

Scilab Textbook Companion for  
Machine Design-i  
by Dr. Sadhu Singh<sup>1</sup>

Created by  
Rituraj  
B. Tech  
Computer Engineering  
Uktu  
College Teacher  
None  
Cross-Checked by  
None

July 30, 2019

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

# Book Description

**Title:** Machine Design-i

**Author:** Dr. Sadhu Singh

**Publisher:** Umesh Publications Delhi

**Edition:** 2

**Year:** 2012

**ISBN:** 978-93-80117-23-2

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

**Exa** Example (Solved example)

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

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

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

# Contents

List of Scilab Codes	4
1 Introduction	5
3 Design Against Static Load	9
4 Design Against Fluctuating Load	21
5 Riveted Joints	39
6 Shafts	50
7 Keys and couplings	59
8 Mechanical Springs	76
9 Power Screws	89

# List of Scilab Codes

Exa 1.1	Preferred Speeds . . . . .	5
Exa 1.2	Capacities of models . . . . .	5
Exa 1.4	Capacities of additional models . . . . .	6
Exa 1.6	Power capacity of models . . . . .	7
Exa 3.1	Dimension of cross section . . . . .	9
Exa 3.2	Dimension of cross section . . . . .	10
Exa 3.3	Dimension of cross section . . . . .	11
Exa 3.4	Dimension of cross section . . . . .	12
Exa 3.5	Suitable diameter of the shaft . . . . .	13
Exa 3.6	Diameter of the shaft . . . . .	15
Exa 3.8	Suitable diameter of the shaft . . . . .	15
Exa 3.10	Maximum value of torque . . . . .	16
Exa 3.11	Diameter of the shaft . . . . .	17
Exa 3.12	Cross section of hanger . . . . .	18
Exa 3.13	Diameter of the shaft . . . . .	19
Exa 4.1	Thickness of plate . . . . .	21
Exa 4.2	Diameter of the axle . . . . .	22
Exa 4.3	Endurance Limit . . . . .	23
Exa 4.4	Thickness of plate . . . . .	24
Exa 4.5	Endurance strength of specimen . . . . .	25
Exa 4.6	Diameter of the beam . . . . .	26
Exa 4.7	Diameter d at fillet cross section . . . . .	27
Exa 4.8	Diameter of the shaft . . . . .	28
Exa 4.9	Life of the spring . . . . .	29
Exa 4.10	factor of safety . . . . .	30
Exa 4.11	Diameter of the shaft . . . . .	31
Exa 4.12	Diameter of the shaft . . . . .	33
Exa 4.14	Minimum required ultimate strength . . . . .	34

Exa 4.16	Size of piston rod . . . . .	34
Exa 4.18	Suitable diameter of rod . . . . .	35
Exa 4.19	Thickness of plate . . . . .	36
Exa 4.20	Shaft size for infinite life . . . . .	37
Exa 4.21	Thickness of bar . . . . .	38
Exa 5.1	Design the longitudinal and circumferential joints . . . . .	39
Exa 5.2	Design an economical joint . . . . .	42
Exa 5.3	Diameter of rivets . . . . .	43
Exa 5.4	Design a double riveted butt joint . . . . .	44
Exa 5.6	Suitable diameter of rivets . . . . .	46
Exa 5.7	Diameter of rivets . . . . .	47
Exa 5.8	Diameter of rivets . . . . .	49
Exa 6.1	Shaft Diameter using AMSE code . . . . .	50
Exa 6.2	Diameter of the shaft . . . . .	51
Exa 6.3	Diameter of the shaft . . . . .	52
Exa 6.4	Shaft Diameter using AMSE code . . . . .	53
Exa 6.5	Diameter of the shaft . . . . .	54
Exa 6.6	Diameter of the shaft . . . . .	56
Exa 6.7	Diameter of the bar . . . . .	56
Exa 6.8	Required shaft diameter . . . . .	57
Exa 7.1	Design a protected type rigid cast iron flange . . . . .	59
Exa 7.2	Design a bushed pin type flexible coupling . . . . .	62
Exa 7.3	Length of hub . . . . .	65
Exa 7.4	Design the rectangular key . . . . .	66
Exa 7.6	Suitable dimensions for shaft . . . . .	67
Exa 7.7	Design the rectangular key . . . . .	69
Exa 7.8	Design the rectangular key . . . . .	70
Exa 7.9	Design a protected type rigid cast iron flange . . . . .	71
Exa 7.10	Diameter of bolts . . . . .	74
Exa 8.1	Design a close coiled helical compression spring . . . . .	76
Exa 8.2	Factor of safety . . . . .	78
Exa 8.3	Initial torsional shear stress . . . . .	79
Exa 8.4	Diameter of spring wire . . . . .	79
Exa 8.5	Design a spring for boiler safety . . . . .	81
Exa 8.7	Design the spring . . . . .	82
Exa 8.8	Design a spring for a balance . . . . .	84
Exa 8.10	Wire diameter of spring . . . . .	85

Exa 8.11	Design a helical spring . . . . .	86
Exa 8.12	Find the axial load . . . . .	87
Exa 9.1	Stress in the screw . . . . .	89
Exa 9.2	Power required to drive the screw . . . . .	90
Exa 9.3	Length of handle and shear stress in screw . . . . .	91
Exa 9.4	Power required to drive the slide . . . . .	92
Exa 9.5	Clamping force between jaws . . . . .	94
Exa 9.6	Design a screw jack . . . . .	95
Exa 9.7	Torque required to raise and lower the load . . . . .	98
Exa 9.9	Safe capacity of press . . . . .	99
Exa 9.11	Determine the force . . . . .	101
Exa 9.13	Design a screw jack . . . . .	102
Exa 9.14	Torque required to rotate the screw . . . . .	105
Exa 9.15	Design a simple screw jack . . . . .	107

# Chapter 1

## Introduction

Scilab code Exa 1.1 Preferred Speeds

```
1 // exa 1.1 Pg 13
2 clc;clear;close;
3 Nmax=1000;// rpm
4 Nmin=30;// rpm
5 z=9;// no. of steps
6
7 //Rn=Nmax/Nmin=fi**(z-1)
8 fi=(Nmax/Nmin)**(1/(z-1));// common ratio
9
10 printf('The speeds of gear box are:')
11 N1=Nmin;// rpm
12 for i=1:z
13     printf('\n\t\t\tN%d = %.1f rpm ',i,N1)
14     N1=fi*N1;//rpm
15 end;
```

---

Scilab code Exa 1.2 Capacities of models



```

1 // exa 1.2 Pg 14
2 clc;clear;close;
3 Pmax=100;// kW
4 Pmin=10;// kW
5 z=5;// no. of models
6
7 //Rn=Pmax/Pmin=fi**(z-1)
8 fi=(Pmax/Pmin)**(1/(z-1));// common ratio
9
10 printf('The power of generating sets are:')
11 P1=Pmin;// kW
12 for i=1:z
13     printf('\n\t\t\tP%d = %.1 f kW',i,P1)
14     P1=fi*P1;//kW
15 end;
16
17 printf('\nExpanding for 10 models. ');
18 z=10;// no. of models
19
20 fi=(Pmax/Pmin)**(1/(z-1));// common ratio
21
22 printf('\nThe power of generating sets are:')
23 P1=Pmin;// kW
24 for i=1:z
25     printf('\n\t\t\tP%d = %.1 f kW',i,P1)
26     P1=fi*P1;//kW
27 end;

```

---

#### Scilab code Exa 1.4 Capacities of additional models

```

1 // exa 1.4 Pg 15
2 clc;clear;close;
3 Pmax=50;// kW
4 Pmin=5;// kW
5 z=4;// no. of models

```

```

6
7 //Rn=Pmax/Pmin=fi**(z-1)
8 fi=(Pmax/Pmin)**(1/(z-1)); // common ratio
9
10 printf('The models are:')
11
12 for i=0:z-1
13     P1=fi**(i)*Pmin; // kW
14     printf('\n\t\t\tP%d = %.1 f kW',i,P1)
15 end;
16
17 printf('\n for 8 models.')
18
19 z=8; // no. of models
20
21 //Rn=Pmax/Pmin=fi**(z-1)
22 fi=(Pmax/Pmin)**(1/(z-1)); // common ratio
23
24 printf('The models are:')
25
26 for i=0:z-1
27     P1=fi**(i)*Pmin; // kW
28     printf('\n\t\t\tP%d = %.1 f kW',i,P1)
29 end;

```

---

### Scilab code Exa 1.6 Power capacity of models

```

1 // exa 1.6 Pg 15
2 clc;clear;close;
3 Pmax=75; // kW
4 Pmin=7.5; // kW
5 z=5; // no. of models
6
7 //Rn=Pmax/Pmin=fi**(z-1)
8 fi=(Pmax/Pmin)**(1/(z-1)); // common ratio

```

```
9
10 printf('The models are:')
11
12 for i=0:z-1
13     P1=fi**(i)*Pmin; // kW
14     printf('\n\t\t\tP%d = %.1 f kW',i,P1)
15 end;
```

---

# Chapter 3

## Design Against Static Load

Scilab code Exa 3.1 Dimension of cross section

```
1 // exa 3.1 Pg 62
2
3 clc;clear;close;
4
5 // Given Data
6 P=30;// kN
7 Sut=350;// MPa
8 n=2.5;// factor of safety
9
10 sigma_w=Sut/n;// MPa (Working stress for the link)
11
12 t=poly(0,'t');// thickness of link
13 A=2.5*t**2;// mm.sq.
14 I=t*(2.5*t)**3/12;// mm^4 (Moment of Inertia about N
    -A)
15 sigma_d=P/A;// N/mm.sq.
16 e=10+1.25*t;//mm
17 M=P*10**3*e;// N.mm
18 sigma_t=M*1.25*t/I;// N/mm.sq.
19 //maximum tensile stress at the top fibres = sigma_d
    +sigma_t=sigma_w ...eqn(1)
```

```

20 expr=sigma_d+sigma_t-sigma_w ;// expression of
    polynomial from above eqn.
21 t=roots( numer(expr)); // solving the equation (as
    denominator will be multiplied by zero on R.H.
    S)
22 t=t(1); // mm // discarding -ve roots
23 printf('dimension of cross section of link , t=%f mm
    . Adopt t=21 mm. ',t)

```

---

### Scilab code Exa 3.2 Dimension of cross section

```

1 // exa 3.2 Pg 63
2
3 clc;clear;close;
4
5 // Given Data
6 P=6; // kN
7 alfa=30; // degree
8 Sut=250; // MPa
9 n=2.5; // factor of safety
10
11 sigma_w=Sut/n; // MPa (Working stress for the link)
12 PH=P*10**3*cosd(alfa); // kN
13 PV=P*10**3*sind(alfa); // kN
14
15 t=poly(0, 't'); // thickness of link
16 A=2*t*t; // mm.sq.
17 sigma_d=PH/A; // N/mm.sq.
18 M=PH*100+PV*250; // N.mm
19 I=t*(2*t)**3/12; // mm^4 (Moment of Inertia)
20 sigma_t=M*t/I; // N/mm.sq.
21 //maximum tensile stress at the top fibres = sigma_d
    +sigma_t=sigma_w ... eqn(1)
22 expr=sigma_d+sigma_t-sigma_w ;// expression of
    polynomial from above eqn.

```

```

23 t=roots( numer(expr)); // solving the equation (as
    denominator will me be multiplied by zero on R.H.
    S)
24 t=t(1); // mm // discarding -ve roots
25 printf('dimension of cross section of link , t=%f mm
    . ',t)

```

---

### Scilab code Exa 3.3 Dimension of cross section

```

1 // exa 3.3 Pg 64
2
3 clc;clear;close;
4
5 // Given Data
6 P=20; // kN
7 Sut=300; // MPa
8 n=3; // factor of safety
9
10 sigma_w=Sut/n; // MPa (Working stress for the link)
11
12 t=poly(0, 't'); // thickness of link
13 A=4*t*t; // mm.sq.
14 sigma_d=P*10**3/A; // N/mm.sq.
15 e=6*t; //mm
16 M=P*10**3*e; // N.mm
17 z=t*(4*t)**2/6; // mm^3 (section modulus at x1-x2)
18 sigma_b=M/z; // N/mm.sq.
19 //maximum tensile stress at x1 = sigma_d+sigma_b=
    sigma_w ...eqn(1)
20 expr=sigma_d+sigma_b-sigma_w ; // expression of
    polynomial from above eqn.
21 t=roots( numer(expr)); // solving the equation (as
    denominator will me be multiplied by zero on R.H.
    S)
22 t=t(2); // mm // discarding -ve roots

```

```

23 printf('dimension of cross section of link , t=%0.2f
    mm. Use 23 mm. ',t)

```

---

#### Scilab code Exa 3.4 Dimension of cross section

```

1 // exa 3.4 Pg 65
2
3 clc;clear;close;
4
5 // Given Data
6 P=15; // kN
7 sigma_t=20; // MPa
8 sigma_c=60; // MPa
9 n=3; // factor of safety
10
11 a=poly(0,'a'); // from the diagram.
12 // Area of cross section
13 A1=2*a*a; // mm.sq.
14 A2=2*a*a/2; // mm.sq.
15 A=A1+A2; // mm.sq.
16
17 // Location of neutral axis
18 //3*a**2*y_bar=2*a**2*a/2+a**2*(a+a/2)
19 y_bar=(2*a**2*a/2+a**2*(a+a/2))/(3*a**2); // mm
20
21 // Moment of Inertia about neutral axis N-A
22 I=2*a*a**3/12+2*a**2*(y_bar-0.5*a)**2+2*((a/2)*(a
    **3/12)+(a**2/2)*(1.5*a-y_bar)**2); // mm^4
23 yt=y_bar; //mm
24 yc=2*a-y_bar; // mm
25 e=y_bar-0.5*a; //mm
26 M=P*10**3*e; // N.mm
27 sigma_d=P*10**3/A; // N/mm.sq.
28 sigma_t1=M*yt/I; //N/mm.sq.
29 sigma_c1=M*yc/I; //N/mm.sq.

```

```

30 sigma_r_t=sigma_d+sigma_t1;// N/mm.sq. (sigma_r_t=
    resultant tensile stress at AB=sigma_d+sigma_t)
31 sigma_r_c=sigma_c1-sigma_d;// N/mm.sq. (sigma_r_t=
    resultant tensile stress at AB=sigma_d+sigma_t)
32
33 //equating resulting tensile stress with given value
    sigma_t-sigma_r_t=0...eqn(1)
34 expr1=sigma_t-sigma_r_t;// expression of polynomial
    from above eqn.
35 a1=roots(numer(expr1));// solving the equation (as
    denominator will me be multiplied by zero on R.H.
    S)
36 a1=a1(2);// mm // discarding -ve roots
37 printf('Equating resultant tensile stress gives , a =
    %.2f mm',a1)
38
39 //equating resulting compressive stress with given
    value sigma_c-sigma_c_t=0...eqn(1)
40 expr2=sigma_c-sigma_r_c;// expression of polynomial
    from above eqn.
41 a2=roots(numer(expr2));// solving the equation (as
    denominator will me be multiplied by zero on R.H.
    S)
42 a2=a2(2);// mm // discarding -ve roots
43 printf('\n Equating resultant compressive stress
    gives , a = %.2f mm',a2)
44 a=ceil(a1);//mm
45 printf('\n dimension of cross section of link , a=%.2
    f mm. adopt a=%.f mm. ',a1,a)

```

---

**Scilab code Exa 3.5** Suitable diameter of the shaft

```

1 // exa 3.5 Pg 67
2
3 clc;clear;close;

```



```

4
5 // Given Data
6 Syt=760; // MPa
7 M=15; // kN.m
8 T=25; //kN.m
9 n=2.5; // factor of safety
10 E=200; // GPa
11 v=0.25; // Poisson 's ratio
12
13 sigma_d=Syt/n; // MPa
14 // let d is diameter of the shaft
15 sigma_b_into_d_cube=32*M*10**6/%pi; // N/mm.sq. (
    where sigma_b_into_d_cube = sigma_d*d**3)
16 tau_into_d_cube=16*T*10**6/%pi//d**3; // N/mm.sq. (
    where tau_into_d_cube = tau*d**3)
17 sigma1_into_d_cube=sigma_b_into_d_cube/2+1/2*sqrt(
    sigma_b_into_d_cube**2+4*tau_into_d_cube**2) ; //
    (where sigma1_into_d_cube=sigma1*d**3)
18 sigma2_into_d_cube=sigma_b_into_d_cube/2-1/2*sqrt(
    sigma_b_into_d_cube**2+4*tau_into_d_cube**2); //
    (where sigma2_into_d_cube=sigma2*d**3)
19 printf('\n (i) Maximum shear stress theory')
20 tau_max_into_d_cube=(sigma1_into_d_cube-
    sigma2_into_d_cube)/2; //(where
    tau_max_into_d_cube = tau_max*d**3)
21 d=(tau_max_into_d_cube/(sigma_d/2))**(1/3); //mm
22 printf('diameter of shaft , d=%0.1f mm or %0.f mm',d,
    ceil(d))
23
24 printf('\n (ii) Maximum strain energy theory')
25 //sigma1**2+sigma2**2-2*v*sigma1*sigma2=sigma_d**2
26 d=((sigma1_into_d_cube**2+sigma2_into_d_cube**2-2*v*
    sigma1_into_d_cube*sigma2_into_d_cube)/sigma_d
    **2)**(1/6)
27 printf('diameter of shaft , d=%0.1f mm',d)
28 printf('\n Adopt d=100mm')

```

---

### Scilab code Exa 3.6 Diameter of the shaft

```
1 // exa 3.6 Pg 69
2
3 clc;clear;close;
4
5 // Given Data
6 N=200; // rpm
7 P=200; // kW
8 tau_d=42; // Mpa
9 W=900; // N
10 L=3; // m
11 sigma_t=56; // MPa
12 sigma_c=56; // MPa
13
14 T=P*60*10**3/(2*pi*N); // N.m
15 M=W*L/4; // N.m
16 Te=sqrt(M**2+T**2); // N.m
17 //Te=(%pi/16)*d**3*tau_d
18 d=(Te/((%pi/16)*tau_d)*1000)**(1/3); // mm
19 printf('\n Using equivalent torque equation,\n shaft
        diameter d = %.f mm',d)
20
21 Me=(1/2)*(M+sqrt(M**2+T**2)); // N.m
22 //Me=(%pi/32)*d**3*sigma_d
23 d=(Me/((%pi/32)*sigma_c)*10**3)**(1/3); //mm
24 printf('\n Using equivalent bending moment equation
        ,\n shaft diameter d = %.2f mm or %.f mm',d, ceil
        (d))
25 printf('\n Adopt d=105 mm.')
```

---

### Scilab code Exa 3.8 Suitable diameter of the shaft

```

1 // exa 3.8 Pg 70
2
3 clc;clear;close;
4
5 // Given Data
6 M=15; // N.m
7 P=5; // kW
8 N=500; // rpm
9 tau_d=40; // Mpa
10 sigma_d=58; // MPa
11
12 T=P*60*10**3/(2*pi*N); // N.m
13 Te=sqrt(M**2+T**2); // N.m
14 //Te=(pi/16)*d**3*tau_d
15 d=(Te/((pi/16)*tau_d)*1000)**(1/3); // mm
16 printf('\n Using equivalent torque equation,\n shaft
        diameter d = %.f mm',d)
17
18 Me=(1/2)*(M+sqrt(M**2+T**2)); // N.m
19 //Me=(pi/32)*d**3*sigma_d
20 d=(Me/((pi/32)*sigma_d)*10**3)**(1/3); //mm
21 printf('\n Using equivalent bending moment equation
        ,\n shaft diameter d = %.2f mm or %.f mm',d, ceil
        (d))
22 printf('\n Adopt d=23 mm. ')

```

---

### Scilab code Exa 3.10 Maximum value of torque

```

1 // exa 3.10 Pg 71
2
3 clc;clear;close;
4
5 // Given Data
6 d=4; // cm
7 M=15000; // N.cm

```

```

8 Syt=20000; // N/cm.sq.
9
10 printf('\n (i) Maximum Principal Stress Theory-')
11 z=%pi*d**3/32; // cm.cube.
12 sigma_b=M/z; // N/cm.sq.
13 T=poly(0, 'T')
14 tau=16*T/(%pi*d**3); // N/cm.sq.
15 //sigma1=(1/2)*(sigma_b+sqrt(sigma_b**2+4*tau**2))
    // Maximum principal stress
16 //sigma1=(sigma_b/2+sqrt(sigma_b**2/4+tau**2)) // on
    solving
17 //tau=sqrt((sigma1-sigma_b/2)**2-sigma_b**2/4)
18 sigma1=Syt; // N/cm.sq.
19 T=sqrt((sigma1-sigma_b/2)**2-sigma_b**2/4)*(%pi*d
    **3)/16; // N.cm.
20 printf('\n Maximum value of torque , T = %.f N.cm.',T
    )
21
22 printf('\n (ii) Maximum Shear Stress Theory')
23 tau_d=0.5*Syt; // N.cm.
24 //Te=sqrt(M**2+T**2)=(%pi/16)*d**3*tau_d
25 T=sqrt(((%pi/16)*d**3*tau_d)**2-M**2); // N.cm.
26 printf('\n Maximum value of torque , T = %.f N.cm.',T
    )
27 // Answer in the textbook is not accurate.

