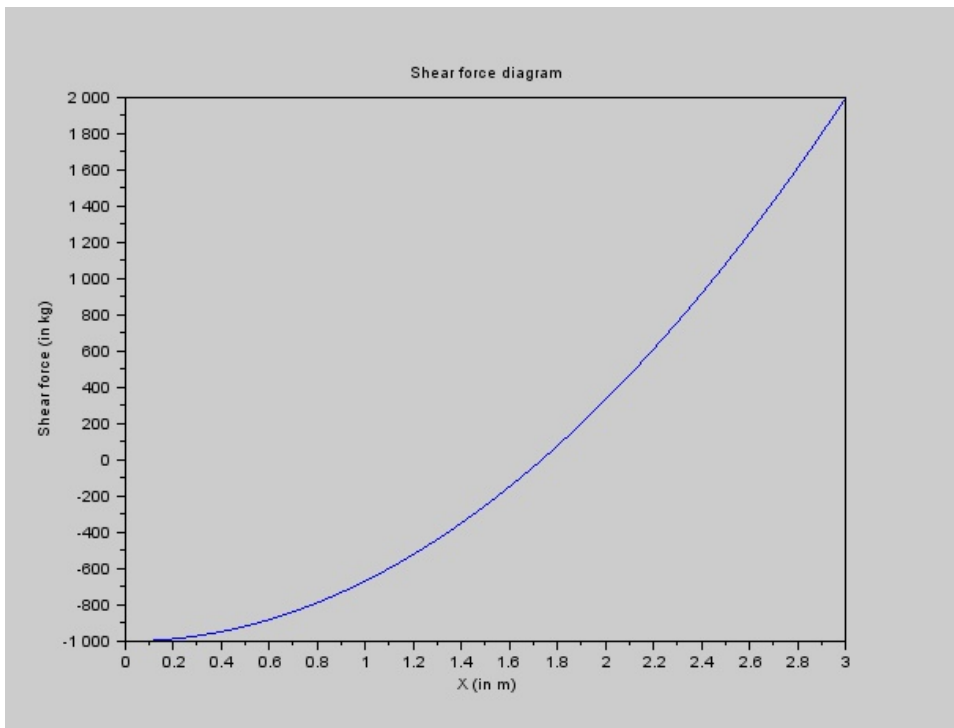
Figure 7.1: Chapter7 Example 2

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

Scilab code Exa 7.8 Chapter7 Example 8

```
1  clc
2  //initialization of new variables
3  clear
4  b=10075  //mm
```
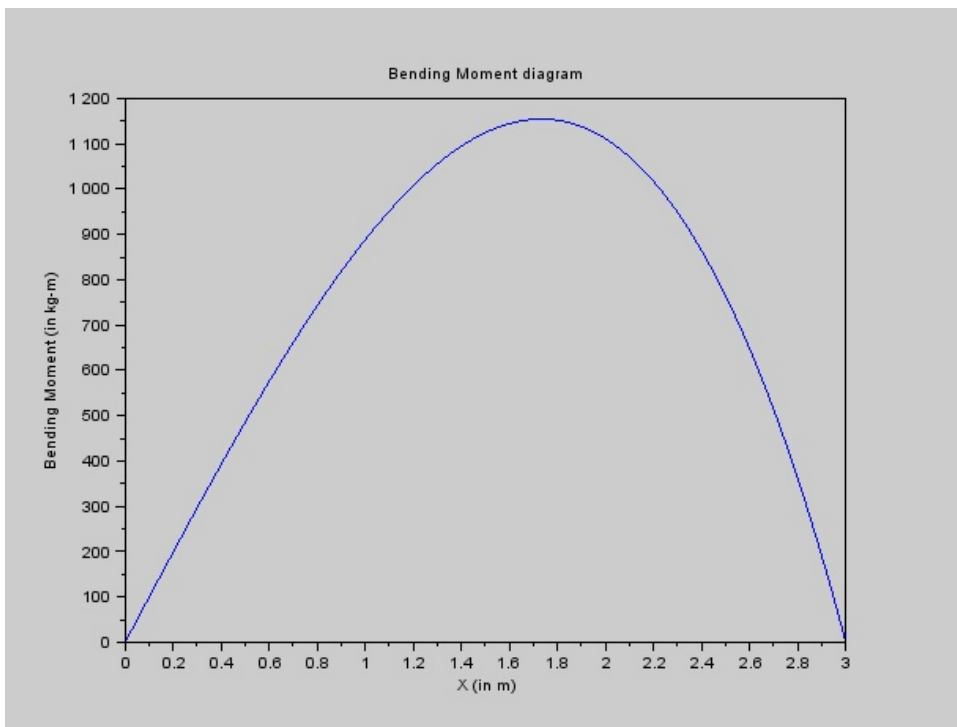
Figure 7.2: Chapter7 Example 2

```
 5  h=10  //mm
 6  p1=7.5
 7  p2=9
 8  // part (a)
 9  ybar=1*p1*0.5+1*p2*5.5
10  ybar=ybar/16.5
11  xbar=1*p1*0.5+1*p1*4.75
12  xbar=xbar/16.5
13  printf('part (a)')
14  printf('\n Centroid coordinates (x,y) = (%.2f, %.2f)
        cm',xbar,ybar)
15
16  // part (b)
17  Ixx=p1*1^3/12+p1*1*(3.23-0.5)^2+1*p2^3/12+p2
        *1*(5.5-3.23)^2
18  Iyy=1*p1^3/12+p1*1*(3.75-2.43)^2+p2*1^3/12+p2
        *1*(2.43-0.5)^2
19  Ixy=p1*1.32*2.73+9*(-1.93)*(-2.27)
20  printf('\n part (b)')
21  printf('\n Moment of Areas: \n Ixx = %.1f cm^4 \n
        Iyy = %.1f cm^4 \n Ixy=%.1f cm^4',Ixx,Iyy,Ixy)
22
23  // part (c)
24  alpha=0.5*atan(2*Ixy/(Iyy-Ixx))
25  alpha=alpha*180/%pi
26  printf('\n part (c)')
27  printf('\n Direction of principal axes:')
28  printf('\n alpha = %.2f degrees',alpha)
29
30  // part (d)
31  Iuu=(Ixx+Iyy)/2+sqrt((Iyy-Ixx)^2/4+Ixy^2)
32  Ivv=(Ixx+Iyy)/2-sqrt((Iyy-Ixx)^2/4+Ixy^2)
