Scilab Textbook Companion for Machine Design-i by Dr. Sadhu Singh[1]

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

July 30, 2019

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

Book Description

Title: 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

Chapter 1

Introduction

Scilab code Exa 1.1 Preferred Speeds

```
// exa 1.1 Pg 13
clc;clear;close;
Nmax=1000;// rpm
Nmin=30;// rpm
z=9;// no. of steps

//Rn=Nmax/Nmin=fi**(z-1)
fi=(Nmax/Nmin)**(1/(z-1));// common ratio

printf('The speeds of gear box are:')
N1=Nmin;// rpm
for i=1:z
    printf('\n\t\t\tN%d = %.1f rpm',i,N1)
    N1=fi*N1;//rpm
end;
```

Scilab code Exa 1.2 Capacities of models

```
// exa 1.2 Pg 14
clc;clear;close;
Pmax=100;// kW
Pmin=10;// kW
z=5;// no. of models

//Rn=Pmax/Pmin=fi**(z-1)
fi=(Pmax/Pmin)**(1/(z-1));// common ratio

printf('The power of generating sets are:')
P1=Pmin;// kW
for i=1:z
    printf('\n\t\t\tP%d = %.1f kW',i,P1)
    P1=fi*P1;//kW
end;

printf('\nExpanding for 10 models.');
z=10;// no. of models

fi=(Pmax/Pmin)**(1/(z-1));// common ratio

printf('\nThe power of generating sets are:')
P1=Pmin;// kW
for i=1:z
    printf('\n\t\t\tP%d = %.1f kW',i,P1)
    P1=fi*P1;//kW
end;
```

Scilab code Exa 1.4 `Capacities of additional models`

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

```

//Rn=Pmax/Pmin=fi**(z-1)
fi=(Pmax/Pmin)**(1/(z-1));// common ratio

printf('The models are:')

for i=0:z-1
    P1=fi**(i)*Pmin;// kW
    printf('\n\t\t\tP%d = %.1f kW',i,P1)
end;

printf('\n for 8 models.')

z=8;// no. of models

//Rn=Pmax/Pmin=fi**(z-1)
fi=(Pmax/Pmin)**(1/(z-1));// common ratio

printf('The models are:')

for i=0:z-1
    P1=fi**(i)*Pmin;// kW
    printf('\n\t\t\tP%d = %.1f kW',i,P1)
end;
```

Scilab code Exa 1.6 Power capacity of models

```
// exa 1.6 Pg 15
clc;clear;close;
Pmax=75;// kW
Pmin=7.5;// kW
z=5;// no. of models

//Rn=Pmax/Pmin=fi**(z-1)
fi=(Pmax/Pmin)**(1/(z-1));// common ratio
```

```
printf('The models are:')

for i=0:z-1
    P1=fi**(i)*Pmin;// kW
    printf('\n\t\t\tP%d = %.1f kW',i,P1)
end;
```

Chapter 3

Design Against Static Load

Scilab code Exa 3.1 Dimension of cross section

```
// exa 3.1 Pg 62

clc;clear;close;

// Given Data
P=30;// kN
Sut=350;// MPa
n=2.5;// factor of safety

sigma_w=Sut/n;// MPa (Working stress for the link)

t=poly(0,'t');// thickness of link
A=2.5*t**2;// mm.sq.
I=t*(2.5*t)**3/12;// mm^4 (Moment of Inertia about N-A)
sigma_d=P/A;// N/mm.sq.
e=10+1.25*t;//mm
M=P*10**3*e;// N.mm
sigma_t=M*1.25*t/I;// N/mm.sq.
//maximum tensile stress at the top fibres = sigma_d+sigma_t=sigma_w ...eqn(1)
```

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

Scilab code Exa 3.2 Dimension of cross section

```
// exa 3.2 Pg 63

clc;clear;close;

// Given Data
P=6;// kN
alfa=30;// degree
Sut=250;// MPa
n=2.5;// factor of safety

sigma_w=Sut/n;// MPa (Working stress for the link)
PH=P*10**3*cosd(alfa);// kN
PV=P*10**3*sind(alfa);// kN

t=poly(0,'t');// thickness of link
A=2*t*t;// mm.sq.
sigma_d=PH/A;// N/mm.sq.
M=PH*100+PV*250;// N.mm
I=t*(2*t)**3/12;// mm^4 (Moment of Inertia)
sigma_t=M*t/I;// N/mm.sq.
//maximum tensile stress at the top fibres = sigma_d
    +sigma_t=sigma_w   ...eqn(1)
expr=sigma_d+sigma_t-sigma_w ;// expression of
    polynomial from above eqn.
```

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

Scilab code Exa 3.3 Dimension of cross section

```
// exa 3.3 Pg 64

clc;clear;close;

// Given Data
P=20;// kN
Sut=300;// MPa
n=3;// factor of safety

sigma_w=Sut/n;// MPa (Working stress for the link)

t=poly(0,'t');// thickness of link
A=4*t*t;// mm.sq.
sigma_d=P*10**3/A;// N/mm.sq.
e=6*t;//mm
M=P*10**3*e;// N.mm
z=t*(4*t)**2/6;// mm^3 (section modulus at x1-x2)
sigma_b=M/z;// N/mm.sq.
//maximum tensile stress at x1 = sigma_d+sigma_b=sigma_w ...eqn(1)
expr=sigma_d+sigma_b-sigma_w ;// expression of polynomial from above eqn.
t=roots(numer(expr));// solving the equation (as denominator will me be multiplied by zero on R.H.S)
t=t(2);// mm // discarding -ve roots
```

```
printf('dimension of cross section of link , t=%.2f
    mm. Use 23 mm.',t)
```