```

---

### Scilab code Exa 3.11 Diameter of the shaft

```

1 // exa 3.11 Pg 72
2
3 clc;clear;close;
4
5 // Given Data
6 N=200; // rpm
7 P=25; // kW

```

```

8 tau_d=42; // MPa
9 W=900; // N
10 L=3; // m
11 Syt=56; // MPa
12 Syc=56; // MPa
13 sigma_d=56; // MPa
14
15 T=P*60*10**3/(2*pi*N); // N.m
16 M=W*L/4; // N.m
17 Te=sqrt(M**2+T**2); // N.m
18 // Te=(pi/16)*d**3*tau_d
19 d=(Te*10**3/((pi/16)*tau_d))**(1/3); // mm
20 printf('\n shaft diameter(using equivalent torque)-\n
        n d=%f mm.',d)
21
22 Me=(1/2)*(M+sqrt(M**2+T**2)); //N.m
23 // Me=(pi/32)*d**3*sigma_d
24 d=(Me*10**3/((pi/32)*sigma_d))**(1/3); // mm
25 printf('\n shaft diameter(using equivalent bending
        moment)-\n d=%f mm.',d)
26 printf('\n adopt d=57 mm.')

```

---

### Scilab code Exa 3.12 Cross section of hanger

```

1 // exa 3.12 Pg 72
2
3 clc;clear;close;
4
5 // Given Data
6 sigma_w=60; // MPa
7 F=10; // kN
8 alfa=30; // degree
9
10 FH=F*sind(alfa); // kN
11 FV=F*cosd(alfa); // kN

```

```

12 t=poly(0, 't'); // mm
13 A=t*t; // mm.sq.
14 sigma_d=FV*10**3/A
15 M=FV*10**3*120+FH*10**3*150; // N.mm
16 I=t*(2*t)**3/12; // mm^4
17 sigma_t=M*t/I; // N/mm.sq.
18 // Tensile stress at A=sigma_d+sigma_t=sigma_w ...
    eqn(1)
19 expr = sigma_d+sigma_t-sigma_w; // polynomial from
    above eqn.
20 t=roots( numer(expr)); // roots of the polynomial
21 t=t(1); // mm // discarding -ve roots
22 printf('\n value of t = %.1f mm',t)
23 A=2*t**2; // mm.sq.
24 printf('\n Area of cross-section of Hanger, A = %.f
    mm.sq.',A)
25 // Note-Answer in the textbook is slightly wrong and
    cross section not calculated.

```

---

### Scilab code Exa 3.13 Diameter of the shaft

```

1 // exa 3.13 Pg 74
2
3 clc;clear;close;
4
5 // Given Data
6 P=15; // kW
7 n1=200; // rpm
8 l=600; // mm
9 z2=18; // no. of teeth
10 m2=5; // mm
11 alfa2=14.5; // degree
12 l2=120; // mm
13 z1=72; // no. of teeth
14 m1=5; // mm

```

```

15  alfa1=14.5; // degree
16  l1=150; // mm
17  sigma_d=80; // MPa
18
19  d1=m1*z1; // mm
20  v1=%pi*d1*n1/(60*10**3); // m/s
21  Ft1=10**3*P/v1; // N (outwards)
22  Fr1=Ft1*tand(alfa1); // N (Downwards)
23  d2=m2*z2; // mm
24  v2=%pi*d2*n1/(60*10**3); // m/s
25  Ft2=10**3*P/v2; // N (outwards)
26  Fr2=Ft2*tand(alfa2); // N (Upwards)
27
28  // RAV*600=Fr1*450+Fr2*120 (Taking moments about
    bearing B)
29  RAV=(Fr1*450+Fr2*120)/600; // N (Downwards)
30  RBV=(Fr1-Fr2-RAV); // N (upwards)
31  MCV=RAV*l1; // N.mm
32  MBV=Fr2*l2; // N.mm
33
34  // RAH*600=-Ft1*450+Ft2*120 (Taking moments about
    bearing B)
35  RAH=(-Ft1*450+Ft2*120)/600; // N (Outwards)
36  RBH=Ft1+Ft2+RAH; // N (inwards)
37  MCH=RAH*l1; // N.mm
38  MBH=Ft2*l2; // N.mm
39
40  // Resultant Bending Moments
41  MC=sqrt(MCV**2+MCH**2); // N.mm
42  MB=sqrt(MBV**2+MBH**2); // N.mm
43  Mmax=max(MC,MB); // N.mm
44  T=10**3*P/(2*%pi*n1); // N.m
45  Me=(1/2)*(Mmax+sqrt(Mmax**2+T**2)); // N.mm
46  // Me=(%pi/32)*d**3*sigma_d
47  d=(Me/((%pi/32)*sigma_d))**(1/3)
48  printf('\n shaft diameter is : %.f mm',d)

```

---

# Chapter 4

## Design Against Fluctuating Load

Scilab code Exa 4.1 Thickness of plate

```
1 // exa 4.1 Pg 102
2 clc;clear;close;
3 P=6; // kN
4
5 //dimensions of plate
6 r=5; //mm
7 d=40; //mm
8 D=50; //mm
9 d0=10; //mm
10 w=40; //mm
11 Sut=200; //MPa
12 n=2.5; // factor of safety
13
14 // Fillet -
15 rBYd=r/d;
16 DBYd=D/d;
17 Kt=1.75; // factor
18 printf('for stepped plate under tension , Kt=%0.2f for
        r/d = %0.3f & D/d = %0.2f ',Kt,rBYd,DBYd)
```



```

19 t=poly(0, 't')
20 sigma_max = Kt*P/t; // N per mm sq.
21
22 // Hole -
23 d0BYw=d0/w;
24 Kt=2.42; // factor
25 printf('\n for finite width plate under tension with
        a hole, Kt=%.2f for d0/w = %.2f ',Kt,d0BYw)
26 sigma_max_into_t = Kt*P/(w-d0); //N/mm sq.
27
28 //Design stress
29 sigma_d = Sut/n; // MPa
30 //putting sigma_max=sigma_d
31 t=sigma_max_into_t/sigma_d*1000; // mm
32 printf('\n Thickness of plate = %.2f mm or %.f mm',t
        ,t)

```

---

#### Scilab code Exa 4.2 Diameter of the axle

```

1 // exa 4.2 Pg 104
2 clc;clear;close;
3
4 // Given Data
5 rBYd=0.1;
6 DBYd=1.2;
7 P=3; // kN
8 Syt=300; //MPa
9 n=3; // factor of safety
10 //dimensions of plate
11 l1=400; //mm
12 l2=300; //mm
13 l3=400; //mm
14
15
16 sigma_d=Syt/n; // MPa

```

```

17 Kt=1.65;// factor for circular fillet radius member
18 Rp=P/2;//kN (bearing reaction due to symmetry)
19 Mf=Rp*l1;// kN.mm (bending moment at fillet)
20 Mc=P*(l1+l2+l3)/4;// kN.mm (bending moment at centre
    )
21
22 // Fillet
23 //sigma_max=Kt*32*Mf/(%pi*d**3)
24 sigma_max_into_d_cube_1 = Kt*32*Mf*1000/%pi
25
26
27 // Centre
28 //sigma_max=32*Mc/(%pi*d**3)
29 sigma_max_into_d_cube_2 = Kt*32*Mf*1000/%pi
30 sigma_max_into_d_cube=max(sigma_max_into_d_cube_1,
    sigma_max_into_d_cube_2);// (getting max)
31
32 //putting sigma_max=sigma_d
33 t=(sigma_max_into_d_cube/sigma_d)**(1/3);// mm
34 printf('\n Diameter of axle = %.1f mm',t)

```

---

#### Scilab code Exa 4.3 Endurance Limit

```

1 // exa 4.3 Pg 105
2 clc;clear;close;
3
4 // Given Data
5 Sut=440;//MPa
6 d=25;//mm
7 R=95/100;// reliability
8 Kt=1.8;// stress concentration factor
9 q=0.86;// sensitivity factor
10
11 Se_dash = 0.5*Sut;// MPa
12

```

```

13 // for machined surface
14 ka=0.82; // surface finish factor
15 kb=0.85; // size factor
16 kc=0.868; // reliability factor
17 kd=1; // temperature factor
18 ke=0.577; // load factor
19
20 Kf=1+q*(Kt-1); // fatigue strength factor
21 kf=1/Kf ; // fatigue strength reduction factor
22 Se=ka*kb*kc*kd*ke*kf*Se_dash; // (MPa) Endurance
    limit
23 printf('\n Endurance limit = %.2f MPa',Se)

```

---

#### Scilab code Exa 4.4 Thickness of plate

```

1 // exa 4.4 Pg 105
2 clc;clear;close;
3
4 // Given Data
5 Sut=440; //MPa
6 w=60; //mm
7 d=12; // mm
8 P=20; // kN
9 q=0.8; // sensitivity factor
10 R=90/100; // reliability
11 n=2; // factor of safety
12
13 Kt=2.52; // stress concentration factor
14 Se_dash = 0.5*Sut; // MPa
15 // for hot rolled normalized condition
16 ka=0.67; // surface finish factor
17 kb=0.85; // size factor (assuming t<50 mm)
18 kc=0.897; // reliability factor
19 kd=1; // temperature factor
20 ke=0.9; // load factor

```

```

21 dBYw=d/w; //(for circular hole)
22
23 Kf=1+q*(Kt-1); // fatigue strength factor
24 kf=1/Kf ; // fatigue strength reduction factor
25 Se=ka*kb*kc*kd*ke*kf*Se_dash; // (MPa) Endurance
    limit
26 sigma_d=Se/n; // MPa (design stress)
27 // sigma_max=P/(w-d)/t
28 sigma_max_into_t = P*1000/(w-d);
29 // putting sigma_max=sigma_d
30 t=sigma_max_into_t/sigma_d; // mm
31 printf('\n Thickness of plate = %.2f mm or 20 mm',t)

```

---

#### Scilab code Exa 4.5 Endurance strength of specimen

```

1 // exa 4.5 Pg 107
2 clc;clear;close;
3
4 // Given Data
5 Sut=650; //MPa
6 N=10**5; // cycles
7 Se_dash = 0.5*Sut; // MPa
8 of=5; // unit
9 ob=6; //unit
10 bf=ob-of; // unit
11 be=3; //unit
12
13 // calculating endurance section wise
14 OE=log10(Se_dash);
15 OA=log10(0.9*Sut);
16 AE=OA-OE;
17 //log10_Sf=OD=OE+ED=OE+FC
18 log10_Sf=OE+(bf/be)*AE;
19 Sf=10**log10_Sf; // (MPa) Endurance
20 printf('\n Endurance of specimen = %.2f MPa',Sf)

```

---

Scilab code Exa 4.6 Diameter of the beam

```
1 // exa 4.6 Pg 108
2 clc;clear;close;
3
4 // Given Data
5 Sut=540;//MPa
6 N=10**4;// cycles
7 q=0.85;// sensitivity factor
8 R=90/100;// reliability
9 P=1500;// N
10 l=160;// mm
11
12 Se_dash = 0.5*Sut;// MPa
13 // for cold drawn steel
14 ka=0.79;// surface finish factor
15 kb=0.85;// size factor (assuming t<50 mm)
16 kc=0.897;// reliability factor
17 kd=1;// temperature factor
18 ke=1;// load factor
19
20 Kt=1.33;// under bending
21
22 Kf=1+q*(Kt-1);// fatigue strength factor
23 kf=1/Kf ;// fatigue strength reduction factor
24 Se=ka*kb*kc*kd*ke*kf*Se_dash;// MPa( Endurance limit
    )
25
26 of=4;// unit
27 ob=6;//unit
28 bf=ob-of;// unit
29 be=3;//unit
30
31 // calculating endurance section wise
```

```

32 OE=log10(Se);
33 OA=log10(0.9*Sut);
34 AE=OA-OE;
35 //log10_Sf=OD=OE+ED=OE+FC
36 log10_Sf=OE+(bf/be)*AE;
37 Sf=10**log10_Sf; // (MPa) Endurance
38
39 MB=P*1; // N.mm
40 // 32*MB/%pi/d**3 = Sf
41 d=(32*MB/%pi/Sf)**(1/3)
42 printf('\n diameter of beam %.f mm',d)

```

---

#### Scilab code Exa 4.7 Diameter d at fillet cross section

```

1 // exa 4.7 Pg 110
2 clc;clear;close;
3
4 // Given Data
5 Sut=600; //MPa
6 Syt=380; //MPa
7 q=0.9; // sensitivity factor
8 R=90/100; // reliability
9 n=2; // factor of safety
10 Pmin=-100; // N
11 Pmax=200; // N
12 l=150; // mm
13
14 Se_dash = 0.5*Sut; // MPa
15 // for cold drawn steel
16 ka=0.76; // surface finish factor
17 kb=0.85; // size factor (assuming t<50 mm)
18 kc=0.897; // reliability factor
19 kd=1; // temperature factor
20 ke=1; // load factor
21

```

```

22 Kt=1.4; // under bending
23
24 Kf=1+q*(Kt-1); // fatigue strength factor
25 kf=1/Kf ; // fatigue strength reduction factor
26 Se=ka*kb*kc*kd*ke*kf*Se_dash; // MPa( Endurance limit
    )
27 Mmax=Pmax*l; // N.mm
28 Mmin=Pmin*l; // N.mm
29 Mm=(Mmax+Mmin)/2; // N.mm
30 Ma=(Mmax-Mmin)/2; // N.mm
31 theta=atand(Ma/Mm); // degree
32
33 //equation of Goodman - sigma_m/Sut+sigma_a/Se=1
34 //here sigma_a/sigma_m=3
35 sigma_m=1/(1/Sut+3/Se); //MPa
36 sigma_a=3*sigma_m; // MPa
37
38 sigma_da=sigma_a/n; // MPa
39 //sigma_da=32*Ma/%pi/d**3
40 d=(32*Ma/%pi/sigma_da)**(1/3); // mm
41 printf('\n diameter d at fillet cross section = %.f
    mm', d)

```

---

#### Scilab code Exa 4.8 Diameter of the shaft

```

1 // exa 4.8 Pg 112
2 clc;clear;close;
3
4 // Given Data
5 Sut=500; //MPa
6 Syt=300; //MPa
7 R=90/100; // reliability
8 n=2; // factor of safety
9 Tmin=-200; // N.m
10 Tmax=500; // N.m

```

```

11
12 Se_dash = 0.5*Sut; // MPa
13 // for cold drawn steel
14 ka=0.80; // surface finish factor
15 kb=0.85; // size factor (assuming t<50 mm)
16 kc=0.897; // reliability factor
17 kd=1; // temperature factor
18 ke=0.577; // load factor
19
20 Ses=ka*kb*kc*kd*ke*Se_dash; // MPa( Endurance limit)
21 Sys=ke*Syt; // MPa
22 Tm=(Tmax+Tmin)/2; // N.m
23 Ta=(Tmax-Tmin)/2; // N.m
24 theta=atand(Ta/Tm); // degree
25 Sms=Ses/tand(theta); //MPa
26 Sas=Ses; //MPa
27 tau_da=Sas/n; //MPa
28 // tua_da=16*Ta/%pi/d**3
29 d=(16*Ta*1000/%pi/tau_da)**(1/3); //mm
30 printf('\n diameter of shaft = %.f mm',d)

```

---

#### Scilab code Exa 4.9 Life of the spring

```

1 // exa 4.9 Pg 113
2 clc;clear;close;
3
4 // Given Data
5 Sut=860; //MPa
6 Syt=690; //MPa
7 Pmin=60; // N
8 Pmax=120; // N
9 R=50/100; // reliability
10 l=500; //mm
11 d=10; //mm
12 Se_dash = 0.5*Sut; // MPa

```



```

13 // for machines surface
14 ka=0.70; // surface finish factor
15 kb=0.85; // size factor (assuming t<50 mm)
16 kc=1; // reliability factor
17 kd=1; // temperature factor
18 ke=1; // load factor
19
20 Se=ka*kb*kc*kd*ke*Se_dash; // MPa( Endurance limit)
21 Mmax=Pmax*1; // N.mm
22 Mmin=Pmin*1; // N.mm
23 Mm=(Mmax+Mmin)/2; // N.mm
24 Ma=(Mmax-Mmin)/2; // N.mm
25 Sm=32*Mm/%pi/d**3; //MPa
26 sigma_m=Sm; //MPa
27 Sa=32*Ma/%pi/d**3; //MPa
28 sigma_a=Sa; //MPa
29 Sf=Sa*Sut/(Sut-Sm); //MPa
30
31 //calculating section
32 OB=6; //unit ref. o at 3
33 BE=OB-3; //unit
34 OC=Sf; // MPa
35 AE=log10(0.9*Sut)-log10(Se); //MPa
36 AC=log10(0.9*Sut)-log10(Sf); //MPa
37 CD=BE*AC/AE; //
38 //log10(N)=3+CD
39 N=10**(3+CD); // cycle
40 printf('\n life of the spring , N = %.f cycles ',N)
41 //Note : answer in the textbook is wrong.