33  printf('\n part (d)')
34  printf('\n Iuu = %.2f cm^4 \n Ivv = %.2f cm^4',Iuu,
        Ivv)
```

Scilab code Exa 7.10 Chapter7 Example 10

```
1  clc
2  // initialization of variables
3  clear
4  Ys =17000 //kg/cm^2
5  E=2*10^6 //kg/cm^2
6  d1 =1 //mm
7  d=1 //cm
8  // calculations : 1 cm
9  R=E*d/(2*Ys)
10 M=Ys*%pi*d^3/32
11 // results
12 printf ('%d cm daimeter wire : ',d)
13 printf ('\n Minimum radius = %.2 f cm ',R)
14 printf ('\n Bending Moment = %.2 f kg-cm ',M)
15 // calculations : 1 mm
16 R1=R/(d1*10)
17 M1=M/(d1*1000)
18 // results
19 printf ('\n %d mm daimeter wire : ',d1)
20 printf ('\n Minimum radius = %.2 f cm ',R1)
21 printf ('\n Bending Moment = %.2 f kg-cm ',M1)
```

Scilab code Exa 7.11 Chapter7 Example 11

```
1  clc
2  // initialization of variables
3  clear
4  t=0.5 //cm
5  s=2 //m
6  p1 =7.5 //cm
```

```scilab
 7 p2 =10  //cm
 8 d=p2/2
 9 str =1650  //kg/cm^2
10 // calculations
11 // ab
12 IxX=p1*t^3/12+t*p1*d^2
13 // bc
14 alpha=atan(3/4)
15 Ixx=t*(p1+d)^3/12
16 Iyy=0
17 Ixy=0
18 Iuu=Ixx*cos(alpha)^2+Iyy*sin(alpha)^2-Ixy*sin(2*
      alpha)
19 Ixx=Iuu+IxX
20 IXX=Ixx*100/(2*p1)
21 Z=IXX/(d+t/2)
22 w=str*Z*8/(s^2*100)
23 w=w/1000
24 //Results
25 printf('w = %.1f tonne/m',w)
```