Scilab code Exa 3.4 Dimension of cross section

```
// exa 3.4 Pg 65

clc;clear;close;

// Given Data
P=15;// kN
sigma_t=20;// MPa
sigma_c=60;// MPa
n=3;// factor of safety

a=poly(0,'a');// from the diagram.
// Area of cross section
A1=2*a*a;// mm.sq.
A2=2*a*a/2;// mm.sq.
A=A1+A2;// mm.sq.

// Location of neutral axis
//3*a**2*y_bar=2*a**2*a/2+a**2*(a+a/2)
y_bar=(2*a**2*a/2+a**2*(a+a/2))/(3*a**2);// mm

// Moment of Inertia about neutral axis N-A
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
yt=y_bar;//mm
yc=2*a-y_bar;// mm
e=y_bar-0.5*a;//mm
M=P*10**3*e;// N.mm
sigma_d=P*10**3/A;// N/mm.sq.
sigma_t1=M*yt/I;//N/mm.sq.
sigma_c1=M*yc/I;//N/mm.sq.
```

```
sigma_r_t=sigma_d+sigma_t1;// N/mm.sq. (sigma_r_t=
    resultant tensile stress at AB=sigma_d+sigma_t)
sigma_r_c=sigma_c1-sigma_d;// N/mm.sq. (sigma_r_t=
    resultant tensile stress at AB=sigma_d+sigma_t)

//equating resulting tensile stress with given value
     sigma_t-sigma_r_t=0...eqn(1)
expr1=sigma_t-sigma_r_t;// expression of polynomial
    from above eqn.
a1=roots(numer(expr1));// solving the equation (as
    denominator will me be multiplied by zero on R.H.
    S)
a1=a1(2);// mm // discasrding -ve roots
printf('Equating resultant tensile stress gives, a =
     %.2f mm',a1)

//equating resulting compressive stress with given
    value sigma_c-sigma_c_t=0...eqn(1)
expr2=sigma_c-sigma_r_c;// expression of polynomial
    from above eqn.
a2=roots(numer(expr2));// solving the equation (as
    denominator will me be multiplied by zero on R.H.
    S)
a2=a2(2);// mm // discarding -ve roots
printf('\n Equating resultant compressive stress
    gives, a = %.2f mm',a2)
a=ceil(a1);//mm
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

```
// exa 3.5 Pg 67

clc;clear;close;
```

```

// Given Data
Syt=760;// MPa
M=15;// kN.m
T=25;//kN.m
n=2.5;// factor of safety
E=200;// GPa
v=0.25;// Poisson's ratio

sigma_d=Syt/n;// MPa
// let d is diameter of the shaft
sigma_b_into_d_cube=32*M*10**6/%pi;// N/mm.sq. (where sigma_b_into_d_cube = sigma_d*d**3)
tau_into_d_cube=16*T*10**6/%pi//d**3;// N/mm.sq. (where tau_into_d_cube = tau*d**3)
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)
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)
printf('\n (i) Maximum shear stress theory')
tau_max_into_d_cube=(sigma1_into_d_cube-sigma2_into_d_cube)/2; //(where tau_max_into_d_cube = tau_max*d**3)
d=(tau_max_into_d_cube/(sigma_d/2))**(1/3);//mm
printf('diameter of shaft, d=%.1f mm or %.f mm',d,ceil(d))

printf('\n (ii) Maximum strain energy theory')
//sigma1**2+sigma2**2-2*v*sigma1*sigma2=sigma_d**2
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)
printf('diameter of shaft, d=%.1f mm',d)
printf('\n Adopt d=100mm')
```

Scilab code Exa 3.6 Diameter of the shaft

```
// exa 3.6 Pg 69

clc;clear;close;

// Given Data
N=200;// rpm
P=200;// kW
tau_d=42;// Mpa
W=900;// N
L=3;// m
sigma_t=56;// MPa
sigma_c=56;// MPa

T=P*60*10**3/(2*%pi*N);// N.m
M=W*L/4;// N.m
Te=sqrt(M**2+T**2);// N.m
//Te=(%pi/16)*d**3*tau_d
d=(Te/((%pi/16)*tau_d)*1000)**(1/3);// mm
printf('\n Using equivalent torque equation,\n shaft
    diameter d = %.f mm',d)

Me=(1/2)*(M+sqrt(M**2+T**2));// N.m
//Me=(%pi/32)*d**3*sigma_d
d=(Me/((%pi/32)*sigma_c)*10**3)**(1/3);//mm
printf('\n Using equivalent bending moment equation
   ,\n shaft diameter d = %.2f mm or %.f mm',d, ceil
   (d))
printf('\n Adopt d=105 mm.')
```

Scilab code Exa 3.8 Suitable diameter of the shaft

```
// exa 3.8 Pg 70

clc;clear;close;

// Given Data
M=15;// N.m
P=5;// kW
N=500;// rpm
tau_d=40;// Mpa
sigma_d=58;// MPa

T=P*60*10**3/(2*%pi*N);// N.m
Te=sqrt(M**2+T**2);// N.m
//Te=(%pi/16)*d**3*tau_d
d=(Te/((%pi/16)*tau_d)*1000)**(1/3);// mm
printf('\n Using equivalent torque equation,\n shaft
    diameter d = %.f mm',d)

Me=(1/2)*(M+sqrt(M**2+T**2));// N.m
//Me=(%pi/32)*d**3*sigma_d
d=(Me/((%pi/32)*sigma_d)*10**3)**(1/3);//mm
printf('\n Using equivalent bending moment equation
   ,\n shaft diameter d = %.2f mm or %.f mm',d, ceil
   (d))
printf('\n Adopt d=23 mm.')
```

Scilab code Exa 3.10 Maximum value of torque

```
// exa 3.10 Pg 71

clc;clear;close;

// Given Data
d=4;// cm
M=15000;// N.cm
```

```
Syt=20000;// N/cm.sq.

printf('\n (i) Maximum Principal Stress Theory-')
z=%pi*d**3/32;// cm.cube.
sigma_b=M/z;// N/cm.sq.
T=poly(0,'T')
tau=16*T/(%pi*d**3);// N/cm.sq.
//sigma1=(1/2)*(sigma_b+sqrt(sigma_b**2+4*tau**2)) // Maximum principal stress
//sigma1=(sigma_b/2+sqrt(sigma_b**2/4+tau**2)) // on solving
//tau=sqrt((sigma1-sigma_b/2)**2-sigma_b**2/4)
sigma1=Syt;// N/cm.sq.
T=sqrt((sigma1-sigma_b/2)**2-sigma_b**2/4)*(%pi*d**3)/16;// N.cm.
printf('\n Maximum value of torque, T = %.f N.cm.',T)

printf('\n (ii) Maximum Shear Stress Theory')
tau_d=0.5*Syt;// N.cm.
//Te=sqrt(M**2+T**2)=(%pi/16)*d**3*tau_d
T=sqrt(((%pi/16)*d**3*tau_d)**2-M**2);// N.cm.
printf('\n Maximum value of torque, T = %.f N.cm.',T)
// Answer in the textbook is not accurate.
```

Scilab code Exa 3.11 Diameter of the shaft