```

---

#### Scilab code Exa 4.10 factor of safety

```

1 // exa 4.10 Pg 116
2 clc;clear;close;
3

```

```

4 // Given Data
5 Sut=600; //MPa
6 Se=280; //MPa
7 sigma_x_min=50; // MPa
8 sigma_x_max=100; // MPa
9 sigma_y_min=20; // MPa
10 sigma_y_max=70; // MPa
11
12 sigma_xm=(sigma_x_max+sigma_x_min)/2; // MPa
13 sigma_xa=(sigma_x_max-sigma_x_min)/2; // MPa
14 sigma_ym=(sigma_y_max+sigma_y_min)/2; // MPa
15 sigma_ya=(sigma_y_max-sigma_y_min)/2; // MPa
16
17 // distortion energy theory -
18 sigma_m=sqrt(sigma_xm**2+sigma_ym**2-sigma_xm*
    sigma_ym); // MPa
19 sigma_a=sqrt(sigma_xa**2+sigma_ya**2-sigma_xa*
    sigma_ya); // MPa
20 theta=atand(sigma_a/sigma_m); // degree
21 // Sm/Sut+Sa/Se=1 where Sa=Sm*tan(theta)
22 Sm=1/(1/Sut+tand(theta)/Se); // MPa
23 Sa=tand(theta)*Sm; // MPa
24 n=Sa/sigma_a; // factor of safety
25
26 printf('\n factor of safety , n = %.2f',n)

```

---

#### Scilab code Exa 4.11 Diameter of the shaft

```

1 // exa 4.11 Pg 117
2 clc;clear;close;
3
4 // Given Data
5 Sut=600; //MPa
6 Syt=400; //MPa
7 Se=200; //MPa

```

```

8 Mmin=200; // N.m
9 Mmax=500; // N.m
10 Tmin=60; // N.m
11 Tmax=180; // N.m
12 n=2; // factor of safety
13
14 Mm=(Mmax+Mmin)/2; // N.mm
15 Ma=(Mmax-Mmin)/2; // N.mm
16 Tm=(Tmax+Tmin)/2; // N.mm
17 Ta=(Tmax-Tmin)/2; // N.mm
18 // sigma_xm=32*Mm/%pi/d**3
19 sigma_xm_into_d_cube=(32*Mm*1000)/%pi;
20 // sigma_xa=32*Ma/%pi/d**3
21 sigma_xa_into_d_cube=(32*Ma*1000)/%pi;
22 //Txy=16*Tm/%pi/d**3
23 Txy_into_d_cube=16*Tm*1000/%pi;
24 //Txya=16*Ta/%pi/d**3
25 Txya_into_d_cube=16*Ta*1000/%pi;
26 // sigma_m=sqrt(sigma_xm**2+3*Txy**2)
27 sigma_m_dash=sqrt(sigma_xm_into_d_cube**2+3*
    Txy_into_d_cube**2); // taken sigma_m_dash =
    sigma_m*d**(-3) for calculation
28 // sigma_a=sqrt(sigma_xa**2+3*Txya**2)
29 sigma_a_dash=sqrt(sigma_xa_into_d_cube**2+3*
    Txya_into_d_cube**2); // taken sigma_a_dash =
    sigma_a*d**(-3) for calculation
30 //tan(theta) = sigma_a/sigma_m
31 theta = atan(sigma_a_dash/sigma_m_dash); // radian
32 //Sm/Sut+Sa/Se= 1 where Sa/Sm=0.4348
33 Sm= 1/(1/Sut+0.4348/Se); // MPa
34 Sa=0.4348 * Sm; // MPa
35 //sigma_a=Sa/n
36 d=(Sa/n/sigma_a_dash)**(1/3)*1000; // mm
37 printf('\n diameter of shaft , d = %.2f mm',d)
38 // Note - Ans in the textbook is wrong.

```

---

#### Scilab code Exa 4.12 Diameter of the shaft

```
1 // exa 4.12 Pg 119
2 clc;clear;close;
3
4 // Given Data
5 Sut=620;//MPa
6 Syt=380;//MPa
7 R=90/100;// Reliability
8 n=2.5;// factor of safety
9 Tmin=-200;// N.m
10 Tmax=400;// N.m
11
12 Se_dash=0.5*Sut;//MPa
13 // for ground shaft
14 ka=0.92;// surface finish factor
15 kb=0.85;// size factor (assuming t<50 mm)
16 kc=0.897;// reliability factor
17 kd=1;// temperature factor
18 ke=0.577;// load factor
19 Ses=ka*kb*kc*kd*ke*Se_dash;// MPa( Endurance limit)
20 Sys=ke*Syt;// MPa
21 Tm=(Tmax+Tmin)/2;// N.mm
22 Ta=(Tmax-Tmin)/2;// N.mm
23 theta=atan(Ta/Tm);//radian
24 Sas=Ses;// MPa
25 Sms=Sas/3;// MPa
26 //Tda=Sas/n=16*Ta/%pi/d**3
27 d=(16*Ta*1000/%pi/(Sas/n))**(1/3);// mm
28 printf('\n diameter of shaft , d = %.2f mm or %d mm',
    d, ceil(d))
```

---

#### Scilab code Exa 4.14 Minimum required ultimate strength

```
1 // exa 4.14 Pg 121
2 clc;clear;close;
3
4 // Given Data
5 sigma_max=300; // MPa
6 sigma_min=-150; // MPa
7 n=1.5; // factor of safety
8
9
10 sigma_m=(sigma_max+sigma_min)/2; // MPa
11 sigma_a=(sigma_max-sigma_min)/2; // MPa
12 // Goodman failure line - sigma_m/Sut+sigma_a/Se=1/n
13 Sut=(sigma_m+sigma_a/(0.5))*n ; // putted Se=0.5*Sut
14 printf('\n Minimum required ultimate strength , Sut =
        %.1f MPa',Sut)
```

---

#### Scilab code Exa 4.16 Size of piston rod

```
1 // exa 4.16 Pg 122
2 clc;clear;close;
3
4 // Given Data
5 Pmin=-15; // kN
6 Pmax=25; // kN
7 Se_dash=360; // MPa
8 Sy=400; // MPa
9 Ki=1.25; // impact factor
10 n=2.25; // factor of safety
11 ka=0.88; // surface finish factor
12 Kt=2.25; // stress concentration factor
13 Pm=(Pmax+Pmin)/2; // kN
14 Pa=(Pmax-Pmin)/2; // kN
15 q=0.8; // sensitivity factor
```

```

16
17 // sigma_m=4*Pm/%pi/d**2
18 sigma_m_into_d_sq = 4*Pm*1000/%pi;
19 sigma_a_into_d_sq = 4*Pa*1000/%pi;
20 Kf=1+q*(Kt-1); // fatigue strength factor
21 kf=1/Kf ; // fatigue strength reduction factor
22 kb=0.85; // size factor
23 ke=0.9; //load factor
24 ki=1/Ki; // reverse impact factor
25 Se=ka*kb*ke*kf*ki*Se_dash; // MPa
26 //soderburg failure equation - sigma_m/Sy+sigma_a/Se
    =1/n
27 d=sqrt((sigma_m_into_d_sq/Sy+sigma_a_into_d_sq/Se)*n
    )
28 printf('\n Size of piston rod, d = %.f mm',d)

```

---

#### Scilab code Exa 4.18 Suitable diameter of rod

```

1 // exa 4.18 Pg 123
2 clc;clear;close;
3
4 // Given Data
5 Pmin=-300; // kN
6 Pmax=700; // kN
7 Se_dash=280; // MPa
8 Sy=350; // MPa
9 Kf=1.8; //fatigue strength factor
10 n=2; // factor of safety
11
12 Pm=(Pmax+Pmin)/2; // kN
13 Pa=(Pmax-Pmin)/2; // kN
14 // sigma_m=4*Pm/%pi/d**2
15 sigma_m_into_d_sq = 4*Pm*1000/%pi;
16 sigma_a_into_d_sq = 4*Pa*1000/%pi;
17 kf=1/Kf ; // fatigue strength reduction factor

```

```

18 kb=0.85; // size factor
19 ke=0.9; //load factor
20 ka=0.93; // surface finish factor
21 Se=ka*kb*ke*kf*Se_dash; // MPa
22 //Goodman failure equation –  $\sigma_m/S_y + \sigma_a/S_e$ 
    =1/n
23 d=sqrt((sigma_m_into_d_sq/S_y+sigma_a_into_d_sq/Se)
    *2.25)
24 printf('\n Suitable diameter of rod , d = %.f mm',d)
25 // Note – Ans in the textbook is wrong.

```

---

#### Scilab code Exa 4.19 Thickness of plate

```

1 // exa 4.19 Pg 124
2 clc;clear;close;
3
4 // Given Data
5 w=110; // mm
6 Pmin=98.1; // kN
7 Pmax=250; // kN
8 Se=225; // N/mm. sq
9 Sy=300; // N/mm. sq
10 n=1.5; // factor of safety
11
12 Pm=(Pmax+Pmin)/2; // kN
13 Pa=(Pmax-Pmin)/2; // kN
14 //  $\sigma_m = P_m/w/t$ 
15 sigma_m_into_t = Pm/w;
16 sigma_a_into_t = Pa/w;
17 //Soderburg failure equation –  $\sigma_m/S_y + \sigma_a/S_e$ 
    =1/n
18 d=(sigma_m_into_t/S_y+sigma_a_into_t/Se)*n*1000; // mm
19 printf('\n thickness of plate , t = %.1f mm',d)

```

---

#### Scilab code Exa 4.20 Shaft size for infinite life

```
1 // exa 4.20 Pg 124
2 clc;clear;close;
3
4 // Given Data
5 Mmin=200; // kN.mm
6 Mmax=600; // kN.mm
7 Tmin=60; // kN
8 Tmax=180; // kN
9 Su=550; // MPa
10 Sy=400; // MPa
11 Se=0.5*Su; // MPa
12 n=1.5; // factor of safety
13 Ktb=1.5; // stress concentration factor in bending
14 Kts=1.2; // stress concentration factor in torsion
15
16 Mm=(Mmax+Mmin)/2; // kN.mm
17 Ma=(Mmax-Mmin)/2; // kN.mm
18
19 //sigma_xm=32*Mm/%pi/d**3
20 sigma_xm_into_d_cube=32*Mm/%pi;
21 //sigma_xa=32*Ma/%pi/d**3
22 sigma_xa_into_d_cube=32*Ma/%pi;
23 Tm=(Tmax+Tmin)/2; // kN.mm
24 Ta=(Tmax-Tmin)/2; // kN.mm
25 Txym_into_d_cube=16*Tm/%pi;
26 Txya_into_d_cube=16*Ta/%pi;
27 // using distortion energy theory
28 // sigma_m=sqrt(sigma_xm**2+3*Txym**2)
29 sigma_m_into_d_cube=sqrt(sigma_xm_into_d_cube**2+3*
    Txym_into_d_cube**2);
30 // sigma_a=sqrt((Ktb*sigma_xa)**2+3*(Kts*Txym)**2)
31 sigma_a_into_d_cube=sqrt((Ktb*sigma_xa_into_d_cube)
```



```

    **2+3*(Kts*Txya_into_d_cube)**2);
32 // Sodeburg equation - sigma_m + (Su/Se)*sigma_a=Sy/
    n
33 d=((sigma_m_into_d_cube + (Su/Se)*
    sigma_a_into_d_cube)*1000/(Sy/n))**(1/3)
34 printf('\n shaft size , d = %.f mm',d)

```

---

#### Scilab code Exa 4.21 Thickness of bar

```

1 // exa 4.21 Pg 126
2 clc;clear;close;
3
4 // Given Data
5 // Hole -
6 d=25; //mm
7 w=150; //mm
8 Kt=2.56; // stress concentration factor
9 P=50; // kN
10 sigma_max=100; // N/mm.sq
11 t=Kt*P*1000/(w-d)/sigma_max; // mm
12 printf('Calculating for hole - \n thickness is : %.2
    f mm',t)
13
14 // Notch -
15 d=30; //mm
16 w=120; //mm
17 w=150; //mm
18 Kt=2.3; // stress concentration factor
19 P=50; // kN
20 sigma_max=100; // N/mm.sq
21 t=Kt*P*1000/(w-d)/sigma_max; // mm
22 printf('\n Calculating for notch - \n thickness is :
    %.2 f mm',t)
23 disp('Suggestion , Adopt t = 11 mm')

```

---

# Chapter 5

## Riveted Joints

Scilab code Exa 5.1 Design the longitudinal and circumferential joints

```
1 // exa 5.1 Pg 142
2 clc;clear;close;
3
4 // Given Data
5 ps=2.5;// MPa
6 D=1.5;//m
7 sigma_t=80;// MPa
8 tau=60;// MPa
9 sigma_c=120;// MPa
10 n=5;// no. of rivets
11
12 printf('DESIGNING LONGITUDINAL JOINT - \n')
13 printf('\n Plate Thickness ')
14 eta_1=80;// % (efficiency)
15 t = ps*D*1000/2/sigma_t/(eta_1/100)+1;// mm
16 printf(', t = %.2 f mm',t)
17 t=32;//mm (adopted for design)
18 printf('\n use t = %d mm',t)
19 printf('\n Diameter of rivet hole , do = ')
20 d0=6*sqrt(t);//mm (for t>8 mm)
21 printf('%.2 f mm',d0)
```

```

22 d0=34.5; // suggested for design
23 printf('\n Use do = %.f mm',d0)
24 printf('\n Diameter of rivet , d = ')
25 d=d0-1.5; //mm
26 printf('%.2 f mm',d)
27 printf('\n Pitch of rivets , p = ')
28 Ps=(4*1.875+1)*%pi/4*d0**2*tau; // N
29 // Putting Pt=Ps where Pt=(p-d0)*t*sigma_t; // N
30 Pt=Ps; //N
31 p=Pt/(t*sigma_t)+d0; // N
32 printf('%.1 f mm',p)
33 C=6; // for 5 no. of rivets
34 pmax=C*t+40; // mm (as per IBR)
35 printf('\n as per IBR-\n pitch , pmax = %.f mm',pmax)
36 p=220; // mm (adopted for design)
37 printf('\n Use p = %.f mm',p)
38 pi=p/2; // mm
39 printf('\n pitch of rivets in inner row, pi = %.f mm
    ',pi)
40
41 //Distance between rows of rivets
42 dis1=0.2*p+1.115*d0; // mm
43 printf('\n distance between outer and adjacent row =
    %.1 f mm',dis1)
44 dis1=85; //mm (adopted for design)
45 printf('\n take & use this distance = %.f mm', dis1)
46 dis2=0.165*p+0.67*d0; // mm
47 printf('\n distance between inner row for zig-zag
    riveting = %.1 f mm', dis2)
48 dis2=60; //mm (adopted for design)
49 printf('\n take & use this distance = %.f mm', dis2)
50 printf('\n Thickness of wide butt strap , t= ')
51 t1=0.75*t; // mm (wide butt strap)
52 printf(' %.f mm',t1)
53 t2=0.625*t; // mm (narrow butt strap)
54 printf('\n Thickness of narrow butt strap , t= %.f mm
    ',t2)
55 //margin

```

```

56 m=ceil(1.5*d0); // mm
57 printf('\n margin , m = %.f mm',m)
58 // Efficiency of joint
59 Pt=(p-d0)*t*sigma_t; // N
60 Ps=Ps; // N (shearing resistance of rivets)
61 Pc=n*d0*t*sigma_c; // N (crushing resistance of
    rivets)
62 sigma_com = (p-2*d0)*t*sigma_t+%pi/4*d0**2*tau; // N
63 printf('\n strength of the joint = %d N',sigma_com)
64 P=p*t*sigma_t; //N (strength of solid plate)
65 printf('\n strength of solid plate = %d N',P)
66 eta_l=sigma_com/P*100; // % (efficiency)
67 printf('\n Efficiency of joint , eta_l = %.1f %%',
    eta_l)
68
69 printf('\n\n DESIGNING CIRCUMFERENTIAL JOINT- \n')
70 t=32; // mm
71 d0=34.5; //mm
72 d=33; //mm
73 printf('\n Plate Thickness ')
74 printf(', t = %.2f mm',t)
75 printf('\n Diameter of rivet hole , do = ')
76 printf('%0.2f mm',d0)
77 printf('\n Diameter of rivet , d = ')
78 printf('%0.2f mm',d)
79 n=(D*1000/d0)**2*(ps/tau); // no. of rivets
80 printf('\n no. of rivets = %.1f',n)
81 n=80; // adopted for design
82 printf('\n take n = %d',n)
83 // Pitch of rivets
84 n1=n/2; // no. of rivets per row
85 pc=%pi*(D*1000+t)/n1; // mm (pitch of rivets)
86 printf('\n pitch of rivets , pc = %.2f mm\n use pc =
    %.f mm',pc,pc)
87 eta_c=(pc-d0)/pc*100; // % (efficiency of joint)
88 printf('\n Efficiency of joint , eta_c = %.2f %%',
    eta_c)
89 dis=0.33*pc+0.67*d0; // mm (distance between rows of

```

```

        rivets)
90 printf('\n for zig-zag riveting , distance between
        rows of rivets = %.1f mm. use 65 mm', dis)
91 m=1.5*d0;// mm (Margin)
92 printf('\n margin , m = %.f mm',m)

```

---

### Scilab code Exa 5.2 Design an economical joint

```

1 // exa 5.2 Pg 147
2 clc;clear;close;
3
4 // Given Data
5 w=400;//mm
6 t=20;//mm
7 sigma_t=90;// MPa
8 tau=60;// MPa
9 sigma_c=140;// MPa
10
11 printf('\n Diameter of rivet , do = ')
12 d0=6*sqrt(t);//mm (for t>8 mm)
13 printf('%.2f mm',d0)
14 d0=29;//mm (standard)
15 printf('\n Standard diameter of rivet hole , do = %.f
        mm & corresponding diameter of rivet = 27 mm',d0
        )
16 Pt=(w-d0)*t*sigma_t;//N max. tearing strength of
        plate
17 Ps=1.75*pi/4*d0**2*tau;// N (shearing strength of
        one rivet)
18 n1=Pt/Ps;// no. of rivets
19 n=ceil(n1);
20 printf('\n no. of rivets , n = %.3f. Use n = %.f ',n1
        ,n)
21 t1=0.75*t;// mm
22 t2=t1;// mm

```

```

23 printf('\n thickness of inner butt strap , t1 = %.f
      mm', t1)
24 printf('\n thickness of outer butt strap , t2 = %.f
      mm', t2)
25 // section 1-1
26 P1=(w-d0)*t*sigma_t;//N
27 // section 2-2
28 P2=(w-2*d0)*t*sigma_t+1.75*pi/4*d0**2*tau;//N
29 // section 3-3
30 P3=(w-3*d0)*t*sigma_t+1.75*3*pi/4*d0**2*tau;//N
31 // section 4-4
32 P4=(w-4*d0)*t*sigma_t+1.75*6*pi/4*d0**2*tau;//N
33 Ps=10*Ps;// N (shearing stress of all rivets)
34 Pc=10*d0*t*sigma_c;// N (shearing stress of all
      rivets)
35 Pj=P1;// N (strength f joint)
36 P = w*t*sigma_t;// N (strength of solid plate)
37 eta=P1/P*100; // % (efficiency of joint)
38 printf('\n efficiency of joint = %.2f %%', eta)
39 p1=3*d0+5;// mm (pitch of rivets)
40 p=100;//mm (adopt for design)
41 printf('\n pitch of rivets = %.f mm. Use %.f mm',p1,
      p)
42 m1=1.5*d0;// mm (margin)
43 m=50;//mm
44 w=3*p+2*m;// mm
45 printf('\n margin ,\n m = %.1f mm. Use %.f mm', m1,m)
46 printf('\n w = %.f mm',w)
47 dis=2.5*d0;// mm
48 printf('\n distance between rows = %.1f mm. Use 75
      mm',dis)