Scilab code Exa 7.12 Chapter7 Example 12

```scilab
 1 clc
 2 //initialization of variables
 3 clear
 4 wb=10  //cm
 5 wh=20  //cm
 6 sb=0.5  //cm
 7 sh=10  //cm
 8 s=6  //m
 9 fs=1650  //kg/cm^2
10 fw=150  //kg/cm^2
11 Es=2*10^6  //kg/cm^2
12 Ew=12*10^4  //kg/cm^2
```

```
13
14  //calculations
15  //  Method 1
16  as=2*fs/(21*Es)
17  aw=2*fw/(20*Ew)
18  a=min(as,aw)
19  ss=a*Ew*wh/2
20  //Moment resistance of steel portion
21  F=(fs+1573)/2*sb*sh
22  k=sb/3*(fs+2*1573)/(fs+1573)
23  Ms=2*F*(10.5-k)
24  //Moment resistance of wooden portion
25  F=ss*wb*wb/2
26  Mw=2*(F*(wb-wb/3))
27  M=Ms+Mw
28  //Total udl supported
29  W=M*8/(s*100)
30
31  //Results
32  printf('Using method 1')
33  printf('\n W = %d kg',W)
34
35  //Method 2
36  nE=Es/Ew
37  nf=fs/fw
38  Is=2*(0+sb*sh*10.25^2)
39  Iw=0.6*wh^3/12
40  I=Is+Iw
41  W=fs*I*8/(s*100*10.5)
42
43  //Results
44  printf('\n Using method 2')
45  printf('\n W = %d kg',W)
```

Scilab code Exa 7.13 Chapter7 Example 13

```
1  clc
2  // initialization of variables
3  clear
4  p=6  //mm
5  Ixx =2375  //cm^4
6  Es =2*10^6  //kg/cm^2
7  EAl =0.667*10^6  //kg/cm^2
8  d1 =10.6  //cm
9  d2 =10  //cm
10  // calculations
11  I1 =2*(0+p/10*10*10.3^2)
12  I2 = Ixx * EAl / Es
13  I = I1 + I2
14  n = I / I2
15  // results
16  printf ('stiffness  ratio = %.2 f ',n)
17  n1 = Es * d1 /( d2 * EAl )
18  printf ('\n  Stress  ration = %.2 f ',n1)
```

Scilab code Exa 7.14 Chapter7 Example 14

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

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

Scilab code Exa 7.15 Chapter7 Example 15

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

Scilab code Exa 7.17 Chapter7 Example 17

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

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

Scilab code Exa 7.20 Chapter7 Example 20

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

```
      and flange ')
16 printf ('\n Ratio = %.2 f ',r)
17 // part (2)
18 // calculations
19 r=(2/3*388*29.56)-(2/3*160*20.44)-(228*20.44)
20 r=r/(2*2/3*388*29.56)
21 // results
22 printf ('\n Ratio of S.F in flange to total S.F')
23 printf ('\n Ratio = %.2 f percent ',r*100)
24 // part (3)
25 // calculations
26 r=359*200/Io
27 // results
28 printf ('\n Ratio of maximum shear stress in the
      flange to average sher stress in the stem ')
29 printf ('\n Ratio = %.2 f ',r)
30 // part (4)
31 // calculations
32 s=10 //m
33 r=r/0.922
34 sigma=1650 //kg/cm^2
35 shear=945 //kg/cm^2
36 wsh=2*200*shear/(r*s)
37 wsi=8*Io*sigma/(s^2*10*29.56)
38 w=min(wsh,wsi)
39 // results
40 printf ('\n Maximum u.d.l. = %d kg/m ',w)
41
42 //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

```
1 clc
2 //initialization of new variables
```

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

Scilab code Exa 7.23 Chapter7 Example 23

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

Scilab code Exa 7.33 Chapter7 Example 33

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

Chapter 8

STABILITY OF EQUILIBRIUM COLUMNS

Scilab code Exa 8.1 Chapter8 Example 1

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

```
      Tonne ',P*4*2.13/100)
17
18  // Evaluation of critical load (P) in the text is
       wrong
```

Scilab code Exa 8.2 Chapter8 Example 2