```
// exa 3.11 Pg 72

clc;clear;close;

// Given Data
N=200;// rpm
P=25;// kW
```

```
tau_d=42;// MPa
W=900;// N
L=3;// m
Syt=56;// MPa
Syc=56;// MPa
sigma_d=56;// MPa

T=P*60*10**3/(2*%pi*N);// N.m
M=W*L/4;// N.m
Te=sqrt(M**2+T**2);// N.m
// Te=(%pi/16)*d**3*tau_d
d=(Te*10**3/((%pi/16)*tau_d))**(1/3);// mm
printf('\n shaft diameter(using equivalent torque)-\n d=%.f mm.',d)

Me=(1/2)*(M+sqrt(M**2+T**2));//N.m
// Me=(%pi/32)*d**3*sigma_d
d=(Me*10**3/((%pi/32)*sigma_d))**(1/3);// mm
printf('\n shaft diameter(using equivalent bending moment)-\n d=%.f mm.',d)
printf('\n adopt d=57 mm.')
```

Scilab code Exa 3.12 Cross section of hanger

```
// exa 3.12 Pg 72

clc;clear;close;

// Given Data
sigma_w=60;// MPa
F=10;// kN
alfa=30;// degree

FH=F*sind(alfa);// kN
FV=F*cosd(alfa);// kN
```

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

Scilab code Exa 3.13 Diameter of the shaft

```
// exa 3.13 Pg 74

clc;clear;close;

// Given Data
P=15;// kW
n1=200;// rpm
l=600;// mm
z2=18;// no. of teeth
m2=5;// mm
alfa2=14.5;// degree
l2=120;// mm
z1=72;// no. of teeth
m1=5;// mm
```

```
alfa1=14.5;// degree
l1=150;// mm
sigma_d=80;// MPa

d1=m1*z1;// mm
v1=%pi*d1*n1/(60*10**3);// m/s
Ft1=10**3*P/v1;// N (outwards)
Fr1=Ft1*tand(alfa1);// N (Downwards)
d2=m2*z2;// mm
v2=%pi*d2*n1/(60*10**3);// m/s
Ft2=10**3*P/v2;// N (outwards)
Fr2=Ft2*tand(alfa2);// N (Upwards)

// RAV*600=Fr1*450+Fr2*120 (Taking moments about bearing B)
RAV=(Fr1*450+Fr2*120)/600;// N (Downwards)
RBV=(Fr1-Fr2-RAV);// N (upwards)
MCV=RAV*l1;// N.mm
MBV=Fr2*l2;// N.mm

// RAH*600=-Ft1*450+Ft2*120 (Taking moments about bearing B)
RAH=(-Ft1*450+Ft2*120)/600;// N (Outwards)
RBH=Ft1+Ft2+RAH;// N (inwards)
MCH=RAH*l1;// N.mm
MBH=Ft2*l2;// N.mm

// Resultant Bending Moments
MC=sqrt(MCV**2+MCH**2);// N.mm
MB=sqrt(MBV**2+MBH**2);// N.mm
Mmax=max(MC,MB);// N.mm
T=10**3*P/(2*%pi*n1);// N.m
Me=(1/2)*(Mmax+sqrt(Mmax**2+T**2));// N.mm
// Me=(%pi/32)*d**3*sigma_d
d=(Me/((%pi/32)*sigma_d))**(1/3)
printf('\n shaft diameter is : %.f mm',d)
```

Chapter 4

Design Against Fluctuating Load

Scilab code Exa 4.1 Thickness of plate

```
// exa 4.1 Pg 102
clc;clear;close;
P=6;// kN

//dimensions of plate
r=5;//mm
d=40;//mm
D=50;//mm
d0=10;//mm
w=40;//mm
Sut=200;//MPa
n=2.5;// factor of safety

//Fillet -
rBYd=r/d;
DBYd=D/d;
Kt=1.75;// factor
printf('for stepped plate under tension, Kt=%.2f for
    r/d = %.3f & D/d = %.2f ',Kt,rBYd,DBYd)
```

```
t=poly(0,'t')
sigma_max = Kt*P/t;// N per mm sq.

// Hole -
d0BYw=d0/w;
Kt=2.42;// factor
printf('\n for finite width plate under tension with a hole, Kt=%.2f for d0/w = %.2f',Kt,d0BYw)
sigma_max_into_t = Kt*P/(w-d0);//N/mm sq.

//Design stress
sigma_d = Sut/n;// MPa
//putting sigma_max=sigma_d
t=sigma_max_into_t/sigma_d*1000;// mm
printf('\n Thickness of plate = %.2f mm or %.f mm',t,t)
```

Scilab code Exa 4.2 Diameter of the axle

```
// exa 4.2 Pg 104
clc;clear;close;

// Given Data
rBYd=0.1;
DBYd=1.2;
P=3;// kN
Syt=300;//MPa
n=3;// factor of safety
//dimensions of plate
l1=400;//mm
l2=300;//mm
l3=400;//mm


sigma_d=Syt/n;// MPa
```

```
Kt=1.65;// factor for circular fillet radius member
Rp=P/2;//kN (bearing reaction due to symmetry)
Mf=Rp*l1;// kN.mm (bending moment at fillet)
Mc=P*(l1+l2+l3)/4;// kN.mm (bending moment at centre
    )

//Fillet
//sigma_max=Kt*32*Mf/(%pi*d**3)
sigma_max_into_d_cube_1 = Kt*32*Mf*1000/%pi


//Centre
//sigma_max=32*Mc/(%pi*d**3)
sigma_max_into_d_cube_2 = Kt*32*Mf*1000/%pi
sigma_max_into_d_cube=max(sigma_max_into_d_cube_1,
    sigma_max_into_d_cube_2);// (getting max)

//putting sigma_max=sigma_d
t=(sigma_max_into_d_cube/sigma_d)**(1/3);// mm
printf('\n Diameter of axle = %.1f mm',t)
```

Scilab code Exa 4.3 Endurance Limit

```
// exa 4.3 Pg 105
clc;clear;close;

// Given Data
Sut=440;//MPa
d=25;//mm
R=95/100;// reliability
Kt=1.8;// stress concentration factor
q=0.86;// sensitivity factor

Se_dash = 0.5*Sut;// MPa

```

```
// for machined surface
ka=0.82;// surface finish factor
kb=0.85;// size factor
kc=0.868;// reliability factor
kd=1;// temperature factor
ke=0.577;// load factor