```

---

### Scilab code Exa 5.3 Diameter of rivets

```
1 // exa 5.3 Pg 150
```

```

2  clc;clear;close;
3
4  // Given Data
5  n=6; // no. of rivets
6  P=54; // kN
7  e=200; //mm
8  a=50; //mm (from fig.5.13(a))
9  b=100; //mm (from fig.5.13(a))
10 tau=120; // MPa
11
12 Pd=P/n*1000; // N (direct shear load in rivet)
13 C=P*e; // kN.mm (Couple)
14 //l1=l3=l4=l6
15 l1=sqrt(a**2+b**2); // mm
16 l3=l1;l4=l1;l6=l1 //mm
17 l2=a;l5=a; //mm
18 // F1/l1*(4*l1**2+2*l2**2)=C
19 F1=C*1000/(4*l1**2+2*l2**2)*l1; // N
20 theta1=acos(a/l1); // radian
21 R1=sqrt(Pd**2+F1**2+2*Pd*F1*cos(theta1)); // N (
    resultant force in rivet 1)
22 //R1=%pi/4*d0**2*tau
23 d0=sqrt(R1/(%pi/4*tau)); // mm
24 printf('\n diameter of rivets = %.2f mm. Use d0 =
    17.5 mm & d=16 mm for design.',d0)

```

---

**Scilab code Exa 5.4** Design a double riveted butt joint

```

1  // exa 5.4 Pg 151
2  clc;clear;close;
3
4  // Given Data
5  D=0.75; //m
6  ps=1.55; // N/mm.sq
7  eta_1=0.75; // efficiency

```

```

8  sigma_t=90; // MPa
9  sigma_c=140; // MPa
10 tau=56; // MPa
11 n=2; // no. of rivets
12
13 printf('DESIGNING LONGITUDINAL JOINT - \n')
14 printf('\n Plate Thickness')
15 t = ps*D*1000/2/sigma_t/eta_1+1; // mm
16 printf(', t = %.2f mm',t)
17 t=ceil(t); //mm (adopted for design)
18 printf('\n use t = %d mm',t)
19
20 printf('\n Diameter of rivet hole , do = ')
21 d0=6*sqrt(t); //mm (for t>8 mm)
22 printf('%.2f mm',d0)
23 d0=19.5; // suggested for design
24 printf('\n Use do = %.1f mm',d0)
25 printf('\n Diameter of rivet , d = ')
26 d=d0-1.5; //mm
27 printf('%.2f mm',d)
28
29 printf('\n Pitch of rivets , p = ')
30 Ps=(2*1.875)*%pi/4*d0**2*tau; // N
31 // Putting Pt=Ps where Pt=(p-d0)*t*sigma_t; // N
32 Pt=Ps; //N
33 p=Pt/(t*sigma_t)+d0; // N
34 printf('%.2f mm',p)
35 C=3.5; // for 2 no. of rivets
36 pmax=C*t+40; // mm (as per IBR)
37 printf('\n as per IBR-\n pitch , pmax = %.f mm',pmax)
38 p=75; // mm (adopted for design)
39 printf('\n Use p = %.f mm',p)
40
41 //Distance between rows of rivets
42 dis=0.33*p+0.67*d0; // mm
43 printf('\n distance between rows of rivets = %.1f mm
',dis)
44 dis=40; //mm (adopted for design)

```



```

45 printf('\n take & use this distance = %.f mm', dis)
46
47 printf('\n Thickness of butt strap , t= ')
48 t1=0.625*t;// mm
49 printf(' %.2 f mm',t1)
50 t1=7;// mm (adopted for design)
51 printf('\n Use thickness = %.f mm',t1)
52
53 //margin
54 m=ceil(1.5*d0);// mm
55 printf('\n margin , m = %.f mm',m)
56
57 // Efficiency of joint
58 Pt=(p-d0)*t*sigma_t;// N
59 Ps=Ps;// N (shearing resistance of rivets)
60 Pc=n*d0*t*sigma_c;// N (crushing resistance of
    rivets)
61 sigma_com = (p-2*d0)*t*sigma_t+%pi/4*d0**2*tau;// N
62 printf('\n strength of the joint = %d N',Pt)
63 P=p*t*sigma_t;//N (strength of solid plate)
64 printf('\n strength of solid plate = %d N',P)
65 eta_l=Pt/P*100;// % (efficiency)
66 printf('\n Efficiency of joint , eta_l = %.2 f %% = 75
    %% as given',eta_l)

```

---

#### Scilab code Exa 5.6 Suitable diameter of rivets

```

1 // exa 5.6 Pg 153
2 clc;clear;close;
3
4 // Given Data
5 n=5;// no. of rivets
6 P=45;// kN
7 alfa=30;// degree
8 tau=120;// MPa

```

```

9
10
11 Pd=P/n*1000; // N (direct shear load in rivet)
12 // C.G. of rivet group
13 // values below are collected direct from figure
14 x_bar=(3*200)/5; // mm
15 y_bar=(1*50+1*150+1*100+1*200)/5; // mm
16 ex=300+x_bar+y_bar; //mm
17 ey=100; //mm
18 l1=sqrt(x_bar**2+(y_bar/2)**2); // mm
19 l2=l1; //mm
20 l3=sqrt(100**2+80**2); // mm
21 l4=80; //mm
22 l5=13; //mm
23
24 //2*F1*l1+2*F3*l3+F4*l4=P*cos(alfa)*ex+P*sin(alfa)*
    ey
25 F1=(P*1000*cosd(alfa)*ex+P*1000*sind(alfa)*ey)/(2*l1
    **2+2*l3**2+l4**2)*l1; //N
26 // rivet 1 is nearest
27 Beta = atand(x_bar/(y_bar/2)); // degree
28 theta1=Beta-(90-alfa); // degree
29 R1=sqrt(Pd**2+F1**2+2*Pd*F1*cosd(theta1)); // N (
    resultant force in rivet 1)
30 //R1=%pi/4*d0**2*tau
31 d0=sqrt(R1/(%pi/4*tau)); // mm
32 printf('\n diameter of rivets = %.2f mm. Use d0 =
    21.5 mm & d=20 mm for design.',d0)
33 // Note - Ans in the textbook is wrong.

```

---

### Scilab code Exa 5.7 Diameter of rivets

```

1 // exa 5.7 Pg 155
2 clc;clear;close;
3

```

```

4 // Given Data
5 t=6; //mm
6 sigma_t=220; // MPa
7 tau=100; // MPa
8 sigma_c=150; // MPa
9 n=2; // no. of rivets / pitch length
10 //Ps=n*pi/4*d0**2*tau; // shearing strength of
    rivets
11 //Pc=2*d0*t*sigma_c; // Crushing strength of rivets
12 d0=2*t*sigma_c/(n*pi/4*tau); // mm (equating Ps=Pc)
13 printf('Diameter of rivets , d0 = %.2f mm. Take d0
    =13.5 mm & d=12 mm',d0)
14 d0=13.5; //mm
15 d=12; //mm
16 //Pt=(p-d0)*t*sigma_t; // tearing strength
17 // equating Pt=Ps
18 //p= n*pi/4*d0**2*tau/(t*sigma_t)+d0; //mm
19 p= n*pi/4*d0**2*tau/(t*sigma_t)+d0
20 printf('\n Distance between rows of rivet = %.1f mm
    = %.f mm',p,p)
21 p=floor(p); //mm
22 pb=0.6*p; //mm (back pitch)
23 printf('\n back pitch = %.f mm',pb)
24 Pt=(p-d0)*t*sigma_t; // N (tearing strength)
25 printf('\n tearing strength = %.f N',Pt)
26 Ps=n*pi/4*d0**2*tau; // N ( shearing strength)
27 printf('\n shearing strength = %.f N',Ps)
28 Pc=2*d0*t*sigma_c; //N (Crushing strength of rivets)
29 printf('\n crushing strength = %.f N',Pc)
30 joint_strength = Pc; // N
31 printf('\n joint strength = %.f N',joint_strength)
32 P=p*t*sigma_t; //N (strength of solid plate)
33 printf('\n strength of solid plate = %.f N',P)
34 eta = joint_strength/P*100; // % (efficiency)
35 printf('\n efficiency of joint = %.1f %%', eta)

```

---

### Scilab code Exa 5.8 Diameter of rivets

```
1 // exa 5.8 Pg 156
2 clc;clear;close;
3
4 // Given Data
5 P=20; // kN
6 e=80; //mm
7 tau=150; // MPa
8
9
10 Pd=P/4; // kN
11 C=P*e; // kN.mm (Couple)
12 // As C.G. lies at 45mm from top rivet
13 l1=45;l4=45; //mm
14 l2=15;l3=15; //mm
15 // (F1/l1)*(2*l1*l4+2*l2*l3) = C
16 F1= C*1000/(2*l1*l4+2*l2*l3)*l1; //N
17 R1=sqrt(Pd**2+F1**2); // N
18 //R1=%pi/4*d0**2*tau
19 d0=sqrt(R1/(%pi/4*tau)); //mm
20 printf('Diameter of rivets - \n d0 = %.3f mm',d0)
21 printf('\n Use d0 = 13.5 mm & d = 12 mm')
```

---

# Chapter 6

## Shafts

Scilab code Exa 6.1 Shaft Diameter using AMSE code

```
1 // exa 6.1 Pg 168
2 clc;clear;close;
3
4 // Given Data
5 Sut=650; // MPa
6 Syt=380; // MPa
7 F1BYF2 = 2.5; // ratio of tensions
8 Fmax=2.5; // kN
9 da=200; // mm
10 db=400; // mm
11 L=1*1000; //mm
12 Km=1.5; // fatigue factor
13 Kt=1; // shock factor
14
15
16 tau_d1=0.30*Syt; // MPa
17 tau_d2=0.18*Sut; // MPa
18 tau_d=min(tau_d1, tau_d2); // MPa (taking minimum
    value)
19 tau_d=0.75*tau_d; //MPa (Accounting keyway effect)
20
```

```

21 // Pulley A
22 F1=2500; // N
23 F2=1000; // N
24 T=(F1-F2)*da/2; // N.mm
25 Fa=F1+F2; // N (resultant pull Downwards)
26
27 // Pulley B
28 // F3 & F4 are tension in belt (assumed)
29 //T=(F3-F4)*db/2
30 SUB_F3F4 = 2*T/db; // N (where SUB_F3F4 = F3-F4) --
    eqn(1)
31 F3BYF4=F1BYF2; // ratio of tensions --eqn(2)
32 F4 = SUB_F3F4/(F3BYF4-1); // N (using above 2
    equations)
33 F3=F3BYF4*F4; // N
34 Fb=F3+F4; // N (resultant pull right side( -->))
35
36 // BENDING MOMENTS -
37 Mav=Fa*L/4; // N.mm (vertical force)
38 Mc=Fb*da; // N.mm
39 Mah=Mc/2; // N.mm (vertical force)
40 M = sqrt(Mav**2+Mah**2); // N.mm (resultant bending
    moment at A)
41 d=((16/%pi/tau_d)*sqrt((Km*M)**2+(Kt*T)**2))**(1/3);
    // mm
42
43 printf('shaft diameter = %.2f mm. Use diameter = 45
    mm. ',d)

```

---

### Scilab code Exa 6.2 Diameter of the shaft

```

1 // exa 6.2 Pg 170
2 clc;clear;close;
3
4 // Given Data

```

```

5 Tmax=400; // N.m
6 Tmin=140; // N.m
7 Mmax=500; // N.m
8 Mmin=250; // N.m
9 Sut=540; // MPa
10 Syt=400; // MPa
11 n=2; // factor of safety
12 Kf=1.25; // given
13
14 Se_dash=0.4*Sut; // Mpa
15 Se=Se_dash/Kf; //MPa
16 Sys=0.577*Syt; // MPa
17 Ses=0.577*Se; // MPa
18 Mm=(Mmax+Mmin)/2; // N.m
19 Ma=(Mmax-Mmin)/2; // N.m
20 Tm=(Tmax+Tmin)/2; // N.m
21 Ta=(Tmax-Tmin)/2; // N.m
22 // Max. Distortion energy theory - Syt/n = 32/%pi/d
    **3*sqrt((Mm+Ma*(Syt/Se)**2)+0.75*(Tm+Ta*(Sys/Ses
    ))**2)
23 d = (32/%pi*sqrt((Mm+Ma*(Syt/Se)**2+0.75*(Tm+Ta*(
    Sys/Ses)**2)*1000/(Syt/n)**(1/3) ); // mm
24 printf('shaft diameter = %.2f mm. Use %.f mm. ',d,d)

```

---

### Scilab code Exa 6.3 Diameter of the shaft

```

1 // exa 6.3 Pg 171
2 clc;clear;close;
3
4 // Given Data
5 P=5; // kW
6 N=1000; // rpm
7 Syt=300; // N/mm.sq.
8 n=2; // factor of safety
9 v=0.25; // Poisson's ratio

```

```

10
11 //P=2*%pi*N*T/(60*1000)
12 T=P/(2*%pi*N/(60*1000)); // N.m
13 //tau = 16*T/%pi/d**3 // shear stress & sigma1 = tau
    ;sigma2=0;sigma3=-tau
14 // max. shear strain energy theory , sigma1**2+sigma3
    **2+(sigma3-sigma1)**2=2*(Syt/n)**2
15 d=(16*T*1000/%pi/sqrt(2/6*(Syt/n)**2))**(1/3); // mm
    (putting values of tau)
16 printf('shaft diameter = %.1f mm. Use %.f mm. ',d,
    ceil(d))

```

---

#### Scilab code Exa 6.4 Shaft Diameter using AMSE code

```

1 // exa 6.4 Pg 171
2 clc;clear;close;
3
4 // Given Data
5 Sut=700; // MPa
6 Syt=460; // Mpa
7 F1BYF2=3; // ratio of tensions
8 dg=300; // mm
9 dp=400; // mm
10 P=25; // kW
11 N=600; // rpm
12 alfa=20; // degree
13 Km=1.5; // fatigue factor
14 Kt=1.5; // shock factor
15
16 tau_d1=0.30*Syt; // MPa
17 tau_d2=0.18*Sut; // MPa
18 tau_d=min(tau_d1, tau_d2); // MPa (taking minimum
    value)
19 tau_d=0.75*tau_d; //MPa (Accounting keyway effect)
20

```



```

21 // Pulley D
22 // P= 2*%pi*N*T/60
23 T=P/(2*%pi*N/(60*1000)); // N.m
24 // (F1-F2)*dp/2=T
25 SUB_F1F2 = T*2/dp; // N (where SUB_F1F2 = F1-F2)
26 F2 = SUB_F1F2/(F1BYF2-1) ; // N (putting value of
    ratio)
27 F1=F1BYF2*F2; // N
28 F=F1+F2; // N
29 // Gear B
30 Ft=T*2/dg; // N
31 Fr=Ft*tand(alfa); // N
32
33 // Bearing Reactions
34
35 // Vertical forces
36 //RA*2*dg+Fr*dg=F*dg;
37 RA=(F*dg-Fr*dg)/(2*dg); // N (downwards)
38 RC=RA+Fr+F; // N (upwards)
39 MA=0; MB_v=-RA*dg; // N.mm
40 MC=-F*dg; // N.mm
41 //Horizontal forces
42 MB_h=Ft*2*dg/4; // N.mm
43 //Resultant B.M at B
44 MB=sqrt(MB_v**2+MB_h**2); // N.mm
45 Mmax=MC; //N.mm
46 T=T*1000; // N.mm
47 // d**3=16/%pi/tau_d*sqrt((Km*M)**2+(Kt*T)**2)
48 d=(16/%pi/tau_d*sqrt((Km*Mmax*1000)**2+(Kt*T)**2))
    *(1/3)
49 printf('shaft diameter(using ASME Code) = %.1 f mm.
    Use diameter = %. f mm. ',d,d)

```

---

Scilab code Exa 6.5 Diameter of the shaft

```

1 // exa 6.5 Pg 174
2 clc; clear; close;
3
4 // Given Data
5 L=1000; // mm
6 alfa=20; // degree
7 dg=500; // mm
8 L1=250; // mm
9 L2=300; // mm
10 dp=600; // mm
11 Wp=2000; // N
12 F1=2.5*1000; // N
13 F1BYF2=3; // ratio of tensions
14 tau_d=42; // MPa
15
16 F2=F1/F1BYF2; // N
17 T=(F1-F2)*dp/2; // N.mm
18 Ftg=2*T/dg; // N
19 Frg=Ftg*tand(alfa); // N
20 F=F1+F2; // N
21
22 // Vertical Loads
23 RAV=(Ftg*(L1+dg)+Wp*L2)/L; // N
24 RBV=Ftg+Wp-RAV; // N
25 MCV=RAV*L1; //N.mm
26 MDV=RBV*L2; // N.mm
27 // Horizontal Loads
28 RAH=(Frg*(L1+dg)+F*L2)/L; //N
29 RBH=Frg+F-RAH; // N
30 MCH=RAH*L1; // N.mm
31 MDH=RBH*L2; // N.mm
32 MD=sqrt(MDV**2+MDH**2); // N.mm
33 Mmax=MD; //N.mm
34 Te=MCV+MDV; // N.mm
35 //  $d^3 = 16*Te/\pi/\tau_d$ 
36 d = (16*Te/pi/tau_d)**(1/3); //mm
37 printf('shaft diameter = %.1f mm.',d)

```

---

### Scilab code Exa 6.6 Diameter of the shaft

```
1 // exa 6.6 Pg 176
2 clc;clear;close;
3
4 // Given Data
5 Tmax=400; // N.mm
6 Tmin=200; // N.mm
7 Mmax=500; // N.mm
8 Mmin=250; // N.mm
9 Sut=540; // MPa
10 Syt=420; // MPa
11 n=2; // factor of safety
12
13 Se=0.35*Sut; // MPa
14
15 Mm=(Mmax+Mmin)/2; // N.m
16 Ma=(Mmax-Mmin)/2; // N.m
17 Tm=(Tmax+Tmin)/2; // N.m
18 Ta=(Tmax-Tmin)/2; // N.m
19 Sys=0.5*Syt // MPa
20 Ses=0.5*Se; // MPa
21 // Max. Distortion energy theory -  $Syt/n = 32/\pi/d$ 
    **3*sqrt((Mm+Ma*(Syt/Se)**2)+0.75*(Tm+Ta*(Sys/Ses
    ))**2)
22 d = (32/%pi*sqrt((Mm+Ma*(Syt/Se)**2+0.75*(Tm+Ta*(
    Sys/Ses)**2)*1000/(Syt/n)**(1/3) ; // mm
23 printf('shaft diameter = %.f mm.',d)
```

---

### Scilab code Exa 6.7 Diameter of the bar

```
1 // exa 6.7 Pg 177
```

```

2  clc;clear;close;
3
4  // Given Data
5  Wmax=40; // kN
6  Wmin=20; // kN
7  L=500; // mm
8  Se_dash=350; // MPa
9  Sut=650; // MPa
10 Syt=500; // MPa
11 n=1.5; // factor of safety
12 ka=0.9; // surface finish factor
13 kb=0.85; // size factor
14 ke=1; // load factor
15 Kf=1; // fatigue strength factor
16
17 Wm=1/2*(Wmax+Wmin); // N
18 Wa=1/2*(Wmax-Wmin); // N
19 Se=ka*kb*ke*Se_dash; //MPa
20 Mm=Wm*L/1000/4; // kN.m
21 Ma=Wa*L/1000/4; // kN.m
22 //sigma_m=32*Mm/%pi/d**3; & sigma_a=32*Ma/%pi/d**3
23 //soderburg failure criteria - 1/n=sigma_m/Syt+Kf*
    sigma_a/Se
24 //d=((32/%pi*n/1000)*(Mm/Syt+Kf*Ma/Se))*(1/3)
25 d=((32/%pi/1000)*(Mm/Syt+Kf*Ma/Se)*n)**(1/3)*1000; //
    mm
26 printf('shaft diameter = %.f mm.',d)

```

---

### Scilab code Exa 6.8 Required shaft diameter

```

1  // exa 6.8 Pg 178
2  clc;clear;close;
3
4  // Given Data
5  Tmax=300; // N.mm

```

```

6 Tmin=-100; // N.mm
7 Mmax=400; // N.mm
8 Mmin=-200; // N.mm
9 n=1.5; // factor of safety
10 Sut=500; // MPa
11 Syt=420; // MPa
12 sigma_d=280; // MPa
13 ka=0.62; // surface finish factor
14 kb=0.85; // size factor
15 keb=1; // load factor for bending
16 kes=0.58; // load factor for torsion
17 Kfb=1; // fatigue strength factor for bending
18 Kfs=1; // fatigue strength factor for torsion
19
20 Se_dash=0.5*Sut; // MPa
21 Se=ka*kb*keb*Se_dash; // MPa
22 Ses_dash=0.5*Se_dash; // MPa
23 Ses=ka*kb*kes*Ses_dash; // MPa
24 Sys=0.5*Syt; // MPa
25 Mm=(Mmax+Mmin)/2; // N.m
26 Ma=(Mmax-Mmin)/2; // N.m
27 Tm=(Tmax+Tmin)/2; // N.m
28 Ta=(Tmax-Tmin)/2; // N.m
29
30 // tau_d/n = (16/%pi/d**3)*sqrt((Mm+Ma*(Syt/Se)**2)
    +(Tm+Ta*(Sys/Ses))**2)
31 tau_d=sigma_d/2; // MPa
32 d = ((16/%pi)*sqrt((Mm+Ma*(Syt/Se)**2)+(Tm+Ta*(Sys/
    Ses)**2))/(tau_d*10**6/n))**(1/3)*1000; // mm
33 printf('shaft diameter = %.2f mm.',d)
34 // Note - answer in the textbook is not accurate.