```
1  clc
2  //initialization of variables
3  clear
4  E=2*10^6 //kg/cm^2
5  sigma_y=2600 //kg/cm^2
6  I=2502 //cm^4
7  L=500 //cm
8  A=59.7 //cm^2
9  L_tcr=L/sqrt(I/A)
10
11 printf('The actual critical length ratio is %.1f',
       L_tcr)
12 //case (b)
13 L_cr=sqrt(E*%pi^2/sigma_y)
14 printf('\n case (b)')
15 printf('\n The critical length ratio is %.1f',L_cr)
16
17 //case (a)
18 L_cr=sqrt(E*%pi^2/(4*sigma_y))
19 printf('\n case (a)')
20 printf('\n The critical length ratio is %.1f',L_cr)
21
22 //case (c)
23 L_cr=sqrt(4*E*%pi^2/sigma_y)
24 printf('\n case (c)')
25 printf('\n The critical length ratio is %.1f',L_cr)
26
27 // case (d)
```

```
28  L_cr=sqrt(2.05*E*%pi^2/sigma_y)
29  // Results
30  printf('\n case (d)')
31  printf('\n The critical length ratio is %.1f',L_cr)
32  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

```
1  clc
2  //initialzation of variables
3  clear
4  h=3.5  //m
5  A=22.4  //cm^2
6  r=7.08  //cm
7  E=2*10^6  //kg/cm^2
8  Q=1/2
9  // calculations
10  h=h*100
11  Q1=(Q*h/r)^2
12  s_cr=E*%pi^2/Q1
13  // results
14  printf('The critical stress is %d kg/cm^2',s_cr)
15  printf('\n This is much higher than yield stress for
        the material, \n so the column will fail by
        yielding')
16
17  // rounding off errors in the text
```

Scilab code Exa 8.4 Chapter8 Example 4

```
1  clc
2  //initialization of variables
3  clear
4  r_min=1.17  //cm
5  A=17.21  //cm^2
6  Q=1/2
7  h=3.5  //m
8  E=2*10^6  //kg/cm^2
9  h=h*100
10 // calculations
11 Q1=(Q*h/r_min)^2
12 s_cr=E*%pi^2/Q1
13 P_cr=s_cr*A
14 // results
15 printf('The crippling load is %d kg',P_cr)
16
17 // wrong calculations given in the text
```

Scilab code Exa 8.5 Chapter8 Example 5

```
1  clc
2  //initialization of variables
3  clear
4  L=2.5  //m
5  A=6.02  //cm^2
6  Q1=105
7  s=796.5  //kg/cm^2
8  // calculations
9  P=2*A*s
10 printf('The safe load is %d kg',P)
11 // Results
12 // wrong calculations in the text
```

Scilab code Exa 8.6 Chapter8 Example 6

```
1  clc
2  //initialization of variables
3  clear
4  h=3.5 //m
5  r_xx=7.08 //cm
6  A=24.38 //cm^2
7  Q=0.5
8  Q1=Q*h*100/r_xx
9
10 //Permissible load by secent formula
11 P=1231.28*2*A
12 printf('Permissible load by secent formula: %d kg',P
       )
13
14 //Permissible load by Rankine-Gordon formula
15 P=1260/(1+(24.75^2/18000))*2*A
16 printf('\n Permissible load by Rankine-Gordon
       formula: %d kg',P)
17
18 //Permissible load by parabolic formula
19 P=(1050-0.0233*Q1^2)*2*A
20 printf('\n Permissible load by parabolic formula: %d
        kg',P)
21
22 //Permissible load by straight-line formula
23 P=(1120-Q1*4.8)*2*A
24 printf('\n Permissible load by parabolic formula: %d
        kg',P)
25
26 // Rounding off errors in the text
```

Chapter 9

COMBINED STRESSES

Scilab code Exa 9.1 Chapter9 Example 1

```
1  clc
2  //initialization of variables
3  clear
4  //case (a)
5  A=72.9 //cm^2
6  Iy=633 //cm^4
7  Ix=1199 //cm^4
8  t=24/(5*Ix)+13.5/(5*Iy)
9  r=1/(A*t)
10 printf('case (a) \n r = %.3f cm',r)
11 // case (b)
12 t=24/(5*Ix)-13.5/(5*Iy)
13 r=1/(A*t)
14 printf('\n case (b) \n r = %.1f cm',r)
15 //case (c)
16 t=-24/(5*Ix)+13.5/(5*Iy)
17 r=1/(A*t)
18 printf('\n case (a) \n r = %.1f cm',r)
19 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