Kf=1+q*(Kt-1);// fatigue strength factor
kf=1/Kf ;// fatigue strength reduction factor
Se=ka*kb*kc*kd*ke*kf*Se_dash;// (MPa) Endurance
    limit
printf('\n Endurance limit = %.2f MPa',Se)
```

Scilab code Exa 4.4 Thickness of plate

```
// exa 4.4 Pg 105
clc;clear;close;

// Given Data
Sut=440;//MPa
w=60;//mm
d=12;// mm
P=20;// kN
q=0.8;// sensitivity factor
R=90/100;// reliability
n=2;// factor of safety

Kt=2.52;// stress concentration factor
Se_dash = 0.5*Sut;// MPa
// for hot rollednormalized condition
ka=0.67;// surface finish factor
kb=0.85;// size factor (assuming t<50 mm)
kc=0.897;// reliability factor
kd=1;// temperature factor
ke=0.9;// load factor
```

```
dBYw=d/w; //(for circular hole)

Kf=1+q*(Kt-1);// fatigue strength factor
kf=1/Kf ;// fatigue strength reduction factor
Se=ka*kb*kc*kd*ke*kf*Se_dash;// (MPa) Endurance limit
sigma_d=Se/n;// MPa (design stress)
// sigma_max=P/(w-d)/t
sigma_max_into_t = P*1000/(w-d);
// putting sigma_max=sigma_d
t=sigma_max_into_t/sigma_d;// mm
printf('\n Thickness of plate = %.2f mm or 20 mm',t)
```

Scilab code Exa 4.5 Endurance strength of specimen

```
// exa 4.5 Pg 107
clc;clear;close;

// Given Data
Sut=650;//MPa
N=10**5;// cycles
Se_dash = 0.5*Sut;// MPa
of=5;// unit
ob=6;//unit
bf=ob-of;// unit
be=3;//unit

// calculating endurance section wise
OE=log10(Se_dash);
OA=log10(0.9*Sut);
AE=OA-OE;
//log10_Sf=OD=OE+ED=OE+FC
log10_Sf=OE+(bf/be)*AE;
Sf=10**log10_Sf; // (MPa) Endurance
printf('\n Endurance of specimen = %.2f MPa',Sf)
```

Scilab code Exa 4.6 Diameter of the beam

```
// exa 4.6 Pg 108
clc;clear;close;

// Given Data
Sut=540;//MPa
N=10**4;// cycles
q=0.85;// sensitivity factor
R=90/100;// reliability
P=1500;// N
l=160;// mm

Se_dash = 0.5*Sut;// MPa
// for cold drawn steel
ka=0.79;// surface finish factor
kb=0.85;// size factor (assuming t<50 mm)
kc=0.897;// reliability factor
kd=1;// temperature factor
ke=1;// load factor

Kt=1.33;// under bending

Kf=1+q*(Kt-1);// fatigue strength factor
kf=1/Kf ;// fatigue strength reduction factor
Se=ka*kb*kc*kd*ke*kf*Se_dash;// MPa( Endurance limit
    )

of=4;// unit
ob=6;//unit
bf=ob-of;// unit
be=3;//unit

// calculating endurance section wise
```

```
OE=log10(Se);
OA=log10(0.9*Sut);
AE=OA-OE;
//log10_Sf=OD=OE+ED=OE+FC
log10_Sf=OE+(bf/be)*AE;
Sf=10**log10_Sf; // (MPa) Endurance

MB=P*l;// N.mm
// 32*MB/%pi/d**3 = Sf
d=(32*MB/%pi/Sf)**(1/3)
printf('\n diameter of beam %.f mm',d)
```

Scilab code Exa 4.7 Diameter d at fillet cross section

```
// exa 4.7 Pg 110
clc;clear;close;

// Given Data
Sut=600;//MPa
Syt=380;//MPa
q=0.9;// sensitivity factor
R=90/100;// reliability
n=2;// factor of safety
Pmin=-100;// N
Pmax=200;// N
l=150;// mm

Se_dash = 0.5*Sut;// MPa
// for cold drawn steel
ka=0.76;// surface finish factor
kb=0.85;// size factor (assuming t<50 mm)
kc=0.897;// reliability factor
kd=1;// temperature factor
ke=1;// load factor

```

```
Kt=1.4;// under bending

Kf=1+q*(Kt-1);// fatigue strength factor
kf=1/Kf ;// fatigue strength reduction factor
Se=ka*kb*kc*kd*ke*kf*Se_dash;// MPa( Endurance limit )
Mmax=Pmax*l;// N.mm
Mmin=Pmin*l;// N.mm
Mm=(Mmax+Mmin)/2;// N.mm
Ma=(Mmax-Mmin)/2;// N.mm
theta=atand(Ma/Mm);// degree

//equation of Goodman - sigma_m/Sut+sigma_a/Se=1
//here sigma_a/sigma_m=3
sigma_m=1/(1/Sut+3/Se);//MPa
sigma_a=3*sigma_m;// MPa

sigma_da=sigma_a/n;// MPa
//sigma_da=32*Ma/%pi/d**3
d=(32*Ma/%pi/sigma_da)**(1/3);// mm
printf('\n diameter d at fillet cross section = %.f mm',d)
```

Scilab code Exa 4.8 Diameter of the shaft

```
// exa 4.8 Pg 112
clc;clear;close;

// Given Data
Sut=500;//MPa
Syt=300;//MPa
R=90/100;// reliability
n=2;// factor of safety
Tmin=-200;// N.m
Tmax=500;// N.m
```

```
Se_dash = 0.5*Sut;// MPa
// for cold drawn steel
ka=0.80;// surface finish factor
kb=0.85;// size factor (assuming t<50 mm)
kc=0.897;// reliability factor
kd=1;// temperature factor
ke=0.577;// load factor

Ses=ka*kb*kc*kd*ke*Se_dash;// MPa( Endurance limit)
Sys=ke*Syt;// MPa
Tm=(Tmax+Tmin)/2;// N.m
Ta=(Tmax-Tmin)/2;// N.m
theta=atand(Ta/Tm);// degree
Sms=Ses/tand(theta);//MPa
Sas=Ses;//MPa
tau_da=Sas/n;//MPa
//tua_da=16*Ta/%pi/d**3
d=(16*Ta*1000/%pi/tau_da)**(1/3);//mm
printf('\n diameter of shaft = %.f mm',d)
```

Scilab code Exa 4.9 Life of the spring