```

---

# Chapter 7

## Keys and couplings

Scilab code Exa 7.1 Design a protected type rigid cast iron flange

```
1 // exa 7.1 Pg 195
2 clc;clear;close;
3
4 // Given Data
5 P=20; // kW
6 N=240; // rpm
7 tau_s=45; // MPa
8 tau_b=30; // MPa
9 sigma_b=60; // MPa
10 sigma_cs=2*tau_s; // MPa
11 tau_ci=15; // MPa
12 //Tmax=1.25*Tm
13 mu=0.15; // coefficient of friction
14
15 //SHAFT DIAMETER
16 // P= 2*%pi*N*Tm/60/1000
17 Tm=P/(2*%pi*N/60/1000); // N.m
18 Tmax=1.25*Tm; // N.m
19 // %pi*d**3*tau_s/16= Tmax
20 d=(Tmax/(%pi*tau_s/16)*1000)**(1/3); // mm
21 printf('shaft diameter = %.2f mm. Use d = 50 mm.',d)
```

```

22 d=50; // mm
23
24 // HUB DIAMETER
25 // Tmax=%pi/16*((d1**4-d**4)/d1)*tau_h
26 tau_h=tau_ci; // MPa
27 //d1*(Tmax/(%pi/16)/tau_h)-d1**4=d**4 -- eqn(1)
28 Tmax=Tmax*1000; // N.mm
29 p=[1 0 0 -Tmax/(%pi*tau_h/16) -d**4] ;// polynomial
    coefficients from eqn(1)
30 d1=roots(p); // roots of poly
31 d1=d1(1); // mm (taking +ve value)
32 d1=100; // mm (empirically adopted)
33 t1=(d1-d)/2; // mm (thickness of hub)
34 printf('\n thickness of hub = %.f mm',t1)
35 d4=d+t1; // mm (diameter of recess in flanges)
36 printf('\n diameter of recess in flanges = %.f mm',
    d4)
37 d3=4*d; // mm (outside diameter of protecting flange)
38 printf('\n outside diameter of protecting flange = %
    .f mm',d3)
39
40 // Hub length
41 b=d/4; // mm (width of key)
42 l=1.5*d; // mm (length of key)
43 printf('\n width of key = %.1f mm. Use b = 15 mm',b)
44 b=15; // mm
45 printf('\n length of key = %.f mm.',l)
46 t=b; // mm (thickness for square key)
47 printf('\n thickness for square key = %.f mm',t)
48 printf('\n Hub length = %.f mm',l)
49
50 //Number of bolts
51 n=floor(4*d/150+3); // no. of bolts
52 printf('\n Number of bolts = %.f',n)
53
54 // Bolt diameter
55 r2=1.5*d; // mm
56 F=Tmax/r2/n; // N

```

```

57 // %pi/4*db**2*tau_b=F
58 db=sqrt(F/(%pi/4*tau_b)); // mm
59 printf('\n Bolt diameter = %.2f mm. Use db=12 mm', db
   )
60 bolt_dia=db; //mm
61
62 // Bolt diameter based on Tensile load
63 r3=d3/2; // mm
64 r4=d4/2; // mm
65 rf=2/3*((r3**3-r4**3)/(r3**2-r4**2)); // mm
66 //Tmax=n*mu*Pi*rf; // N
67 Pi=Tmax/(n*mu*rf); // N
68 // Pi=%pi/4*db**2*sigma_t
69 sigma_t=sigma_b; // MPa
70 db=sqrt(Pi/(%pi/4*sigma_t)); // mm
71 printf('\n Bolt diameter (based on Tensile load) = %
   .1f mm. Use db=15 mm', db)
72 db=15; // mm (adopted)
73
74 // Flange thickness
75 t2=0.5*t1+6; // mm (empirically)
76 printf('\n Flange thickness = %.1f mm. Use t=20 mm',
   t2)
77 t2=20; // mm (adopted)
78 //F=n*db*t2*sigma_c
79 sigma_ci=F/n/db/t2; // MPa
80 //2*%pi*d1**2*tau*t2/4=Tmax
81 tau=Tmax/(2*%pi*d1**2*t2/4); // MPa
82 printf('\n permissible bearing stress in flange = %
   .2f MPa < 30 MPa', sigma_ci)
83 printf('\n shearing of the flange at the junction
   with hub = %.2f MPa < 15 MPa.', tau)
84 printf(' Values are acceptable.')
85
86 // Check for crushing of bolt
87 //n*db*t2*sigma_cb*d2/2=Tmax
88 d2=d1+d; // mm
89 db=bolt_dia; //mm

```



```

90 sigma_cb=Tmax/(n*db*t2*d2/2); // MPa
91 printf('\n permissible crushing strength of bolts =
    %.1f MPa < 60 MPa. ',sigma_cb)
92 printf(' Hence design is safe.')
93
94 // Thickness of protecting flange
95 t3=0.5*t2; // mm
96 printf('\n Thickness of protecting flange = %.f mm',
    t3)
97 // Hub overlap
98 ho=3; // mm (min)
99 printf('\n Hub overlap = %.f mm (min) ',ho)
100 //Note – Answer for **Bolt diameter based on Tensile
    load** is calculated wrong in the textbook(error
    in Pi calculation).

```

---

Scilab code Exa 7.2 Design a bushed pin type flexible coupling

```

1 // exa 7.2 Pg 200
2 clc;clear;close;
3
4 // Given Data
5 P=30; // kW
6 N=750; // rpm
7 //Tmax=1.2*Tm; // MPa
8 tau_s=35; // MPa
9 tau_b=35; // MPa
10 tau_k=35; // MPa
11 sigma_cs=70; // MPa
12 sigma_ck=70; // MPa
13 sigma_cb=70; // MPa
14 tau_ci=15; // MPa
15 pb=0.8; // MPa
16
17 //sigma_cs=2*tau_s; // MPa

```

```

18
19 //Tmax=1.5*Tm
20 mu=0.15; // coefficient of friction
21
22 //SHAFT DIAMETER
23 // P= 2*%pi*N*Tm/60/1000
24 Tm=P/(2*%pi*N/60/1000); // N.m
25 Tmax=1.2*Tm; // N.m
26 // %pi*d**3*tau_s/16= Tmax
27 d=(Tmax/(%pi*tau_s/16)*1000)**(1/3); // mm
28 printf('shaft diameter = %.2f mm. Use d = 42 mm.',d)
29 d=42; // mm
30
31 // HUB DIAMETER
32 // Tmax=%pi/16*((d1**4-d**4)/d1)*tau_h
33 tau_h=tau_ci; // MPa
34 //d1*(Tmax/(%pi/16)/tau_h)-d1**4=d**4 --- eqn(1)
35 Tmax=Tmax*1000; // N.mm
36 p=[1 0 0 -Tmax/(%pi*tau_h/16) -d**4] ; // polynomial
    coefficients from eqn(1)
37 d1=roots(p); // roots of poly
38 d1=d1(1); // mm (taking +ve value)
39 d1=2*d; // mm (empirically adopted)
40 t1=(d1-d)/2; // mm (thickness of hub)
41 printf('\n thickness of hub = %.f mm',t1)
42 //d4=d+t1; // mm (diameter of recess in flanges)
43 //printf('\n diameter of recess in flanges = %.f mm
    ',d4)
44 d3=4*d; // mm (outside diameter of protecting flange)
45 printf('\n outside diameter of protecting flange = %
    .f mm. Use 170 mm',d3)
46 d3=170; // mm (adopted)
47
48 //Key size & Hub length
49 b=d/4; // mm (width of key)
50 l=1.5*d; // mm (length of key)
51 printf('\n width of key = %.1f mm. Use b = 12 mm',b)
52 b=12; // mm

```

```

53 printf('\n length of key = %.f mm. ',l)
54 t=b;// mm (thickness for square key)
55 printf('\n thickness for square key = %.f mm',t)
56 printf('\n Hub length = %.f mm',l)
57
58 //Number of bolts
59 n=(0.04*d+3);// no. of bolts
60 printf('\n Number of bolts = %.2f. Use n=6',n)
61 n=6;// adopted
62
63 // Bolt diameter
64 db=0.5*d/sqrt(n);// mm
65 printf('\n Bolt diameter = %.2f mm. Use db=20 mm for
        design purpose',db)
66 db=20;//mm (adopted)
67 bolt_dia=db;//mm
68 dsb=24;// mm(shank diameter of bolt for design)
69
70 // Outer diameter of rubber bush
71 trb=2;// mm (thickness of rubber bush)
72 tbb=6;// mm (thickness of brass bush)
73 d3=dsb+2*trb+2*tbb;// mm
74 d2=d1+d3+2*tbb;// mm (pitch circle diameter of bolts
    )
75 printf('\n pitch circle diameter of bolts = %.f mm ',
        ,d2)
76
77 // Check of shear in bolt
78 F=2*Tmax/n/d2;// N
79 //pi/4*db*2*tau=F
80 tau=F/(%pi/4*db**2);//MPa
81 printf('\n Permissible shear stress in bolts = %.2f
        MPa < 35 MPa. Hence design is safe.', tau)
82
83 // Length of brush
84 pb=0.8;// MPa(bearing pressure of brush)
85 //F=12*d3*pb;
86 l2=F/d3/pb;// mm

```

```

87 printf('\n length of bush = %.f mm',l2)
88
89 // Check for pin in bending
90 c=5;// mm (clearance between two flanges)
91 l3=(l2-c)/2+c;//mm
92 //M=%pi/32*db**3*sigma_b & M=F*l3
93 sigma_b = F*l3/(%pi/32*db**3);// MPa
94 printf('\n Bending stress in pin = %.1f MPa',sigma_b
    )
95
96 // Maximum shear stress in pin
97 tau_max=sqrt((sigma_b/2)**2+tau**2);//MPa
98 printf('\n Maximum shear stress in pin = %.2f MPa <
    35 MPa. Hence design is safe.',tau_max)
99
100 // Flange thickness
101 t2=0.5*t1+6;// mm (empirically)
102 printf('\n Flange thickness = %.1f mm. Use t=18 mm',
    t2)
103 t2=18;// mm (adopted)
104 tau=Tmax/(2*%pi*d1**2*t2/4);// MPa
105 printf('\n shearing of the flange at the junction
    with hub = %.2f MPa < 15 MPa.',tau)
106 printf(' Values are acceptable.')
107
108 //Note – Answer in llast part is calculated wrong in
    the textbook(error in calculation).

```

---

### Scilab code Exa 7.3 Length of hub

```

1 // exa 7.3 Pg 204
2 clc;clear;close;
3
4 // Given Data
5 n=8;// no. of spline

```

```

6 d=52; // mm
7 D=60; // mm
8 pm=6; // MPa
9 mu=0.06; // coefficient of friction
10 N=320; // rpm
11 P=20; // kW
12
13 T=60*10**3*P/2/%pi/N; // N.m
14 l=8*T*10**3/pm/n/(D**2-d**2); // mm
15 printf('length of hub = %.1f mm',l)
16 Rm=(D+d)/4; // mm
17 F=T*10**3/Rm; // N
18 Ff=mu*F; //N (Force of friction)
19 printf('\n Force required to shift the connection =
        %.1f N',Ff)

```

---

#### Scilab code Exa 7.4 Design the rectangular key

```

1 // exa 7.4 Pg 204
2 clc;clear;close;
3
4 // Given Data
5 d=75; // mm
6 tau=50; // MPa
7 sigma_c=75; // MPa
8 printf('for key to be equally strong in shear &
        crushing - \n')
9 b=d/4; // mm
10 printf(' b= %.2f mm. Use b=20 mm. ',b)
11 b=20; //mm
12 //2*b/t=sigma_c/tau for key to be equally strong in
        shear & crushing
13 t=2*b/(sigma_c/tau); // mm
14 printf('\n t=%.2f mm. Use t=27 mm',t)
15 l= %pi*d**2/8/b; // mm (for key to be equally strong

```

```

    in shear as shaft)
16 printf('for key to be equally strong in shear as
    shaft - \n')
17 printf(' l=%0.1f mm. Use l=115 mm',l)

```

---

### Scilab code Exa 7.6 Suitable dimensions for shaft

```

1 // exa 7.6 Pg 205
2 clc;clear;close;
3
4 // Given Data
5 P=135; // kW
6 N=120; // rpm
7 tau_s=55; // MPa
8 tau_b=45; // MPa
9 tau_ci=175; // MPa
10 sigma_ci=75; // MPa
11
12 //sigma_cs=2*tau_s; // MPa
13
14 //Tmax=1.5*Tm
15 mu=0.15; // coefficient of friction
16
17 //SHAFT DIAMETER
18 // P= 2*pi*N*Tm/60/1000
19 Tm=P/(2*pi*N/60/1000); // N.m
20 // pi*d**3*tau_s/16= Tm
21 d=(Tm/(pi*tau_s/16)*1000)**(1/3); // mm
22 d=ceil(d)
23 printf('shaft diameter = %0.2f mm.',d)
24 Tmax=Tm; // N.m
25
26 // HUB DIAMETER
27 // Tmax=pi/16*((d1**4-d**4)/d1)*tau_h
28 tau_h=tau_ci; // MPa

```

```

29 //d1*(Tmax/(%pi/16)/tau_h)-d1**4=d**4 -- eqn(1)
30 Tmax=Tmax*1000;// N.mm
31 p=[1 0 0 -Tmax/(%pi*tau_h/16) -d**4] ;// polynomial
    coefficients from eqn(1)
32 d1=roots(p);// roots of poly
33 d1=d1(1);// mm (taking +ve value)
34 d1=2*d;// mm (empirically adopted)
35 t1=(d1-d)/2;// mm (thickness of hub)
36 printf('\n thickness of hub = %.f mm',t1)
37 d4=d+t1;// mm (diameter of recess in flanges)
38 printf('\n diameter of recess in flanges = %.f mm',
    d4)
39 d3=4*d;// mm (outside diameter of protecting flange)
40 printf('\n outside diameter of protecting flange = %
    .f mm. ',d3)
41
42 //Key size & Hub length
43 b=d/4;// mm (width of key)
44 l=1.5*d;// mm (length of key)
45 printf('\n width of key = %.1f mm. ',b)
46 printf('\n length of key = %.f mm. ',l)
47 t=b;// mm (thickness for square key)
48 printf('\n thickness for square key = %.f mm',t)
49 printf('\n Hub length = %.f mm',l)
50
51 //Number of bolts
52 n=ceil(4*d/150+3);// no. of bolts
53 printf('\n Number of bolts = %.2f. ',n)
54
55 // Bolt diameter
56 r2=1.5*d;// mm
57 F=Tm*1000/r2/n;//N
58 //(%pi/4)*db**2*tau_b=F
59 db=sqrt(F/((%pi/4)*tau_b));// mm
60 printf('\n Bolt diameter = %.2f mm. Use db=20 mm for
    design purpose ',db)
61 db=20;// mm (adopted for design)
62 bolt_dia=db;//mm

```

```

63
64 // Flange thickness
65 t2=0.5*t1+6;// mm (empirically)
66 printf('\n Flange thickness = %.1f mm. Use t=20 mm',
        t2)
67 //F=n*db*t2*sigma_c
68 sigma_ci=F/n/db/t2;// MPa
69 //2*%pi*d1**2*tau*t2/4=Tmax
70 tau=Tmax/(2*%pi*d1**2*t2/4);// MPa
71 printf('\n permissible bearing stress in flange = %
        .2f MPa < 75 MPa',sigma_ci)
72 printf('\n shearing of the flange at the junction
        with hub = %.2f MPa < 175 MPa.',tau)
73 printf(' Values are acceptable.')
74
75 // Check for crushing of bolt
76 //n*db*t2*sigma_cb*d2/2=Tmax
77 d2=d1+d;// mm
78 db=bolt_dia;//mm
79 sigma_cb=Tmax/(n*db*t2*d2/2);// MPa
80 printf('\n permissible crushing strength of bolts =
        %.2f MPa < 60 MPa.',sigma_cb)
81 printf(' Hence design is safe.')
82 // Thickness of protecting flange
83 t3=0.5*t2;// mm
84 printf('\n Thickness of protecting flange = %.f mm',
        t3)
85 // Hub overlap
86 ho=3;// mm (min)
87 printf('\n Hub overlap = %.f mm (min)',ho)

```

---

Scilab code Exa 7.7 Design the rectangular key

```

1 // exa 7.7 Pg 208
2 clc;clear;close;

```



```

3
4 // Given Data
5 d=50; // mm
6 tau=42; // MPa
7 sigma_c=72; // MPa
8 printf('for key to be equally strong in shear &
        crushing - \n')
9 b=d/4; // mm
10 printf(' b= %.2f mm. Use b=15 mm. ',b)
11 b=15; //mm
12 //2*b/t=sigma_c/tau for key to be equally strong in
        shear & crushing
13 t=2*b/(sigma_c/tau); // mm
14 printf('\n t=%.2f mm. Use t=20 mm',t)
15 l= %pi*d**2/8/b; // mm (for key to be equally strong
        in shear as shaft)
16 printf('\n for key to be equally strong in shear as
        shaft - \n')
17 printf(' l=%.2f mm. Use l=70 mm',l)

```

---

### Scilab code Exa 7.8 Design the rectangular key

```

1 // exa 7.8 Pg 208
2 clc;clear;close;
3
4 // Given Data
5 d=25; // mm
6 N=550; // rpm
7 P=12; // kW
8 sigma_yt=400; // N/mm.sq.
9 sigma_yc=400; // N/mm.sq.
10 n=2.5; // factor of safety
11
12 // P= 2*%pi*N*T/(60*10**3)
13 T=P/(2*%pi*N/(60*10**3)); // N.m