```
1  clc
2  //initialization of variables
3  clear
4  b=14  //cm
5  d=20  //cm
6  rx=8.46  //cm
7  ry=2.99  //cm
8  // calculations
9  ex=2*rx^2/d
10 ey=2*ry^2/b
11 h=2*ex
12 w=2*ey
13 // results
14 printf('for steel height=%.3f cm and width=%.3f cm',
       h,w)
15 // ISHB 225
16 b=22.5  //cm
17 d=22.5  //cm
18 rx=9.8  //cm
19 ry=4.96  //cm
20 // calculations
21 ex=2*rx^2/d
22 ey=2*ry^2/b
23 h=2*ex
24 w=2*ey
25 // results
26 printf('\n for an ISHB height=%.3f cm and width=%.3f
       cm',h,w)
```

Scilab code Exa 9.4 Chapter9 Example 4

```
1  clc
2  //initialization of variables
3  clear
4  t=280  //kg/cm^2
5  c=840  //kg/cm^2
6  xbar=7.5  //cm from AB
7  A=210  //cm^2
8  // calculations
9  e=50+xbar  //cm
10 Iyy=7433  //cm^2
11 k=(1/210+e*xbar/Iyy)
12 P=t/k
13 k1=(-1/210+e*(xbar+5)/Iyy)
14 P1=c/k1
15 P_safe=min(P1,P)
16 // results
17 printf('The safe load is %d kg',P_safe)
```

Scilab code Exa 9.5 Chapter9 Example 5

```
1  clc
2  //initialization of the variables
3  clear
4  s=1.6  //m
5  s1=4  //m
6  pi=28  //degrees
7  w=16  //kg/m^2
8  p=100  //kg/m^2
9  pl=20  //cm
10 pb=10  //cm
11 r=500  //kg/m^3
12 // calculations
13 pi=pi*%pi/180  //radians
14 W=w*s+(r*pl*pb/(100*100))
15 P=p*s
```

```
16  L=P+W*cos(pi)
17  Mx=L*s1^2*100/8
18  sigma_1=Mx*6/(pb*pl^2)
19  My=W*sin(pi)*s1^2*100/8
20  sigma_2=My*6/(pl*pb^2)
21  sigma=sigma_1+sigma_2
22  // results
23  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

```
1  clc
2  //initialization of the problems
3  clear
4  s=1.6  //m
5  s1=4  //m
6  pi=28  //degrees
7  w=16  //kg/m^2
8  p=100  //kg/m^2
9  pl=20  //cm
10 pb=10  //cm
11 r=500  //kg/m^3
12 Zx=54.8  //cm^3
13 Zy=3.9  //cm^3
14 // calculations
15 pi=pi*%pi/180  //radians
16 W=w*s+8.1
17 P=p*s
18 L=P+W*cos(pi)
19 Mx=L*s1^2*100/8
20 sigma_1=Mx/Zx
21 My=W*sin(pi)*s1^2*100/8
```

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

Scilab code Exa 9.7 Chapter9 Example 7

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

Scilab code Exa 9.8 Chapter9 Example 8

```
1  clc
2  //initialization of variables
3  clear
4  P=3  //tonne/m
5  s=6  //m
6  l=50  //cm
7  b=20  //cm
8  k=0.5  //m
9  //calculations
10 R=P*s/2
11 sf=R-k*P
12 bm=R*k-P*k^2/2
13 tau_xy=1.5*sf*1000/(l*b)
14 tau_max=tau_xy
15 str=bm*s*10^5/(b*l*l)
16
17 // consider the line a-a
18
19 sigma_x=str*12.5/25
20 sigma_y=0
21 tau_xy=tau_xy*(1-(12.5/25)^2)
22
23 sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
24 sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
25
26 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)
27 printf('\n The principal stresses are %.2f kg/cm^2 (
```

```
     tension) %.2 f kg/cmˆ2 (compression) ',sigma_1,
     sigma_2)
28
29 //consider the line c-c
30 printf('\n For the line c-c the bending stress and
     shearing stress are \n respectively %.2 f kg/cmˆ2,
     %.2 f kg/cmˆ2 ',sigma_x,tau_xy)
31 printf('\n The principal stresses are %.2 f kg/cmˆ2 (
     compression) %.2 f kg/cmˆ2 (tension) ',sigma_2,
     sigma_1)
32
33 //for the line b-b
34 tau_xy=tau_max
35 sigma_x=0
36 sigma_y=0
37 sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
     sigma_y))ˆ2+tau_xyˆ2)
38 sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
     sigma_y))ˆ2+tau_xyˆ2)
39 // results
40 printf('\n For the line b-b the bending stress and
     shearing stress are \n respectively %.2 f kg/cmˆ2,
     %.2 f kg/cmˆ2 ',sigma_x,tau_xy)
41 printf('\n The principal stresses are %.2 f kg/cmˆ2 (
     tension) %.2 f kg/cmˆ2 (compression) ',sigma_1,
     sigma_2)
```