```
// exa 4.9 Pg 113
clc;clear;close;

// Given Data
Sut=860;//MPa
Syt=690;//MPa
Pmin=60;// N
Pmax=120;// N
R=50/100;// reliability
l=500;//mm
d=10;//mm
Se_dash = 0.5*Sut;// MPa
```

```
// for machines surface
ka=0.70;// surface finish factor
kb=0.85;// size factor (assuming t<50 mm)
kc=1;// reliability factor
kd=1;// temperature factor
ke=1;// load factor

Se=ka*kb*kc*kd*ke*Se_dash;// MPa( Endurance limit)
Mmax=Pmax*l;// N.mm
Mmin=Pmin*l;// N.mm
Mm=(Mmax+Mmin)/2;// N.mm
Ma=(Mmax-Mmin)/2;// N.mm
Sm=32*Mm/%pi/d**3;//MPa
sigma_m=Sm;//MPa
Sa=32*Ma/%pi/d**3;//MPa
sigma_a=Sa;//MPa
Sf=Sa*Sut/(Sut-Sm);//MPa

//calculating section
OB=6;//unit ref. o at 3
BE=OB-3;//unit
OC=Sf;// MPa
AE=log10(0.9*Sut)-log10(Se);//MPa
AC=log10(0.9*Sut)-log10(Sf);//MPa
CD=BE*AC/AE;//
//log10(N)=3+CD
N=10**(3+CD);// cycle
printf('\n life of the spring , N = %.f cycles',N)
//Note : answer in the textbook is wrong.
```

Scilab code Exa 4.10 factor of safety

```
// exa 4.10 Pg 116
clc;clear;close;

```

```
// Given Data
Sut=600;//MPa
Se=280;//MPa
sigma_x_min=50;// MPa
sigma_x_max=100;// MPa
sigma_y_min=20;// MPa
sigma_y_max=70;// MPa

sigma_xm=(sigma_x_max+sigma_x_min)/2;// MPa
sigma_xa=(sigma_x_max-sigma_x_min)/2;// MPa
sigma_ym=(sigma_y_max+sigma_y_min)/2;// MPa
sigma_ya=(sigma_y_max-sigma_y_min)/2;// MPa

// distortion energy theory -
sigma_m=sqrt(sigma_xm**2+sigma_ym**2-sigma_xm*sigma_ym);// MPa
sigma_a=sqrt(sigma_xa**2+sigma_ya**2-sigma_xa*sigma_ya);// MPa
theta=atand(sigma_a/sigma_m);// degree
// Sm/Sut+Sa/Se=1 where Sa=Sm*tan(theta)
Sm=1/(1/Sut+tand(theta)/Se);// MPa
Sa=tand(theta)*Sm;// MPa
n=Sa/sigma_a;// factor of safety

printf('\n factor of safety , n = %.2f',n)
```

Scilab code Exa 4.11 Diameter of the shaft

```
// exa 4.11 Pg 117
clc;clear;close;

// Given Data
Sut=600;//MPa
Syt=400;//MPa
Se=200;//MPa
```

```
Mmin=200;// N.m
Mmax=500;// N.m
Tmin=60;// N.m
Tmax=180;// N.m
n=2;// factor of safety

Mm=(Mmax+Mmin)/2;// N.mm
Ma=(Mmax-Mmin)/2;// N.mm
Tm=(Tmax+Tmin)/2;// N.mm
Ta=(Tmax-Tmin)/2;// N.mm
// sigma_xm=32*Mm/%pi/d**3
sigma_xm_into_d_cube=(32*Mm*1000)/%pi;
// sigma_xa=32*Ma/%pi/d**3
sigma_xa_into_d_cube=(32*Ma*1000)/%pi;
//Txym=16*Tm/%pi/d**3
Txym_into_d_cube=16*Tm*1000/%pi;
//Txya=16*Ta/%pi/d**3
Txya_into_d_cube=16*Ta*1000/%pi;
// sigma_m=sqrt(sigma_xm**2+3*Txym**2)
sigma_m_dash=sqrt(sigma_xm_into_d_cube**2+3*
    Txym_into_d_cube**2);// taken sigma_m_dash =
    sigma_m*d**(-3) for calculation
// sigma_a=sqrt(sigma_xa**2+3*Txya**2)
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
//tan(theta) = sigma_a/sigma_m
theta = atan(sigma_a_dash/sigma_m_dash);// radian
//Sm/Sut+Sa/Se= 1 where Sa/Sm=0.4348
Sm= 1/(1/Sut+0.4348/Se);// MPa
Sa=0.4348 * Sm;// MPa
//sigma_a=Sa/n
d=(Sa/n/sigma_a_dash)**(1/3)*1000;// mm
printf('\n diameter of shaft , d = %.2f mm',d)
// Note - Ans in the textbook is wrong.
```

Scilab code Exa 4.12 `Diameter of the shaft`

```
// exa 4.12 Pg 119
clc;clear;close;

// Given Data
Sut=620;//MPa
Syt=380;//MPa
R=90/100;// Reliability
n=2.5;// factor of safety
Tmin=-200;// N.m
Tmax=400;// N.m

Se_dash=0.5*Sut;//MPa
// for ground shaft
ka=0.92;// surface finish factor
kb=0.85;// size factor (assuming t<50 mm)
kc=0.897;// reliability factor
kd=1;// temperature factor
ke=0.577;// load factor
Ses=ka*kb*kc*kd*ke*Se_dash;// MPa( Endurance limit)
Sys=ke*Syt;// MPa
Tm=(Tmax+Tmin)/2;// N.mm
Ta=(Tmax-Tmin)/2;// N.mm
theta=atan(Ta/Tm);//radian
Sas=Ses;// MPa
Sms=Sas/3;// MPa
//Tda=Sas/n=16*Ta/%pi/d**3
d=(16*Ta*1000/%pi/(Sas/n))**(1/3);// mm
printf('\n diameter of shaft, d = %.2f mm or %d mm',
   d, ceil(d))
```

Scilab code Exa 4.14 Minimum required ultimate strength

```
// exa 4.14 Pg 121
clc;clear;close;