```

```

14 tau=0.5*sigma_yt;// MPa
15 tau_d=tau/n;// N/mm.sq.
16 printf('design shear stress = %.f N/mm.sq. ',tau_d)
17 sigma_d=sigma_yc/n;// N/mm.sq.
18 printf('\n design crushing strength = %.f N/mm.sq. ',
    sigma_d)
19 b=d/4;//mm
20 printf('\n width of key = %.f mm. Use 7mm',b)
21 b=ceil(d/4);// mm
22 t=b;// mm
23 printf('\n thickness of key = %.f mm. ',t)
24 l_s=2*T*10**3/(d*b*tau_d);// mm (length of key based
    on shear failure)
25 printf('\n length of key based on shear failure = %
    .2f mm or %.f mm',l_s, l_s)
26 l_c=4*T*10**3/(d*t*sigma_d);// mm (length of key
    based on crushing failure)
27 printf('\n length of key based on crushing failure =
    %.2f mm or %.f mm',l_c, l_c)

```

---

### Scilab code Exa 7.9 Design a protected type rigid cast iron flange

```

1 // exa 7.9 Pg 209
2 clc;clear;close;
3
4 // Given Data
5 d=36;// mm
6 P=15;// kW
7 N=720;// rpm
8 //Tmax=1.25*Tm
9 sigma_yt=245;// MPa (for C20 steel)
10 n=3;// factor of safety
11 sigma=82;// MPa (Design tensile stress)
12
13 tau=0.577*sigma;// MPa (shear stress)

```

```

14 sigma_u=200; // MPa (for FG 200 cast Iron)
15 n2=5; // factor of safety (for FG 200 cast Iron)
16 tau2=20; // MPa shear stress (for FG 200 cast Iron)
17
18 // Max. torque transmitted
19 //P=2*%pi*N*Tm/(60*10**3)
20 Tm=P/(2*%pi*N/(60*10**3))*1000; // N.mm
21 Tmax=1.25*Tm; // N.mm
22 printf('\n Maximum transmitted torque = %.f N.mm',
        Tmax)
23
24 // Hub diameter
25 tau_h=20; // MPa (permissible shear stress in hub)
26 //Tmax=(%pi/16)*(d1**4-d**4)/d1*tau_h ... eqn(1)
27 d1=2*d; //mm (empirically)
28 tau_h=Tmax*1000/((%pi/16)*(d1**4-d**4)/d1); // MPa
29 t1=(d1-d)/2; // mm (thickness of hub)
30 printf('\n Hub diameter = %.f mm',d1)
31 printf('\n Thickness of hub = %.f mm',t1)
32 d4=d+t1; // mm
33 printf('\n Diameter of recess in flanges = %.f mm',
        d4)
34 d3=4*d; //mm
35 printf('\n Outside diameter of protecting flange = %
        .f mm',d3)
36
37 //Hub length
38 b=d/4; // mm (width of key)
39 l=1.5*d; // mm (length of key)
40 printf('\n width of key = %.1f mm. ',b)
41 printf('\n length of key = %.f mm. ',l)
42 t=b; // mm (thickness for square key)
43 printf('\n thickness for square key = %.f mm',t)
44 printf('\n Hub length = %.f mm',l)
45
46 //Number of bolts
47 n=ceil(4*d/150+3); // no. of bolts
48 printf('\n Number of bolts = %.2f. ',n)

```

```

49
50 // Bolt diameter
51 r2=1.5*d; // mm
52 F=Tmax/r2/n; //N
53 //((%pi/4)*db**2*tau_b=F
54 db=sqrt(F/((%pi/4)*tau)); // mm
55 printf('\n Bolt diameter = %.2f mm. Use db=6 mm for
    design purpose ',db)
56 db=6; // mm (adopted for design)
57 bolt_dia=db; //mm
58
59 // Flange thickness
60 t2=0.5*t1+6; // mm (empirically)
61 printf('\n Flange thickness = %.1f mm. Use t=20 mm',
    t2)
62 //F=n*db*t2*sigma_c
63 sigma_ci=F/n/db/t2; // MPa
64 //2*%pi*d1**2*tau*t2/4=Tmax
65 tau=Tmax/(2*%pi*d1**2*t2/4); // MPa
66 printf('\n permissible bearing stress in flange = %
    .2f MPa < 40 MPa',sigma_ci)
67 printf('\n shearing of the flange at the junction
    with hub = %.2f MPa < 20 MPa.',tau)
68 printf(' Values are acceptable.')
69
70 // Check for crushing of bolt
71 //n*db*t2*sigma_cb*d2/2=Tmax
72 d2=d1+d; // mm
73 db=bolt_dia; //mm
74 sigma_cb=Tmax/(n*db*t2*d2/2); // MPa
75 printf('\n permissible crushing strength of bolts =
    %.2f MPa < 82 MPa.',sigma_cb)
76 printf(' Hence design is safe.')
77 // Thickness of protecting flange
78 t3=0.5*t2; // mm
79 printf('\n Thickness of protecting flange = %.f mm',
    t3)
80 // Hub overlap

```

```

81 ho=3; // mm (min)
82 printf('\n Hub overlap = %.f mm (min)',ho)

```

---

### Scilab code Exa 7.10 Diameter of bolts

```

1 // exa 7.10 Pg 212
2 clc;clear;close;
3
4 // Given Data
5 d=35; // mm
6 d2=125; // mm
7 n=6; // factor of safety
8 T=800; // N.m
9 N=350; // rpm
10 tau_s=63; // MPa
11 tau_b=56; // MPa
12 tau_CI=10; // MPa
13 tau_k=46; // MPa
14
15 // Diameter of bolts:
16 F=2*T*10**3/d2/n; // N
17 // %pi/4*db**2*tau_b=F
18 db=sqrt(F/(%pi/4*tau_b)); // mm
19 printf('\n (i) Diameter of bolts = %.2f mm. Use 8 mm
    .',db)
20
21 // Flange thickness
22 d1=2*d; // mm
23 //T=%pi/2*d1**2*t2*tau_CI
24 t2=T*1000/(%pi/2*d1**2*tau_CI); // mm
25 printf('\n (ii) Flange thickness = %.1f mm. Use t2 =
    12 mm',t2)
26 t2=12; // mm
27
28 //Key dimensions

```

```

29 b=10; // mm (width of key)
30 t=7; // mm (from tables)
31 //T=l*b*tau_k*d/2
32 l=T*10**3/(b*tau_k*d/2); // mm
33 l=ceil(l); // mm
34 printf('\n (iii) Length of key = %.f mm\n\t\t d=%.f
      mm\n\t\t t=%.f mm',l,d,b)
35
36 // Hub length
37 lh=1; // mm (length of hub)
38 printf('\n (iv) Hub length = %.f mm',lh)
39 tau_c=T*10**3/(%pi/16*(d1**4-d**4)/d1); // N/mm.sq.
40 printf('\n shear stress in hub = %.2f N/mm.sq.',
      tau_c)
41 printf('It is nearly equal to %.f N/mm.sq.',tau_CI)
42 printf('\n hence design parameters are fine.')
43
44 // Power transmitted
45 P=2*%pi*N*T/60/10**3; // kW
46 printf('\n (v) Power transmitted = %.2f kW',P)

```

---

# Chapter 8

## Mechanical Springs

Scilab code Exa 8.1 Design a close coiled helical compression spring

```
1 // exa 8.1 Pg 227
2 clc;clear;close;
3
4 // Given Data
5 Fmin=250;// N
6 Fmax=300;// N
7 del=8;// mm
8 C=8;// spring index
9 tau_d=420;// MPa
10 G=84;// GPa
11
12 // 1. Wahl's correction factor
13 Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction
    factor
14 printf("\n Wahl's correction factor = %.3f ",Kw)
15 // 2. Wire diameter
16 // tau_d=Kw*8*Fmax*C/%pi/d**2
17 d=sqrt(Kw*8*Fmax*C/%pi/tau_d);// mm
18 printf('\n Wire diameter = %.2f mm. Use 4.25 mm.',d)
19 d=4.25;// mm
20 // 3. Mean coil diameter
```

```

21 Dm=8*d; // mm
22 printf('\n Mean coil diameter = %.f mm.',Dm)
23 // 4. Stiffness of spring
24 k=(Fmax-Fmin)/del; // N/mm
25 // 5. no. of active turns
26 n = G*10**3*d/8/C**3/k ; // no. of active turns
27 printf('\n no. of active turns = %.f',n)
28 // 6. total no. of turns for squared and ground ends
29 nt=n+2; // total no. of turns for squared and ground
    ends
30 printf('\n total no. of turns for squared and ground
    ends = %.f',nt)
31 // 7. Free length of spring
32 // lf=l_s+del_max+clashallowance(=0.15*del_max)
33 del_max=del*Fmax/(Fmax-Fmin); //mm
34 l_s=nt*d; // mm
35 lf=l_s+del_max+0.15*del_max; // mm
36 printf('\n Free length of spring = %.1f mm Use 124
    mm',lf)
37 lf=124; //mm
38 // 8. Pitch of coils
39 p=lf/(nt-1); //mm
40 printf('\n Pitch of coils = %.2f mm',p)
41 // 9. Check for buckling
42 printf('\n Check for buckling - ')
43 m=lf/Dm; // > 2.6 provided guide
44 printf('\n ratio lf/Dm = %.3f > 2.6. So, Providing
    guide is necessary.',m)
45 kl_1=0.22; // for hinged ends
46 kl_2=0.62; // for fixed ends
47 Fcr_1=k*kl_1*lf; //N (for hinged ends)
48 Fcr_2=k*kl_2*lf; //N (for fixed ends)
49 printf('\n Critical load for buckling - ')
50 printf('\n Fcr = %.1f N for hinged ends < Fmax',
    Fcr_1)
51 printf('\n Fcr = %.1f N for fixed ends > Fmax',Fcr_2
    )
52 printf('\n From above two calculatio , it can be seen

```



```

        that spring is safe in buckling for fixed ends.'
    )
53 // 10. Lowest natural frequency for both ends fixed
54 rau=7800; // N/mm.cube. (Density of spring material)
55 fn=d/(%pi*n*Dm**2)*sqrt(G*10**3/8/(rau*10**-9)); //
56 printf('\n\n Lowest natural frequency for both ends
        fixed , fn = %.3 f Hz' ,fn)

```

---

### Scilab code Exa 8.2 Factor of safety

```

1 // exa 8.2 Pg 228
2 clc;clear;close;
3
4 // Given Data
5 Fmin=60; // N
6 Fmax=140; // N
7 d=3; // mm
8 Dm=18; // mm
9 Sut=1430; // MPa
10
11 C=Dm/d; // spring index
12 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
13 Ks=1+0.5/C; // Shear Stress factor
14 Fm=(Fmax+Fmin)/2; // N
15 Fa=(Fmax-Fmin)/2; // N
16 tau_m=Ks*(8*Fm*C)/(%pi*d**2); // MPa
17 tau_a=Kw*(8*Fa*C)/(%pi*d**2); // MPa
18 Ses_dash=0.22*Sut; // MPa
19 Sys=0.45*Sut; // MPa
20 //tau_m/Sys+tau_a/Ses_dash*(2-Ses_dash/Sys)=1/n
21 n=1/(tau_m/Sys+tau_a/Ses_dash*(2-Ses_dash/Sys)); //
    factor of safety
22 printf('\n factor of safety = %.2 f' ,n)

```

---

### Scilab code Exa 8.3 Initial torsional shear stress

```
1 // exa 8.3 Pg 229
2 clc;clear;close;
3
4 // Given Data
5 Fi=40; // N
6 d=3; // mm
7 C=6; // spring index
8 n=15; // factor of safety
9 tau=650; // N/mm.sq.
10 G=84; // kN/mm.sq.
11
12 // Wahl's correction factor
13 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
14 printf("\n Wahl''s correction factor = %.4f ",Kw)
15
16 // Initial torsional shear stress
17 tau_i=Kw*(8*Fi*C)/(pi*d**2); // MPa
18 printf('\n Initial torsional shear stress = %.2f MPa
    ',tau_i)
19 k=G*10**3*d/(8*C**3*n); // spring stiffness
20 printf('\n spring stiffness = %.2f N/mm',k)
21 // Spring load to cause yielding
22 //tau=Kw*(8*Fi*C)/(pi*d**2)
23 F=tau/(Kw*(8*C)/(pi*d**2))
24 printf('\n Spring load to cause yielding = %.1f N',F
    )
```

---

### Scilab code Exa 8.4 Diameter of spring wire

```

1 // exa 8.4 Pg 230
2 clc;clear;close;
3
4 // Given Data
5 Fmin=500; // N
6 Fmax=1200; // N
7 C=6; // spring index
8 n=1.5; // factor of safety
9 Sys=760; // MPa
10 Ses_dash=350; // MPa
11 del=25; // mm
12 G=82; // kN/mm.sq.
13
14 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
15 Ks=1+0.5/C; // Shear stress factor
16 Fm=(Fmax+Fmin)/2; // N
17 Fa=(Fmax-Fmin)/2; // N
18 tau_m_into_d_sq=Ks*(8*Fm*C)/(%pi); // where
    tau_m_into_d_sq = tau_m*d**2
19 tau_a_into_d_sq=Kw*(8*Fa*C)/(%pi); // where
    tau_a_into_d_sq = tau_a*d**2
20
21 // (tau_m-tau_a)/Sys+2*tau_a/Ses_dash=1/n
22 d=sqrt(n)*sqrt((tau_m_into_d_sq-tau_a_into_d_sq)/Sys
    +2*tau_a_into_d_sq/Ses_dash); // mm
23 printf('\n diameter of spring wire = %.2f mm or %.f
    mm',d, ceil(d))
24 d=ceil(d); // mm
25 Dm=C*d; // mm
26 printf('\n Mean coil diameter = %.f mm', Dm)
27 // del=8*Fmax*Ci**3/(G*d)
28 i=(del/(8*Fmax*C**3/(G*10**3*d))); // no. of active
    coils
29 i=ceil(i); // no. of active coils
30 printf('\n no. of active coils = %.f',i)
31 nt=i+2; // no. of active coils (for square & ground
    ends)

```

```

32 lf=nt*d+1.15*del; // mm
33 printf('\n free length of spring = %.2f mm',lf)

```

---

### Scilab code Exa 8.5 Design a spring for boiler safety

```

1 // exa 8.5 Pg 231
2 clc;clear;close;
3
4 // Given Data
5 p=125; // MPa
6 dv=60; // mm
7 del1=40; // mm
8 del2=20; // mm
9 tau_max=600; // MPa
10 G=85; // kN/mm.sq.
11 C=6; // spring index
12
13 Fv=(%pi/4)*dv**2*p/100; // N (Force on the valve)
14 del_max=del1+del2; // mm (Max. deflection)
15 Fmax=Fv*dv/del1; // N (Max. force)
16 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
17 // tau = 8*Fmax*C*Kw/(%pi*d**2)
18 d=sqrt((8*Fmax*C*Kw/(%pi))/tau_max); // mm (Diameter
    of spring wire)
19 Dm=6*d; // mm (Mean coil diameter)
20 n=G*10**3*d*del_max/(8*Fmax*C**3); // no. of turns
21 n = ceil(n); // no. of turns
22 nt=n+2; // total no. of turns
23 lf=nt*d+1.15*del_max; // mm (Free length)
24 p=lf/(nt-1); // mm (Pitch of coil)
25 printf('\n Force on the valve = %.1f N',Fv)
26 printf('\n Maximum deflection = %.f mm', del_max)
27 printf('\n Maximum force = %.1f N', Fmax)
28 printf('\n Wahl's correction factor = %.4f ',Kw)

```

```

29 printf('\n Diameter of spring wire = %.f mm',d)
30 printf('\n Mean coil diameter = %.f mm', Dm)
31 printf('\n number of turns = %.f ',n)
32 printf('\n Total number of turns for square & ground
    ends = %.f ',nt)
33 printf('\n Free length = %.f mm. Use 200 mm',lf)
34 printf('\n Pitch of coil = %.1f mm',p)

```

---

### Scilab code Exa 8.7 Design the spring

```

1 // exa 8.7 Pg 232
2 clc;clear;close;
3
4 // Given Data
5 dv=30; // mm
6 Wv=10; // N
7 Wl=25; // N
8 lf=100; // mm
9 del1=20; // mm
10 p=3.5; // N/mm.sq.
11 valve_lift=2; // mm
12 C=6; // spring index
13 tau=500; // N/mm.sq.
14 G=0.84*10**5; // N/mm.sq.
15
16 W=(%pi/4)*dv**2*p; // N (load on the valve at
    operating condition)
17 W1=W-Wv; //N (Net load on the valve at operating
    condition)
18 //W1*100=Wl*150+S1*200+P*300 // taking momens about
    the fulcrum
19 //S1*200+P*300=Wl*100-W1*150 ... eqn(1)
20 valve_lift=20*100/200; // mm //from figure (when
    spring is extended by 20 mm)
21 spring_extension=2*200/100; // mm // from figure (

```

```

    when valve is lifted 2 mm)
22 valve_load=W*12/10; // N // (when valve is lifted 2
    mm)
23 W2=valve_load-Wv; // N // (when valve is lifted 2 mm)
24 del2=del1+4; // mm (when valve is lifted)
25 //S2=S1*del2/del1; // spring force when valve is
    lifted
26 //S1*del2/del1-s2=0 ... eqn(1)
27 //W2*100=W1*150+S2*200+P*300 // taking momens about
    the fulcrum
28 //S2*200+P*300 =W2*100-W1*150 ... eqn(2)
29 //S1*200+P*300=W1*100-W1*150 ... eqn(3)
30 // solving above 3 eqn. by matrix method
31 A=[del2/del1 -1 0;200 0 300;0 200 300];
32 B=[0;W1*100-W1*150;W2*100-W1*150];
33 X=A**-1*B; // solution matrix
34 S1=X(1); // N
35 S2=X(2); // N
36 printf('\n Spring force when valve is lifted = %.1f
    N',S2)
37 printf('\n\n Design of spring - ')
38 k=(S2-S1)/(del2-del1); // N/mm (Spring stiffness)
39 printf('\n Spring stiffness = %.2f N/mm',k)
40 Kw=(4*C-1)/(4*C+0.615/C); // Wahl's correction
    factor
41 printf('\n Wahl''s correction factor = %.4f',Kw)
42 // tau=Kw*8*S2*C/%pi/d**2 max. shear stress
43 d=sqrt(Kw*8*S2*C/%pi/tau); // mm (spring diameter)
44 printf('\n spring diameter = %.2f mm or %.f mm',d,d)
45 d=ceil(d); // mm
46 // k=G*d/(8*C**3*n) (Spring stiffness)
47 n=G*d/(8*C**3*k); // no. of active coils
48 printf('\n no. of active coils = %.2f. Use n=7',n)
49 n=ceil(n); // rounding
50 nt=n+1; // total no. of active coils
51 printf('\n total no. of active coils = %.f',nt)
52 p=1f/(n-1); // mm (pitch of coils)
53 printf('\n pitch of coils = %.2f mm',p)