Scilab code Exa 9.9 Chapter9 Example 9

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

```
 7  b=20  //cm
 8  k=0.5  //m
 9  //calculations
10  R=P*s/2
11  sf=R-k*P
12  bm=R*k-P*k^2/2
13  tau_xy=1.5*sf*1000/(l*b)   //max shear stress
14  tau_max=tau_xy
15  str=bm*s*10^5/(b*l*l)   //max bending stress
16
17  // consider the line a-a
18
19  sigma_x=str*12.5/25
20  sigma_y=0
21  tau_xy=tau_xy*(1-(12.5/25)^2)
22
23  sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
24  sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
25
26  theta=1/2*atan(2*tau_xy/(sigma_x-sigma_y))
27  sigma_p=sigma_1/cos(theta)
28  P=sigma_p*2*l*b/(3*1000)
29  printf('A prestressing force of %.2f Tonne must be
       applied to balance the tension at a-a',P)
30
31  //At bottom point D or C
32  pre_str=P*2*1000/(l*b)
33  net=str-pre_str
34  printf('\n At bottom point D or C')
35  printf('\n Net tension = %.2f kg/cm^2 ',net)
36
37  //consider the line b-b
38  pre_str=P
39  sigma_x=pre_str
40  sigma_y=0
41  tau_xy=tau_max
```

```
42  sigma_1 =( sigma_x + sigma_y )/2+ sqrt ((1/2*( sigma_x -
        sigma_y ))^2+ tau_xy ^2)
43  sigma_2 =( sigma_x + sigma_y )/2- sqrt ((1/2*( sigma_x -
        sigma_y ))^2+ tau_xy ^2)
44  printf ('\n At section b-b ')
45  printf ('\n pre-stress=%.2 f  kg/cm^2 ', pre_str )
46  printf ('\n principal stresses are %.2 f, %.2 f  kg/cm^2
        ', sigma_1 , sigma_2 )
47
48  // for the line c-c
49  sigma_x = str *12.5/25
50  sigma_y =0
51  tau_xy = tau_xy *(1-(12.5/25)^2)
52  sigma_1 =( sigma_x + sigma_y )/2+ sqrt ((1/2*( sigma_x -
        sigma_y ))^2+ tau_xy ^2)
53  sigma_2 =( sigma_x + sigma_y )/2- sqrt ((1/2*( sigma_x -
        sigma_y ))^2+ tau_xy ^2)
54  pre_str = pre_str /2
55  net = sigma_1 + pre_str
56  sigma_x = net
57  sigma_y =0
58  sigma_1 =( sigma_x + sigma_y )/2+ sqrt ((1/2*( sigma_x -
        sigma_y ))^2+ tau_xy ^2)
59  sigma_2 =( sigma_x + sigma_y )/2- sqrt ((1/2*( sigma_x -
        sigma_y ))^2+ tau_xy ^2)
60  // results
61  printf ('\n At section c-c ')
62  printf ('\n the direct stress is %.2 f  kg/cm^2 ', net )
63  printf ('\n pre-stress = %.2 f  kg/cm^2 ', pre_str )
64  printf ('\n principal stresses are %.2 f, %.2 f  kg/cm^2
        ', sigma_1 , sigma_2 )
65
66  // wrong calculations in the thext for some parts
```