// Given Data
sigma_max=300;// MPa
sigma_min=-150;// MPa
n=1.5;// factor of safety


sigma_m=(sigma_max+sigma_min)/2;// MPa
sigma_a=(sigma_max-sigma_min)/2;// MPa
// Goodman failure line - sigma_m/Sut+sigma_a/Se=1/n
Sut=(sigma_m+sigma_a/(0.5))*n ;// putted Se=0.5*Sut
printf('\n Minimum required ultimate strength, Sut =
    %.1f MPa',Sut)
```

Scilab code Exa 4.16 Size of piston rod

```
// exa 4.16 Pg 122
clc;clear;close;

// Given Data
Pmin=-15;// kN
Pmax=25;// kN
Se_dash=360;// MPa
Sy=400;// MPa
Ki=1.25;// impact factor
n=2.25;// factor of safety
ka=0.88;// surface finish factor
Kt=2.25;// stress concentration factor
Pm=(Pmax+Pmin)/2;// kN
Pa=(Pmax-Pmin)/2;// kN
q=0.8;// sensitivity factor
```

```

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

Scilab code Exa 4.18 Suitable diameter of rod

```
// exa 4.18 Pg 123
clc;clear;close;

// Given Data
Pmin=-300;// kN
Pmax=700;// kN
Se_dash=280;// MPa
Sy=350;// MPa
Kf=1.8;//fatigue strength factor
n=2;// factor of safety

Pm=(Pmax+Pmin)/2;// kN
Pa=(Pmax-Pmin)/2;// kN
// sigma_m=4*Pm/%pi/d**2
sigma_m_into_d_sq = 4*Pm*1000/%pi;
sigma_a_into_d_sq = 4*Pa*1000/%pi;
kf=1/Kf ;// fatigue strength reduction factor
```

```
kb=0.85;// size factor
ke=0.9;//load factor
ka=0.93;// surface finish factor
Se=ka*kb*ke*kf*Se_dash;// MPa
//Goodman failure equation - sigma_m/Sy+sigma_a/Se
    =1/n
d=sqrt((sigma_m_into_d_sq/Sy+sigma_a_into_d_sq/Se)
    *2.25)
printf('\n Suitable diameter of rod, d = %.f mm',d)
// Note - Ans in the textbook is wrong.
```

Scilab code Exa 4.19 Thickness of plate

```
// exa 4.19 Pg 124
clc;clear;close;

// Given Data
w=110;// mm
Pmin=98.1;// kN
Pmax=250;// kN
Se=225;// N/mm.sq
Sy=300;// N/mm.sq
n=1.5;// factor of safety

Pm=(Pmax+Pmin)/2;// kN
Pa=(Pmax-Pmin)/2;// kN
// sigma_m=Pm/w/t
sigma_m_into_t = Pm/w;
sigma_a_into_t = Pa/w;
//Soderburg failure equation - sigma_m/Sy+sigma_a/Se
    =1/n
d=(sigma_m_into_t/Sy+sigma_a_into_t/Se)*n*1000;// mm
printf('\n thickness of plate, t = %.1f mm',d)
```

Scilab code Exa 4.20 `Shaft size for infinite life`

```
// exa 4.20 Pg 124
clc;clear;close;

// Given Data
Mmin=200;// kN.mm
Mmax=600;// kN.mm
Tmin=60;// kN
Tmax=180;// kN
Su=550;// MPa
Sy=400;// MPa
Se=0.5*Su;// MPa
n=1.5;// factor of safety
Ktb=1.5;// stress concentration factor in blending
Kts=1.2;// stress concentration factor in torsion

Mm=(Mmax+Mmin)/2;// kN.mm
Ma=(Mmax-Mmin)/2;// kN.mm

//sigma_xm=32*Mm/%pi/d**3
sigma_xm_into_d_cube=32*Mm/%pi;
//sigma_xa=32*Ma/%pi/d**3
sigma_xa_into_d_cube=32*Ma/%pi;
Tm=(Tmax+Tmin)/2;// kN.mm
Ta=(Tmax-Tmin)/2;// kN.mm
Txym_into_d_cube=16*Tm/%pi;
Txya_into_d_cube=16*Ta/%pi;
// using distortion energy theory
// sigma_m=sqrt(sigma_xm**2+3*Txym**2)
sigma_m_into_d_cube=sqrt(sigma_xm_into_d_cube**2+3*Txym_into_d_cube**2);
// sigma_a=sqrt((Ktb*sigma_xa)**2+3*(Kts*Txym)**2)
sigma_a_into_d_cube=sqrt((Ktb*sigma_xa_into_d_cube)
```

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

Scilab code Exa 4.21 Thickness of bar

```
// exa 4.21 Pg 126
clc;clear;close;

// Given Data
// Hole -
d=25;//mm
w=150;//mm
Kt=2.56;// stress concentration factor
P=50;// kN
sigma_max=100;// N/mm.sq
t=Kt*P*1000/(w-d)/sigma_max;// mm
printf('Calculating for hole - \n thickness is : %.2f mm',t)

// Notch -
d=30;//mm
w=120;//mm
w=150;//mm
Kt=2.3;// stress concentration factor
P=50;// kN
sigma_max=100;// N/mm.sq
t=Kt*P*1000/(w-d)/sigma_max;// mm
printf('\n Calculating for notch - \n thickness is : %.2f mm',t)
disp('Suggestion , Adopt t = 11 mm')
```