```

---

Scilab code Exa 8.8 Design a spring for a balance

```
1 // exa 8.8 Pg 234
2 clc;clear;close;
3
4 // Given Data
5 Fmin=0; // N
6 Fmax=1000; // N
7 del=80; // mm
8 Do=25; // mm
9 n=30; // no. of turns
10 G=85; // kN/mm.sq.
11
12 k=(Fmax-Fmin)/del; // N/mm (Spring stiffness)
13 printf('\n Spring stiffness = %.1f N/mm',k)
14 // k=G*d/(8*C**3*n) (Spring stiffness)
15 C_cube_BY_d=G*10**3/(k*8*n); //
16
17 function [C,d]=hitntrial(c3d,Do)
18     for C=5:-0.1:4.5
19         d=C**3/(c3d);
20         Doo=d*C+C;
21         if Doo<Do
22             break;
23         end;
24     end
25
26 endfunction;
27
28 [C,d]=hitntrial(C_cube_BY_d,Do)
29 printf('\n By hit and trial method and using value
30 of C^3/d -')
31 printf('\n value of Spring Index, C = %.1f ',C)
32 printf('\n value of wire diameter, d = %.1f mm',d)
```

```

32 printf('\n But we adopt d=4 mm. ')
33 d=4; // mm (adopted for design)
34 C=(C_cube_BY_d*d)**(1/3); // Spring index
35 printf('Hence, Spring Index = %.2f ',C)
36 Dm=C*d; // mm
37 printf('\n Mean coil diameter = %.2f mm', Dm)
38 Do=Dm+d; // mm
39 printf('\n Outside coil diameter = %.2f mm < 25 mm.
    Hence design is ok.', Do)
40 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
41 printf('\n Wahl's correction factor = %.3f ',Kw)
42 tau=8*Kw*C*Fmax/(%pi*d**2); // N/mm.sq.
43 printf('\n Maximum shear stress = %.2f N/mm.sq.',tau
    )

```

---

#### Scilab code Exa 8.10 Wire diameter of spring

```

1 // exa 8.10 Pg 235
2 clc;clear;close;
3
4 // Given Data
5 Fmin=600; // N
6 Fmax=1000; // N
7 C=6; // spring index
8 n=1.5; // factor of safety
9 Sys=700; // N/mm.sq.
10 Ses_dash=350; // N/mm.sq.
11
12 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
13 Ks=1+0.5/C; // Shear Stress factor
14 Fm=(Fmax+Fmin)/2; // N
15 Fa=(Fmax-Fmin)/2; // N
16 tau_m_into_d_sq=Ks*(8*Fm*C)/(%pi); // where

```



```

    tau_m_into_d_sq = tau_m*d**2
17 tau_a_into_d_sq=Kw*(8*Fa*C)/(%pi); // where
    tau_a_into_d_sq = tau_a*d**2
18
19 //(tau_m-tau_a)/Sys+2*tua_a/Ses_dash=1/n
20 d=sqrt(n)*sqrt((tau_m_into_d_sq-tau_a_into_d_sq)/Sys
    +2*tau_a_into_d_sq/Ses_dash); // mm
21 printf('wire diameter of spring = %.2f mm',d)

```

---

### Scilab code Exa 8.11 Design a helical spring

```

1 // exa 8.11 Pg 236
2 clc;clear;close;
3
4 // Given Data
5 dv=100; //mm
6 C=5.5; // spring index
7 pi=1; // N/mm.sq.
8 p=1.075; // N/mm.sq.
9 del=6; // mm
10 tau_max=400; // N/mm.sq.
11 G=80; // kN/mm.sq.
12
13 Fi=(%pi/4)*dv**2*pi; // N (initial tension in spring)
14 printf('\n initial tension in spring = %.f N', Fi)
15 F=(%pi/4)*dv**2*p; // N (maximum tension in spring)
16 printf('\n maximum tension in spring = %.f N', F)
17 k=(F-Fi)/del; // N/mm (stiffness of spring)
18 printf('\n stiffness of spring = %.2f N/mm', k)
19 //Tmax=F*Dm/2 where Dm=5.5*d
20 Tmax_BY_d=F*5.5/2; // calculation
21 //Tmax=(%pi/16)*d**3*tau_max
22 d=sqrt(Tmax_BY_d/((%pi/16)*tau_max)); // mm
23 printf('\n diameter of spring = %.2f mm. Use 18 mm.'
    ,d)

```

```

24 d=ceil(d); // mm (rounding)
25 Dm=5.5*d; //mm
26 printf('\n mean coil diameter = %.f mm',Dm)
27 Do=Dm+d; //mm
28 printf('\n outside coil diameter = %.f mm',Do)
29 Di=Dm-d; // mm
30 printf('\n initial coil diameter = %.f mm',Di)
31 n=G*10**3*d*del/8/(F-Fi)/C**3; // no. of turns
32 printf('\n no. of turns = %.f',n)
33 nt=n+1; // total no. of turns
34 printf('\n total no. of turns(for extension spring)
    = %.f',nt)
35 gi=1; // mm (initial gap)
36 lf=nt*d+(nt-1)*gi; // mm
37 printf('\n free length of spring = %.f mm',lf)
38 p=lf/(nt-1); //mm
39 printf('\n pitch of coils = %.2f mm',p)

```

---

#### Scilab code Exa 8.12 Find the axial load

```

1 // exa 8.12 Pg 236
2 clc;clear;close;
3
4 // Given Data
5 d=6; //mm
6 Do=75; // mm
7 tau=350; // N/mm.sq.
8 G=84; // kN/mm.sq.
9
10 printf('\n (i) neglecting the effect of curvature')
11 dm=Do-d; // mm
12 C=dm/d; // spring index
13 Ks=1+0.5/C; // shear stress factor
14 //tau=Ks*(8*Fmax*C)/(%pi*d**2)
15 Fmax=tau/(Ks*(8*C)/(%pi*d**2)); // N

```

```

16 printf('\n Axial load = %.1f N',Fmax)
17 delBYi=8*Fmax*C**3/(G*10**3*d); // mm/turn
18 printf('\n deflection per active turn = %.3f mm/turn
    ',delBYi)
19 printf('\n\n (ii) considering the effect of
    curvature ')
20 Kw=(4*C-1)/(4*C-4)+0.615/C; // Wahl's correction
    factor
21 //tau=Kw*(8*Fmax*C)/(G*d)
22 Fmax=tau/(Kw*8*C/(%pi*d**2));
23 printf('\n Axial load = %.1f N',Fmax)
24 delBYn=8*Fmax*C**3/(G*10**3*d); // mm/turn
25 printf('\n deflection per active turn = %.3f mm/turn
    ',delBYn)
26 // Note – answer in the textbook is wrong for last
    part.

```

---

# Chapter 9

## Power Screws

Scilab code Exa 9.1 Stress in the screw

```
1 // exa 9.1 Pg 256
2 clc;clear;close;
3
4 // Given Data
5 d=26; // mm
6 p=5; // mm
7 W=10; // kN
8 Do=50; // mm
9 Di=20; // mm
10 mu=0.2; // coefficient of thread friction
11 mu_c=0.15; // coefficient of collar friction
12 N=15; // rpm
13 pb=6; // MPa
14
15 dm=d-p/2; // mm
16 dc=d-p; // mm
17 t=p/2; //mm
18 l=2*p; // mm
19 alfa=atand(1/(%pi*dm)); // degree
20 fi=atand(mu); // degree
21 Tf=W*dm/2*tand(alfa+fi); // N.mm
```

```

22 Tc=mu_c*W/4*(Do+Di); // N.mm
23 T=Tf+Tc; // N.mm
24 printf('\n (a) Stress in the screw ')
25 sigma_c=4*W*10**3/(%pi*dc**2); // N/mm.sq.
26 printf('\n Direct compressive stress = %.2f N/mm.sq ',
    ,sigma_c)
27 tau=16*T*10**3/(%pi*dc**3); //N/mm.sq.
28 printf('\n Torsional shear stress = %.2f N/mm.sq ',
    tau)
29 tau_max=sqrt(sigma_c**2/4+tau**2); //MPa
30 printf('\n Maximum shear stress = %.2f N/mm.sq ',
    tau_max)
31 n=W*10**3/(%pi*dm*t*pb);
32 printf('\n\n (b) number of threads of nut in
    engagement = %.f ',n)

```

---

### Scilab code Exa 9.2 Power required to drive the screw

```

1 // exa 9.2 Pg 257
2 clc;clear;close;
3
4 // Given Data
5 d=50; // mm
6 p=8; // mm
7 W=2; // kN
8 Do=100; // mm
9 Di=50; // mm
10 mu=0.15; // coefficient of thread friction
11 mu_c=0.10; // coefficient of collar friction
12 N=25; // rpm
13 two_beta=29; // degree
14
15 dm=d-p/2; // mm
16 dc=d-p; // mm
17 t=p/2; //mm

```

```

18 l=2*p; // mm
19 alfa=atand(p/(%pi*dm)); // degree
20 mu_e=mu/cosd(two_beta/2); // virtual coefficient of
    friction
21 fi=atand(mu_e); // degree
22 Tf=W*dm/2*tand(alfa+fi); // N.mm
23 Tc=mu_c*W/4*(Do+Di); // N.mm
24 T=Tf+Tc; // N.mm
25 P=2*%pi*N*T/(60*10**3); // kW
26 printf('\n (a) Power required = %.3f kN',P)
27 To=W*dm/2*tand(alfa); // N.mm
28 eta=To/T*100; // % (efficiency)
29 printf('\n (b) Efficiency of screw = %.2f %%',eta)

```

---

### Scilab code Exa 9.3 Length of handle and shear stress in screw

```

1 // exa 9.3 Pg 259
2 clc;clear;close;
3
4 // Given Data
5 d=10; // mm
6 p=3; // mm
7 mu=0.15; // coefficient of thread friction
8 mu_c=0.20; // coefficient of collar friction
9 dc=15; // mm
10 F=60; // N
11 W=4; // kN
12 two_beta=30; // degree
13 h=25; // mm
14 lf=150; // mm (screw free length)
15
16 dm=d-p/2; // mm
17 alfa=atand(p/(%pi*dm)); // degree
18 mu_e=mu/cosd(two_beta/2); // virtual coefficient of
    friction

```

```

19 fi=atand(mu_e); // degree
20 Tf=W*10**3*dm/2*tand(alfa+fi); // N.mm
21 Tc=mu_c*W*10**3/2*dc; // N.mm
22 T=Tf+Tc; // N.mm
23 //F*l=T
24 l=T/F; // mm (Length of handle)
25 printf('\n (a) Length of handle = %.1f mm',l)
26
27 printf('\n\n (b) Maximum shear stress in screw')
28 printf('\n Section 1-1 : ')
29 dc=d-p; //mm
30 tau=16*T/(%pi*dc**3); // N/mm.sq.
31 M=F*lf; // N.mm
32 sigma_b=32*M/(%pi*dc**3); // N/mm.sq.
33 tau_max=sqrt((sigma_b/2)**2+tau**2); // MPa
34 printf('\n Maximum shear stress = %.2f MPa',tau_max)
35 printf('\n Section 2-2 : ')
36 sigma_c=4*W*10**3/(%pi*dc**2); // N/mm.sq. (Direct
    compressive stress)
37 tau2=16*Tc/(%pi*dc**3); //; // N/mm.sq. (Tortional
    shear stress)
38 tau_max=sqrt((sigma_c/2)**2+tau2**2); // MPa
39 printf('\n Maximum shear stress = %.2f MPa',tau_max)
40
41 //h=n*p; // height of nut
42 n=ceil(h/p); // no. of threads
43 t=p/2; // mm (thickness of threads)
44 pb=W*10**3/(%pi*dm*t*n); // MPa
45 printf('\n\n (b) Bearing pressure on threads = %.1f
    MPa',pb)

```

---

#### Scilab code Exa 9.4 Power required to drive the slide

```

1 // exa 9.4 Pg 260
2 clc;clear;close;

```

```

3
4 // Given Data
5 W=25; // kN
6 two_beta=29; // degree
7 v=0.96; // m/min
8 mu=0.14; // coefficient of thread friction
9 Di=30; // mm
10 Do=66; // mm
11 mu_c=0.15; // coefficient of collar friction
12 d=36; // mm
13 p=6; // mm
14 Sut=630; // MPa
15 Syt=380; // MPa
16
17 dm=d-p/2; // mm
18 dc=d-p; // mm
19 l=2*p; // mm
20 alfa=atand(l/(%pi*dm)); // degree
21 mu_e=mu/cosd(two_beta/2); // virtual coefficient of
    friction
22 fi=atand(mu_e); // degree
23 Tf=W*10**3*dm/2*tand(alfa+fi); // N.mm
24 Tc=mu_c*W*10**3/4*(Do+Di); // N.mm
25 T=Tf+Tc; // N.mm
26 N=v*10**3/l; // rpm
27
28 P=2*%pi*N*T/(60*10**3)*10**-3; // kW
29 printf('\n Power required to drive the slide = %.2f
    kN',P)
30 sigma_c=4*W*10**3/(%pi*dc**2); // MPa
31 tau=16*T/(%pi*dc**3); // MPa
32 sigma1=1/2*(sigma_c+sqrt(sigma_c**2+4*tau**2)); //
    MPa
33 tau_max=sqrt((sigma_c/2)**2+tau**2); // MPa
34 n_t=Syt/sigma1; // factor of safety in tension
35 printf('\n factor of safety in tension = %.2f ',n_t)
36 n_s=Syt/2/tau_max; // factor of safety in shear
37 printf('\n factor of safety in shear = %.2f ',n_s)

```



38 // Note– Answer in the textbook are not accurate.

---

### Scilab code Exa 9.5 Clamping force between jaws

```
1 // exa 9.5 Pg 262
2 clc;clear;close;
3
4 // Given Data
5 d=12; // mm
6 dc=10; // mm
7 p=2; // mm
8 Do=10; //mm
9 mu=0.15; // coefficient of thread friction
10 mu_c=0.18; // coefficient of collar friction
11 F=100; // N
12 l=150; // mm
13
14 dm=dc+p/2; // mm
15 alfa=atand(p/(%pi*dm)); // degree
16 fi=atand(mu); // degree
17 TfByW=dm/2*tand(alfa+fi); // where TfByW = Tf/W
18 TcByW=mu_c/3*Do; // where TcByW = Tc/W
19 TByW=TfByW+TcByW; // N.mm (total torque at B–B)
20 TApplied=F*l; // N.mm (torque applied by the operator
   )
21 //putting T= TApplied
22 W= TApplied/TByW; // N
23 printf('\n (a) Clamping force between the jaws = %.f
   N',W)
24 eta=W*dm/2*tand(alfa)/TApplied*100; // %
25 printf('\n (b) Efficiency of vice = %.2f %%',eta)
26 Tf=TfByW*W; // N.mm
27 printf('\n (c) Torque at A–A, Tf = %.1f N.mm &
   Torque at B–B = %.f N.mm',Tf,TApplied)
28 // Note– Answer in the textbook are not accurate.
```

---

Scilab code Exa 9.6 Design a screw jack

```
1 // exa 9.6 Pg 267
2
3 clc;clear;close;
4
5 // Given Data
6 W=100; // kN
7 lift=400; // mm
8 sigma_ts=100; // MPa
9 sigma_cs=100; // MPa
10 tau_s=60; // MPa
11 tau_tn=50; // MPa
12 sigma_cn=45; // MPa
13 tau_n=40; // MPa
14 pb=15; // MPa
15 mu=0.2; // coefficient of thread friction
16 mu_c=0.15; // coefficient of collar friction
17
18 // sigma_cs=4*W/(%pi*dc**2)
19 dc=sqrt(4*W*10**3/(%pi*sigma_cs)); // mm
20 printf('\n Screw Diameter-\n Core diameter of screw ,
        dc=%0.2f mm. Use dc=40 mm',dc)
21 dc=40; // mm
22 p=7; // mm (for normal series square threads)
23 d=dc+p; //mm
24 printf('\n outside diameter = %.f mm',d)
25 dm=dc+p/2; // mm
26 printf('\n mean diameter = %.1f mm',dm)
27 t=p/2; // mm
28 printf('\n thread thickness = %.1f mm',t)
29
30 printf('\n Maximum tensile & shear stress in screw -
        ')
```

```

31 sigma_c=4*W*1000/%pi/dc**2; // MPa
32 alfa=atand(p/(%pi*dm)); // degree
33 fi=atand(mu); // degree
34 Tf=dm*W*10**3/2*tand(alfa+fi); // where TfByW = Tf/W
35 tau=16*Tf/(%pi*dc**3); // MPa
36 sigma12=(1/2)*(sigma_c+sqrt(sigma_c**2+4*tau**2)); //
    MPa
37 printf('\n Maximum tensile stress = %.f MPa < %.f
    MPA. Hence design is safe.',sigma12,sigma_ts)
38 tau_max=sqrt((sigma_c/2)**2+tau**2); // MPa
39 printf('\n Maximum shear stress = %.2f MPa < %.f MPA
    . Hence design is safe.',tau_max,tau_s)
40
41 printf('\n Height of nut-')
42 n=W*10**3/(%pi/4)/pb/(d**2-dc**2); // no. of threads
43 n= ceil(n); // no. of threads (rounding)
44 h=n*p; // mm
45 printf('\n h=%.f mm',h)
46
47 printf('\n Check for stress in screw and nut')
48 tau_screw=W*10**3/(%pi*n*dc*t); // MPa
49 printf('\n shear stress in screw = %.2f MPa < %.f
    MPa',tau_screw,tau_s)
50 tau_nut=W*10**3/(%pi*n*d*t); // MPa
51 printf('\n shear stress in nut = %.2f MPa < %.f MPa'
    ,tau_nut,tau_n)
52 printf('\n These are within permissible limits.
    Hence design is safe.')
53
54 printf('\n Nut collar size-')
55 // %pi/4*(D1**2-d**2)*sigma_tn=W
56 D1=sqrt(W*10**3/(%pi/4)/tau_tn+d**2); // mm
57 printf('\n Inside diameter of collar = %.2f mm. Use
    D1=70 mm',D1)
58 D1=70; //mm (adopted for design)
59 // %pi/4*(D2**2-D1**2)*sigma_cn=W
60 D2=sqrt(W*10**3/(%pi/4)/sigma_cn+D1**2); // mm
61 printf('\n Outside diameter of collar = %.2f mm. Use

```

```

        D2=90 mm',D2)
62 D2=90; //mm (adopted for design)
63
64 // %pi*D1*tc*tau_n=W
65 tc=W*10**3/(%pi*D1*tau_n); // mm
66 printf('\n thickness of nut = %.2f mm. Use tc=12 mm.
        ',tc)
67 tc=12; // mm (adopted for design)
68
69 printf('\n Head Dimensions-')
70 D3=1.75*d; // mm
71 printf('\n Diameter of head on top of screw = %.2f
        mm. use D3=84 mm. ',D3)
72 D3=84; // mm (adopted for design)
73 D4=D3/4; // mm
74 printf('\n pin diameter in the cup = %.f mm',D4)
75
76 printf('\n Torque required between cup and head-')
77 Tc=mu_c*W*10**3/3*((D3**3-D4**3)/(D3**2-D4**2)); // N
        .mm
78 printf('\n Tc=%.f N.mm (acc. to uniform pressure
        theory) ',Tc)
79 T=Tf+Tc; // N.mm
80 printf('\n Total Torque , T=%.f N.mm',T)
81
82 F=300; // N (as a normal person can apply 100-300 N)
83 l=T/F; //mm
84 printf('\n length of lever = %.f mm. Use 3300 mm',l)
85
86 M=F*l; // N.mm
87 dl=(32*M/%pi/sigma12)**(1/3); // mm
88 printf('\n Diameter of lever , dl=%.1f mm. Use dl=48
        mm. ',dl)
89 dl=48; // mm (adopted for design)
90
91 H=2*dl; // mm
92 printf('\n Height of head , H=%.f mm',H)
93

```

```

94 printf('\n Check for screw in buckling-')
95 L=lift+0.5*h; // mm
96 K=dc/4; // mm
97 C=0.25; // spring index
98 sigma_y=200; // MPa
99 Ac=%pi/4*dc**2; //mm.sq.
100 Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(
    L/K)**2)/1000; // kN
101 printf('\n Buckling or critical load for screw , Wcr
    = %.f kN > 100kN',Wcr)
102
103 To=W*10**3*dm/2*tand(alfa); // N.mm
104 eta=To/T*100; // %
105 printf('\n Efficiency of screw = %.1f %%',eta)
106
107 printf('\n Body dimensions-')
108 D5=1.5*D2; // mm
109 t2=2*tc; // mm
110 t3=0.25*d; //mm
111 D6=2.25*D2; // mm
112 printf('\n Diameter of body at top, D5 = %.f mm', D5
    )
113 printf('\n Thickness of base , t2 = %.f mm', t2)
114 printf('\n Thickness of body, t3 = %.f mm', t3)
115 printf('\n Inside diameter of bottom, D6 = %.1f mm.
    Use D6=205 mm.', D6)
116 D6=205; // mm (adopted for design)
117 D7=1.75*D6; // mm
118 hb=lift+h+100; // mm
119 printf('\n Outside diameter at the bottom, D7 = %.2 f
    mm. Use 360 mm.', D7)
120 printf('\n Height of body = %.f mm. Use 600mm',hb)

```

---

Scilab code Exa 9.7 Torque required to raise and lower the load

```

1 // exa 9.7 Pg 267
2
3 clc;clear;close;
4
5 // Given Data
6 two_beta=30;// degree
7 W=400*10**3;// N
8 d=100;// mm
9 p=12;// mm
10 mu=0.15;// coefficient of thread friction
11
12 dm=d-p/2;// mm
13 dc=d-p;// mm
14 l=2*p;// mm
15 alfa=atand(l/%pi/dm);// degree
16 mu_e=mu/cosd(two_beta/2);// virtual coefficient of
    friction
17 fi=atand(mu);// degree
18 Tf=W*dm/2*tand(alfa+fi);// N.mm (Frictional torque
    for raising load)
19 T=W*dm/4*tand(fi);// N.mm
20 To=W*dm/2*tand(alfa);// N.mm (Torque without
    friction)
21 eta1=To/Tf*100;// %
22 printf('\n Efficiency during raising the load = %.2 f
    %%',eta1)
23 eta2=T/To*100;// %
24 printf('\n Efficiency during lowering the load = %.2
    f %%',eta2)
25 // Note - answer & solution is wrong in the textbook
    .