Scilab code Exa 9.10 Chapter9 Example 10

```
 1  clc
 2  //initialization of variables
 3  clear
 4  b=2  //cm
 5  h=2  //cm
 6  T=2000  //kg−cm
 7  V=250  //kg
 8  M=2000  //kg−cm
 9  // calculations
10  Mmax=M*6/(b*h*b)
11  Vmax=3*V/(2*b*h)
12  Zt=0.208*b^2*h
13  Tmax=T/(Zt)
14
15  sigma=Mmax
16  printf('points A,B,')
17  printf('\n sigma=%d kg/cm^2 (tension)',sigma)
18  printf('\n points C,D,')
19  printf('\n sigma=%d kg/cm^2 (cmpression)',sigma)
20  tau=Vmax+Tmax
21  printf('\n point E')
22  printf('\n tau=%.2f kg/cm^2 shear',tau)
23  tau=Vmax-Tmax
24  printf('\n tau=%.2f kg/cm^2 shear',tau)
25  // at G
26  sigma_x=sigma
27  sigma_y=0
28  tau_xy=Tmax
29  sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
        sigma_y))^2+tau_xy^2)
30  sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
        sigma_y))^2+tau_xy^2)
31  // results
32  printf('\n at point G')
33  printf('\n sigma_1 = %d kg/cm^2 (tension)',sigma_1)
34  printf('\n sigma_2 = %d kg/cm^2 (compression)',
        sigma_2)
35
```

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

Scilab code Exa 9.11 Chapter9 Example 11

```
1  clc
2  // initialization of variables
3  clear
4  w=10  //cm
5  s=2.8  //m
6  P=1  // tonne
7  Ft=1.4  //cm
8  Wt=0.8  //cm
9  Ix=13989.5  //cm^4
10 Z=699.5  //cm^3
11 // calculations
12 BM= 2.8
13 T=P*1000*8.21
14 SF=P*1000
15 BS=BM*10^5/(Z)
16 sigmaXA=BS*18.6/20
17 K=w*Ft*19.3+18.6*Wt*9.3
18 tau_xy_C=SF/(Ix*Wt)*K
19 tau_xy_A=tau_xy_C*(w*Ft*19.3)/K
20 tau_xy_B=tau_xy_A*0.5*Wt/w
21 sigmaXB=sigmaXA*19.3/20
22
23 tau_max=3*Ft*8210/(w*Ft^3+37.2*Wt^3)
24 tau_A=3*Wt*8210/(w*Ft^3+37.2*Wt^3)
25
26 //For point A
27 Shear=tau_xy_A-tau_A
28 sigma_x=sigmaXA
29 sigma_y=0
30 tau_xy=Shear
```

```
31  sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
32  sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
33
34  printf('For point A')
35  printf('\n Total shear= %.1f kg/cm^2 ',Shear)
36  printf('\n Bending stress = %d kg/cm^2 (Compr.)',
       sigma_x)
37  printf('\n Principal stresses are %d (tension), %d (
       comp.) kg/cm^2 ',sigma_1,sigma_2)
38
39  //For point B
40  printf('\n FOr point B')
41  printf('\n Bending shear stress is %.2f k/cm^2',
       tau_xy_B)
42  sigmaXB=BS*19.3/20
43  sigma_x=sigmaXB
44  sigma_y=0
45  tau_xy=tau_max
46  sigma_1=(sigma_x+sigma_y)/2+sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
47  sigma_2=(sigma_x+sigma_y)/2-sqrt((1/2*(sigma_x-
       sigma_y))^2+tau_xy^2)
48  printf('\n Principal stresses are %d (tension), %d (
       comp.) kg/cm^2 ',sigma_1,sigma_2)
49
50  // Answers in the text are approximations
```

Scilab code Exa 9.12 Chapter9 Example 12

```
1  clc
2  //initialization of variables
3  clear
4  b=10  //cm
```

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

Scilab code Exa 9.13 Chapter9 Example 13

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

```
15  Pcr=%pi^2*E*b*h^3/(12*L^2)
16  Pcr=Pcr/1000
17  Smax=-P*1000/(b*h)-(P*1000*1*5*12/10^4)*1/(1-(n*P/
        Pcr))
18  Smax=abs(Smax)
19
20  rr=b*h^3/(12*100)
21  Smax_se=P*1000/(b*h)*(1+e*5/rr*sec(%pi/2*sqrt(n*P/
        Pcr)))
22  Perror=(Smax-Smax_se)/Smax
23  Perror=Perror*100
24  Perror=abs(Perror)
25  // results
26  printf('Using secent formula, stress obtained is %d
        kg/cm^2',Smax_se)
27  printf('\n hence, the percentage error %.2f',Perror)
28  // approximate answees in the text
```