Chapter 5

Riveted Joints

Scilab code Exa 5.1 Design the longitudinal and circumferential joints

```
// exa 5.1 Pg 142
clc;clear;close;

// Given Data
ps=2.5;// MPa
D=1.5;//m
sigma_t=80;// MPa
tau=60;// MPa
sigma_c=120;// MPa
n=5;// no. of rivets

printf('DESIGNING LONGITUDINAL JOINT - \n')
printf('\n Plate Thickness')
eta_l=80;// % (efficiency)
t = ps*D*1000/2/sigma_t/(eta_l/100)+1;// mm
printf(', t = %.2f mm',t)
t=32;//mm (adopted for design)
printf('\n use t = %d mm',t)
printf('\n Diameter of rivet hole, do = ')
d0=6*sqrt(t);//mm (for t>8 mm)
printf('%.2f mm',d0)
```

```
d0=34.5;// suggested for design
printf('\n Use do = %.f mm',d0)
printf('\n Diameter of rivet , d = ')
d=d0-1.5;//mm
printf('%.2f mm',d)
printf('\n Pitch of rivets , p = ')
Ps=(4*1.875+1)*%pi/4*d0**2*tau;// N
// Putting Pt=Ps where Pt=(p-d0)*t*sigma_t;// N
Pt=Ps;//N
p=Pt/(t*sigma_t)+d0;// N
printf('%.1f mm',p)
C=6;// for 5 no. of rivets
pmax=C*t+40;// mm (as per IBR)
printf('\n as per IBR-\n pitch , pmax = %.f mm',pmax)
p=220;// mm (adopted for design)
printf('\n Use p = %.f mm',p)
pi=p/2;// mm
printf('\n pitch of rivets in inner row, pi = %.f mm
    ',pi)

//Distance between rows of rivets
dis1=0.2*p+1.115*d0;// mm
printf('\n distance between outer and adjacent row =
    %.1f mm',dis1)
dis1=85;//mm (adopted for design)
printf('\n take & use this distance = %.f mm', dis1)
dis2=0.165*p+0.67*d0;// mm
printf('\n distance between inner row for zig-zag
    riveting = %.1f mm', dis2)
dis2=60;//mm (adopted for design)
printf('\n take & use this distance = %.f mm', dis2)
printf('\n Thickness of wide butt strap , t= ')
t1=0.75*t;// mm (wide butt strap)
printf(' %.f mm',t1)
t2=0.625*t;// mm (narrow butt strap)
printf('\n Thickness of narrow butt strap , t= %.f mm
    ',t2)
//margin
```

```
m=ceil(1.5*d0);// mm
printf('\n margin, m = %.f mm',m)
// Efficiency of joint
Pt=(p-d0)*t*sigma_t;// N
Ps=Ps;// N (shearing resistance of rivets)
Pc=n*d0*t*sigma_c;// N (crushing resistance of rivets)
sigma_com = (p-2*d0)*t*sigma_t+%pi/4*d0**2*tau;// N
printf('\n strength of the joint = %d N',sigma_com)
P=p*t*sigma_t;//N (strength of solid plate)
printf('\n strength of solid plate = %d N',P)
eta_l=sigma_com/P*100;// % (efficiency)
printf('\n Efficiency of joint, eta_l = %.1f %%',eta_l)

printf('\n\n DESIGNING CIRCUMFERENTIAL JOINT- \n')
t=32;// mm
d0=34.5;//mm
d=33;//mm
printf('\n Plate Thickness')
printf(', t = %.2f mm',t)
printf('\n Diameter of rivet hole, do = ')
printf('%.2f mm',d0)
printf('\n Diameter of rivet, d = ')
printf('%.2f mm',d)
n=(D*1000/d0)**2*(ps/tau);// no. of rivets
printf('\n no. of rivets = %.1f',n)
n=80;// adopted for design
printf('\n take n = %d',n)
// Pitch of rivets
n1=n/2;// no. of rivets per row
pc=%pi*(D*1000+t)/n1;// mm (pitch of rivets)
printf('\n pitch of rivets, pc = %.2f mm\n use pc = %.f mm',pc,pc)
eta_c=(pc-d0)/pc*100;// % (efficiency of joint)
printf('\n Efficiency of joint, eta_c = %.2f %%',eta_c)
dis=0.33*pc+0.67*d0;// mm (distance between rows of
```

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

Scilab code Exa 5.2 Design an economical joint

```
// exa 5.2 Pg 147
clc;clear;close;

// Given Data
w=400;//mm
t=20;//mm
sigma_t=90;// MPa
tau=60;// MPa
sigma_c=140;// MPa

printf('\n Diameter of rivet, do = ')
d0=6*sqrt(t);//mm (for t>8 mm)
printf('%.2f mm',d0)
d0=29;//mm (standard)
printf('\n Standard diameter of rivet hole, do = %.f mm & corresponding diameter of rivet = 27 mm',d0)
Pt=(w-d0)*t*sigma_t;//N max. tearing strength of plate
Ps=1.75*%pi/4*d0**2*tau;// N (shearing strength of one rivet)
n1=Pt/Ps;// no. of rivets
n=ceil(n1);
printf('\n no. of rivets, n = %.3f. Use n = %.f ',n1,n)
t1=0.75*t;// mm
t2=t1;// mm
```

```
printf('\n thickness of inner butt strap, t1 = %.f
   mm', t1)
printf('\n thickness of outer butt strap, t2 = %.f
   mm', t2)
// section 1-1
P1=(w-d0)*t*sigma_t;//N
// section 2-2
P2=(w-2*d0)*t*sigma_t+1.75*%pi/4*d0**2*tau;//N
// section 3-3
P3=(w-3*d0)*t*sigma_t+1.75*3*%pi/4*d0**2*tau;//N
// section 4-4
P4=(w-4*d0)*t*sigma_t+1.75*6*%pi/4*d0**2*tau;//N
Ps=10*Ps;// N (shearing stress of all rivets)
Pc=10*d0*t*sigma_c;// N (shearing stress of all
    rivets)
Pj=P1;// N (strength f joint)
P = w*t*sigma_t;// N (strength of solid plate)
eta=P1/P*100; // % (efficiency of joint)
printf('\n efficiency of joint = %.2f %%', eta)
p1=3*d0+5;// mm (pitch of rivets)
p=100;//mm (adopt for design)
printf('\n pitch of rivets = %.f mm. Use %.f mm',p1,
   p)
m1=1.5*d0;// mm (margin)
m=50;//mm
w=3*p+2*m;// mm
printf('\n margin,\n m = %.1f mm. Use %.f mm', m1,m)
printf('\n w = %.f mm',w)
dis=2.5*d0;// mm
printf('\n distance between rows = %.1f mm. Use 75
   mm',dis)
```

Scilab code Exa 5.3 Diameter of rivets

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

```
clc;clear;close;

// Given Data
n=6;// no. of rivets
P=54;// kN
e=200;//mm
a=50;//mm (from fig.5.13(a))
b=100;//mm (from fig.5.13(a))
tau=120;// MPa

Pd=P/n*1000;// N (direct shear load in rivet)
C=P*e;// kN.mm (Couple)
//l1=l3=l4=l6
l1=sqrt(a**2+b**2);// mm
l3=l1;l4=l1;l6=l1//mm
l2=a;l5=a;//mm
// F1/l1*(4*l1**2+2*l2**2)=C
F1=C*1000/(4*l1**2+2*l2**2)*l1;// N
theta1=acos(a/l1);// radian
R1=sqrt(Pd**2+F1**2+2*Pd*F1*cos(theta1));// N (resultant force in rivet 1)
//R1=%pi/4*d0**2*tau
d0=sqrt(R1/(%pi/4*tau));// mm
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