```

---

### Scilab code Exa 9.9 Safe capacity of press

```

1 // exa 9.9 Pg 272

```

```

2
3 clc;clear;close;
4
5 // Given Data
6 d=70; // mm
7 mu=0.13; // coefficient of thread friction
8 mu_c=0.15; // coefficient of collar friction
9 Do=90; // mm
10 Di=26; // mm
11 L=450; // mm
12 // C-25 steel screw
13 sigma_t1=275; // MPa
14 sigma_c1=275; // MPa
15 tau1=137.5; // MPa
16 // Phosphor bronze nut
17 sigma_t2=100; // MPa
18 sigma_c2=90; // MPa
19 tau2=80; // MPa
20 pb=15; //MPa
21 n=2; // factor of safety
22 //screw
23 sigma_ts=137.5; // MPa
24 sigma_cs=137.5; // MPa
25 tau_s=68.75; // MPa
26 //Nut
27 sigma_tn=50; // MPa
28 sigma_cn=45; // MPa
29 tau_n=40; // MPa
30
31 p=10; // mm (for normal series square threads)
32 dc=d-p; //mm
33 dm=d-p/2; //mm
34 t=p/2; //mm
35 alfa=atand(p/%pi/dm); // degree
36 fi=atand(mu); // degree
37
38 K=dc/4; // mm
39 C=0.25; // spring index

```

```

40 sigma_y=275; // MPa
41 Ac=%pi/4*dc**2; //mm. sq.
42 Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(
    L/K)**2); // N
43 printf('\n (a) Safe Capacity of press or critical
    load for the screw = %.f N',Wcr)
44
45 n=Wcr/(%pi*dm*t*pb); // no. of threads
46 n=ceil(n); // rounding
47 h=n*p; // mm
48 printf('\n (b) Height of nut, h=%.f mm',h)
49
50 W=Wcr; // N
51 Tf=W*dm/2*tand(alfa+fi)/1000; // N.mm (Frictional
    torque)
52 Tc=mu_c*W/4*(Do+Di)/1000; // N.mm (Collar torque)
53 T=Tf+Tc; // N.mm
54 printf('\n (c) Necessary torsional moment or total
    torque = %.2f N.mm',T)
55 // Note - answer in the textbook is wrong.

```

---

#### Scilab code Exa 9.11 Determine the force

```

1 // exa 9.11 Pg 273
2
3 clc;clear;close;
4
5 // Given Data
6 d=26; // mm
7 L=0.25; //m
8 F=300; // N
9 mu=0.14; // coefficient of thread friction
10 p=5; // mm (for normal series)
11
12 dc=d-p; // mm

```



```

13 dm=d-p/2; // mm
14 l=2*p; // mm
15 alfa=atand(l/%pi/dm); // degree
16 fi=atand(mu); // degree
17 To=F*L; // N.m (Torque applied by the operator)
18 // Tf=W*dm/2*tand(alfa+fi); // N.mm
19 // And Tf=To
20 W=To*1000/(dm/2*tand(alfa+fi)); // N
21 printf('The force required for the job is : %.f N',W
    )
22 // Note – answer in the textbook is wrong.

```

---

#### Scilab code Exa 9.13 Design a screw jack

```

1 // exa 9.13 Pg 274
2
3 clc;clear;close;
4
5 // Given Data
6 W=50; // kN
7 lift=200; // mm
8 gc=300; // mm (ground clearance)
9 pb=16; // MPa
10 mu=0.14; // coefficient of collar friction
11
12 //Screw C-35
13 Sut=288; // MPa
14 n=3; // factor of safety for screw
15 // Nut : phosphor-bronze
16 sigma_t=100; // MPa
17 sigma_c=90; // MPa
18 tau=80; // MPa
19 n2=3; // factor of safety for nut
20
21 sigma_ts=Sut/n; // MPa

```

```

22 sigma_cs=Sut/n;// MPa
23 tau_s=sigma_ts/2;// MPa
24 // sigma_cs=4*W/(%pi*dc**2)
25 dc= sqrt(4*W*10**3/(%pi*sigma_cs));// mm
26 printf('\n Screw diameter - \n Core diameter , dc = %
    .2f mm. Use 30 mm',dc)
27 dc=30;// mm (adopted for design)
28 p=6;// mm (for normal series square threads)
29 d=dc+p;//mm
30 printf('\n outside diameter = %.f mm',d)
31 dm=dc+p/2;// mm
32 printf('\n mean diameter = %.1f mm',dm)
33 t=p/2;// mm
34 printf('\n thread thickness = %.1f mm',t)
35
36 printf('\n Maximum tensile & shear tress in screw - '
    )
37 sigma_c=4*W*1000/%pi/dc**2;// MPa
38 alfa=atand(p/(%pi*dm));// degree
39 fi=atand(mu);// degree
40 Tf=dm*W*10**3/2*tand(alfa+fi);// where TfByW = Tf/W
41 tau=16*Tf/(%pi*dc**3);// MPa
42 sigma12=(1/2)*(sigma_c+sqrt(sigma_c**2+4*tau**2));//
    MPa
43 printf('\n Maximum tensile stress = %.1f MPa < %.f
    MPA. Hence design is safe.',sigma12,sigma_ts)
44 tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
45 printf('\n Maximum shear stress = %.2f MPa < %.f MPA
    . Hence design is safe.',tau_max,tau_s)
46
47 printf('\n Height of nut-')
48 n=W*10**3/(%pi/4)/pb/(d**2-dc**2);// no. of threads
49 n= round(n);// no. of threads (rounding)
50 h=n*p;// mm
51 printf('\n h=%f mm',h)
52
53 printf('\n Check for stress in screw and nut')
54 tau_screw=W*10**3/(%pi*n*dc*t);// MPa

```

```

55 printf('\n shear stress in screw = %.2f MPa\' ,
    tau_screw)
56 tau_nut=W*10**3/(%pi*n*d*t); // MPa
57 printf('\n shear stress in nut = %.2f MPa\' ,tau_nut)
58 printf('\n These are within permissible limits.
    Hence design is safe. \' )
59
60 printf('\n Nut collar size-\' )
61 // %pi/4*(D1**2-d**2)*sigma_tn=W
62 D1=sqrt(W*10**3/(%pi/4)/(50)+d**2); // mm
63 printf('\n Inside diameter of collar = %.2f mm. Use
    D1=52 mm\' ,D1)
64 D1=52; //mm (adopted for design)
65 // %pi/4*(D2**2-D1**2)*sigma_cn=W
66 D2=sqrt(W*10**3/(%pi/4)/45+D1**2); // mm
67 printf('\n Outside diameter of collar = %.1f mm. Use
    D2=65 mm\' ,D2)
68 D2=65; //mm (adopted for design)
69
70 // %pi*D1*t_c*tau_cn=W
71 tau_cn=40; // MPa
72 t_c=W*10**3/(%pi*D1*tau_cn); // mm
73 printf('\n thickness of nut = %.2f mm. Use t_c=8 mm. \'
    ,t_c)
74 t_c=8; // mm (adopted for design)
75
76 printf('\n Head Dimensions-\' )
77 D3=1.75*d; // mm
78 printf('\n Diameter of head on top of screw = %.2f
    mm. use D3=64 mm. \' ,D3)
79 D3=64; // mm (adopted for design)
80 D4=D3/4; // mm
81 printf('\n pin diameter in the cup = %.f mm\' ,D4)
82
83 printf('\n Torque required between cup and head-\' )
84 Tc=mu*W*10**3/3*((D3**3-D4**3)/(D3**2-D4**2)); // N.
    mm
85 printf('\n Tc=%.f N.mm (acc. to uniform pressure

```

```

        theory) ',Tc)
86 T=Tf+Tc; // N.mm
87 printf('\n Total Torque , T=%f N.mm',T)
88
89 F=300; // N (as a normal person can apply 100–300 N)
90 l=T/F; //mm
91 printf('\n length of lever = %f mm. Use 1075 mm',l)
92
93 M=F*l; // N.mm
94 dl=(32*M/%pi/sigma12)**(1/3); // mm
95 printf('\n Diameter of lever , dl=%f mm.',dl)
96
97 H=2*dl; // mm
98 printf('\n Height of head , H=%f mm',H)
99
100 printf('\n Check for screw in buckling–')
101 L=lift+0.5*h; // mm
102 K=dc/4; // mm
103 C=0.25; // spring index
104 sigma_y=288; // MPa
105 Ac=%pi/4*dc**2; //mm.sq.
106 Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(
        L/K)**2)/1000; // kN
107 printf('\n Buckling or critical load for screw , Wcr
        = %f kN > 50kN',Wcr)
108 printf('\n Hence design is safe.')

```

---

#### Scilab code Exa 9.14 Torque required to rotate the screw

```

1 // exa 9.14 Pg 278
2
3 clc;clear;close;
4
5 // Given Data
6 d=32; // mm

```

```

7 p=5; // mm
8 W=12; // kN
9 D3=50; // mm
10 D4=20; // mm
11 mu=0.15; // coefficient of thread friction
12 mu_c=0.20; // coefficient of collar friction
13 N=24; // rpm
14 pb=6; // N/mm.sq.
15 tau_s=30; // MPa
16 tau_n=30; // MPa
17
18 dm=d-p/2; // mm
19 dc=d-p; // mm
20 t=p/2; // mm
21 l=2*p; //mm
22 alfa=atand(l/%pi/dm); // degree
23 fi=atand(mu); // degree
24 Tf=W*10**3*dm/2*tand(alfa+fi); // N.mm
25 Tc=mu_c*W*10**3/4*(D3+D4); // N.mm
26 T=Tf+Tc; // N.mm
27 printf('\n (i) Torque required to rotate the screw =
        %.f N.mm',T)
28
29 printf('\n (ii) Stresses induced in screw - ')
30 sigma_c=4*W*10**3/(%pi*dc**2); // N/mm.sq.
31 printf('\n Direct compressive stress = %.2f N/mm.sq'
        ,sigma_c)
32 tau=16*T/(%pi*dc**3); // N/mm.sq.
33 printf('\n Torsional shear stress = %.2f N/mm.sq',
        tau)
34 tau_max=sqrt((sigma_c/2)**2+tau**2); // MPa
35 printf('\n Maximum shear stress = %.2f MPa < %.f MPa
        ',tau_max,tau_s)
36 printf('\n Hence design is safe.')
37 n=W*10**3/(%pi*dm*t*pb); // no. of threads
38 n=ceil(n); // rounding
39 h=n*p; //mm
40 printf('\n (iii) Height of nut = %.f mm',h)

```

---

Scilab code Exa 9.15 Design a simple screw jack

```
1 // exa 9.15 Pg 279
2
3 clc;clear;close;
4
5 // Given Data
6 W=100; // kN
7 lift=260; // mm
8 pb=15; // N/mm.sq.
9 mu=0.15; // coefficient of thread friction
10 mu_c=0.20; // coefficient of collar friction
11 //Screw
12 Suts=800; // N/mm.sq.
13 sigma_ss=340; // N/mm.sq.
14 ns=4; // factor of safety
15 //Nut
16 Sutn=552; // N/mm.sq.
17 sigma_sn=260; // N/mm.sq.
18 nn=5; // factor of safety
19
20 sigma_ts=Suts/ns; // N/mm.sq.
21 sigma_cs=Suts/ns; // N/mm.sq.
22 tau_s=sigma_ss/ns; // N/mm.sq.
23 sigma_tn=Sutn/nn; // N/mm.sq.
24 sigma_cn=Sutn/nn; // N/mm.sq.
25 tau_n=sigma_sn/nn; // N/mm.sq.
26
27 //sigma_cs=4*W/(%pi*dc**2)
28 dc=sqrt(4*W*10**3/(%pi*sigma_cs)); // mm
29 printf('\n Screw Diameter-\n Core diameter of screw ,
        dc=%0.2f mm. Use dc=33 mm',dc)
30 dc=33; // mm
31 p=7; // mm (for normal series square threads)
```

```

32 d=dc+p;//mm
33 printf('\n outside diameter = %.f mm',d)
34 dm=dc+p/2;// mm
35 printf('\n mean diameter = %.1f mm',dm)
36 t=p/2;// mm
37 printf('\n thread thickness = %.1f mm',t)
38
39 printf('\n Maximum stresses in screw -')
40 sigma_c=4*W*1000/%pi/dc**2;// MPa
41 alfa=atand(p/(%pi*dm));// degree
42 fi=atand(mu);// degree
43 Tf=dm*W*10**3/2*tand(alfa+fi);// where TfByW = Tf/W
44 tau=16*Tf/(%pi*dc**3);// MPa
45 sigma12=(1/2)*(sigma_c+sqrt(sigma_c**2+4*tau**2));//
    MPa
46 printf('\n Maximum tensile stress = %.1f N/mm.sq. <
    %.f N/mm.sq.. Hence design is safe.',sigma12,
    sigma_ts)
47 tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
48 printf('\n Maximum shear stress = %.2f N/mm.sq. < %.
    f N/mm.sq.. Hence design is safe.',tau_max,tau_s)
49
50 printf('\n Height of nut-')
51 n=W*10**3/(%pi/4)/pb/(d**2-dc**2);// no. of threads
52 n= ceil(n);// no. of threads (rounding)
53 h=n*p;// mm
54 printf('\n h=%.f mm. Use 120 mm.',h)
55 h=120;// mm
56
57 printf('\n Check for stress in screw and nut')
58 tau_screw=W*10**3/(%pi*n*dc*t);// MPa
59 printf('\n shear stress in screw = %.2f MPa < %.f
    MPa',tau_screw,tau_s)
60 tau_nut=W*10**3/(%pi*n*d*t);// MPa
61 printf('\n shear stress in nut = %.2f MPa < %.f MPa'
    ,tau_nut,tau_n)
62 printf('\n These are within permissible limits.
    Hence design is safe.')

```

```

63
64 printf('\n Nut collar size-')
65 // %pi/4*(D1**2-d**2)*sigma_tn=W
66 D1=sqrt(W*10**3/(%pi/4)/sigma_tn+d**2); // mm
67 printf('\n Inside diameter of collar = %.2 f mm. Use
        D1=55 mm',D1)
68 D1=55; //mm (adopted for design)
69 // %pi/4*(D2**2-D1**2)*sigma_cn=W
70 D2=sqrt(W*10**3/(%pi/4)/sigma_cn+D1**2); // mm
71 printf('\n Outside diameter of collar = %.2 f mm. Use
        D2=70 mm',D2)
72 D2=70; //mm (adopted for design)
73
74 // %pi*D1*tc*tau_n=W
75 tc=W*10**3/(%pi*D1*tau_n); // mm
76 printf('\n thickness of nut = %.f mm. Use tc=15 mm.'
        ,tc)
77 tc=15; // mm (adopted for design)
78
79 printf('\n Head Dimensions-')
80 D3=1.75*d; // mm
81 printf('\n Diameter of head on top of screw = %.2 f
        mm. ',D3)
82 D4=D3/4; // mm
83 printf('\n pin diameter in the cup = %.1 f mm. Use 20
        mm. ',D4)
84 D4=20; // mm (adopted for design)
85
86 printf('\n Torque required between cup and head-')
87 Tc=mu_c*W*10**3/3*((D3**3-D4**3)/(D3**2-D4**2)); // N
        .mm
88 printf('\n Tc=%f N.mm (acc. to uniform pressure
        theory)',Tc)
89 T=Tf+Tc; // N.mm
90 printf('\n Total Torque , T=%f N.mm',T)
91
92 F=300; // N (as a normal person can apply 100-300 N)
93 l=T/F; //mm

```



```

94 printf('\n length of lever = %.f mm or %.2f m',l,l
    /1000)
95
96 M=F*l;// N.mm
97 sigma_b=100;// N/mm.sq. (assumed)
98 dl=(32*M/%pi/sigma_b)**(1/3);// mm
99 printf('\n Diameter of lever , dl=%.1f mm. Use dl=45
    mm. ',dl)
100 dl=45;// mm (adopted for design)
101
102 H=2*dl;// mm
103 printf('\n Height of head , H=%.f mm',H)
104
105 printf('\n Check for screw in buckling-')
106 L=lift+0.5*h;// mm
107 K=dc/4;// mm
108 C=0.25;// spring index
109 sigma_y=200;// MPa
110 Ac=%pi/4*dc**2;//mm.sq.
111 Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3)))*(
    L/K)**2)/1000;// kN
112 printf('\n Buckling or critical load for screw , Wcr
    = %.f kN > 100kN',Wcr)
113
114 To=W*10**3*dm/2*tand(alfa);// N.mm
115 eta=To/T*100;// %
116 printf('\n Efficiency of screw = %.2f %%',eta)
117
118 printf('\n Body dimensions-')
119 D5=1.5*D2;// mm
120 t2=2*tc;// mm
121 t3=0.25*d;//mm
122 D6=2.25*D2;// mm
123 printf('\n Diameter of body at top , D5 = %.f mm', D5
    )
124 printf('\n Thickness of base , t2 = %.f mm', t2)
125 printf('\n Thickness of body , t3 = %.f mm', t3)
126 printf('\n Inside diameter of bottom , D6 = %.1f mm.

```

```
Use D6=160 mm. ', D6)
127 D6=160; // mm (adopted for design)
128 D7=1.75*D6; // mm
129 hb=lift+h+100; // mm
130 printf('\n Outside diameter at the bottom, D7 = %.2 f
mm. ', D7)
131 printf('\n Height of body = %.f mm. ',hb)
```

---