Scilab code Exa 9.14 Chapter9 Example 14

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

```
16  g = e *12.5*100/ rx ^2
17  Smax = F *1000/ A *(1+ g *1/(1 - n *( F / Pcr ) ) )
18  // results
19  printf ('The maximum stress developed is %d kg / cm ^2 ',
        Smax )
20
21  // approximate calculations involved in the text
        book
```

Scilab code Exa 9.15 Chapter9 Example 15

```
1   clc
2   // initialization of variables
3   clear
4   // linked to 9_14
5   // calculations
6   P =400 // kg / m
7   L =10 // m
8   F =10 // tonne
9   n =3
10  Ixx =5943.1 // cm ^4
11  A =52.03 // cm ^2
12  rx =10.69 // cm
13  E =2*10^6 // kg / cm ^2
14  Pcr =% pi ^2* E * Ixx /(( L *100)^2)
15  Pcr = Pcr /1000
16  e = P * L ^2/(8* F *1000)
17  g = e *12.5*100/ rx ^2
18  Smax = F *1000/ A *(1+ g *1/(1+ n *( F / Pcr ) ) )
19  // results
20  printf ('The maximum stress developed is %d kg / cm ^2 ',
        Smax )
21
22  // approximate answer in the text
```

Chapter 10

INTRODUCTION TO ENERGY METHODS

Scilab code Exa 10.3 Chapter10 Example 3

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

Scilab code Exa 10.4 Chapter10 Example 4

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

Scilab code Exa 10.7 Chapter10 Example 7

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

```
12  del=w*L*100/(A*E)
13  P=w*(1+sqrt(1+(2*h/del)))
14  printf('Stress in steeel = %d kg/cm^2 ',P)
15
16  // for cloth laminate
17  del=w*L*100/(A*Ec)
18  P=w*(1+sqrt(1+(2*h/del)))
19  printf('\n Stress in cloth laminate = %.1f kg/cm^2 '
        ,P)
```

Scilab code Exa 10.8 Chapter10 Example 8

```
1   clc
2   //initialization of variables
3   clear
4   w=64  //kg
5   H=60  //cm
6   b=40  //cm
7   h=5  //cm
8   E=0.12*10^6  //kg/cm^2
9   Es=2*10^6  //kg/cm^2
10  // for part (a) and (b)
11  I=b*h^3/12
12  del=4*w*120^3/(E*I)
13  P=w*(1+sqrt(1+(2*H/del)))
14  str=P*240*6/(b*h^2)
15  printf('part (a) and (b)')
16  printf('\n Maximum stress in wood = %d kg/cm^2',str)
17  printf('\n Max. force on divers feet = %d kg',P)
18
19  //for part (c)
20  Ixx=I*E/Es
21  Zxx=19.4  //cm^2
22  Ixx=72.7  //cm^4
23  del=4*w*120^3/(Es*Ixx)
```

```
24  P=w*(1+sqrt(1+(2*H/del)))
25  str=P*240/Zxx
26  // results
27  printf('\n part (c)')
28  printf('\n Maximum stress in steel = %d kg/cm^2',str
       )
29  printf('\n Max. force on divers feet = %d kg',P)
30  printf('\n Hence wood is better than steel')
31
32  // wrong calculations in some parts
```

Scilab code Exa 10.11 Chapter10 Example 11

```
1  clc
2  //initialization of variables
3  clear
4  A=100 //cm^2
5  E=2*10^6 //kg/cm^2
6  // calculations
7  del=1093.5*10^6/(E*A)
8  // 1093.5 from the table
9  // results
10 printf('Central deflection = %.2f mm',del)
```

Scilab code Exa 10.12 Chapter10 Example 12

```
1  clc
2  //initialization of variables
3  clear
4  T=30 //degree celcius
5  alpha=0.0000117 // per degree celcius
6  //AB
7  L=6 //m
```

```
 8  dl = T * alpha * L
 9  df = 0.375  // kg
10  tot = dl * df
11  // BC
12  dl = T * alpha * L
13  df = 0.375  // kg
14  tot = tot + dl * df
15  // CD
16  dl = T * alpha * L
17  df = 0.75  // kg
18  tot = tot + dl * df
19  tot = tot * 100 * 2
20  // results
21  printf ( 'The deflection is %.4 f cm',tot)
```