```
// exa 5.4 Pg 151
clc;clear;close;

// Given Data
D=0.75;//m
ps=1.55;// N/mm.sq
eta_l=0.75;// efficiency
```

```
sigma_t=90;// MPa
sigma_c=140;// MPa
tau=56;// MPa
n=2;// no. of rivets

printf('DESIGNING LONGITUDINAL JOINT - \n')
printf('\n Plate Thickness')
t = ps*D*1000/2/sigma_t/eta_l+1;// mm
printf(', t = %.2f mm',t)
t=ceil(t);//mm (adopted for design)
printf('\n use t = %d mm',t)

printf('\n Diameter of rivet hole, do = ')
d0=6*sqrt(t);//mm (for t>8 mm)
printf('%.2f mm',d0)
d0=19.5;// suggested for design
printf('\n Use do = %.1f mm',d0)
printf('\n Diameter of rivet, d = ')
d=d0-1.5;//mm
printf('%.2f mm',d)

printf('\n Pitch of rivets, p = ')
Ps=(2*1.875)*%pi/4*d0**2*tau;// N
// Putting Pt=Ps where Pt=(p-d0)*t*sigma_t;// N
Pt=Ps;//N
p=Pt/(t*sigma_t)+d0;// N
printf('%.2f mm',p)
C=3.5;// for 2 no. of rivets
pmax=C*t+40;// mm (as per IBR)
printf('\n as per IBR-\n pitch, pmax = %.f mm',pmax)
p=75;// mm (adopted for design)
printf('\n Use p = %.f mm',p)

//Distance between rows of rivets
dis=0.33*p+0.67*d0;// mm
printf('\n distance between rows of rivets = %.1f mm',dis)
dis=40;//mm (adopted for design)
```

```
printf('\n take & use this distance = %.f mm', dis)

printf('\n Thickness of butt strap , t= ')
t1=0.625*t;// mm
printf(' %.2f mm',t1)
t1=7;// mm (adopted for design)
printf('\n Use thickness = %.f mm',t1)

//margin
m=ceil(1.5*d0);// mm
printf('\n margin , m = %.f mm',m)

// Efficiency of joint
Pt=(p-d0)*t*sigma_t;// N
Ps=Ps;// N (shearing resistance of rivets)
Pc=n*d0*t*sigma_c;// N (crushing resistance of rivets)
sigma_com = (p-2*d0)*t*sigma_t+%pi/4*d0**2*tau;// N
printf('\n strength of the joint = %d N',Pt)
P=p*t*sigma_t;//N (strength of solid plate)
printf('\n strength of solid plate = %d N',P)
eta_l=Pt/P*100;// % (efficiency)
printf('\n Efficiency of joint , eta_l = %.2f %% = 75 %% as given',eta_l)
```

Scilab code Exa 5.6 Suitable diameter of rivets

```
// exa 5.6 Pg 153
clc;clear;close;

// Given Data
n=5;// no. of rivets
P=45;// kN
alfa=30;// degree
tau=120;// MPa
```

```
Pd=P/n*1000;// N (direct shear load in rivet)
// C.G. of rivet group
// values below are collected direct from figure
x_bar=(3*200)/5;// mm
y_bar=(1*50+1*150+1*100+1*200)/5;// mm
ex=300+x_bar+y_bar;//mm
ey=100;//mm
l1=sqrt(x_bar**2+(y_bar/2)**2);// mm
l2=l1;//mm
l3=sqrt(100**2+80**2);// mm
l4=80;//mm
l5=l3;//mm

//2*F1*l1+2*F3*l3+F4*l4=P*cos(alfa)*ex+P*sin(alfa)*ey
F1=(P*1000*cosd(alfa)*ex+P*1000*sind(alfa)*ey)/(2*l1**2+2*l3**2+l4**2)*l1;//N
// rivet 1 is nearest
Beta = atand(x_bar/(y_bar/2));// degree
theta1=Beta-(90-alfa);// degree
R1=sqrt(Pd**2+F1**2+2*Pd*F1*cosd(theta1));// N (resultant force in rivet 1)
//R1=%pi/4*d0**2*tau
d0=sqrt(R1/(%pi/4*tau));// mm
printf('\n diameter of rivets = %.2f mm. Use d0 = 21.5 mm & d=20 mm for design.',d0)
// Note - Ans in the textbook is wrong.
```

Scilab code Exa 5.7 Diameter of rivets

```
// exa 5.7 Pg 155
clc;clear;close;

```

```
// Given Data
t=6;//mm
sigma_t=220;// MPa
tau=100;// MPa
sigma_c=150;// MPa
n=2;// no. of rivets / pitch length
//Ps=n*%pi/4**d0**2*tau;// shearing strength of rivets
//Pc=2*d0*t*sigma_c;// Crushing strength of rivets
d0=2*t*sigma_c/(n*%pi/4*tau);// mm (equating Ps=Pc)
printf('Diameter of rivets , d0 = %.2f mm. Take d0 =13.5 mm & d=12 mm',d0)
d0=13.5;//mm
d=12;//mm
//Pt=(p-d0)*t*sigma_t;// tearing strength
// equating Pt=Ps
//p= n*%pi/4**d0**2*tau/(t*sigma_t)+d0;//mm
p= n*%pi/4*d0**2*tau/(t*sigma_t)+d0
printf('\n Distance between rows of rivet = %.1f mm = %.f mm',p,p)
p=floor(p);//mm
pb=0.6*p;//mm (back pitch)
printf('\n back pitch = %.f mm',pb)
Pt=(p-d0)*t*sigma_t;// N (tearing strength)
printf('\n tearing strength = %.f N',Pt)
Ps=n*%pi/4*d0**2*tau;// N ( shearing strength)
printf('\n shearing strength = %.f N',Ps)
Pc=2*d0*t*sigma_c;//N (Crushing strength of rivets)
printf('\n crushing strength = %.f N',Pc)
joint_strength = Pc;// N
printf('\n joint strength = %.f N',joint_strength)
P=p*t*sigma_t;//N (strength of solid plate)
printf('\n strength of solid plate = %.f N',P)
eta = joint_strength/P*100;// % (efficiency)
printf('\n efficiency of joint = %.1f %%', eta)
```

Scilab code Exa 5.8 Diameter of rivets

```
// exa 5.8 Pg 156
clc;clear;close;

// Given Data
P=20;// kN
e=80;//mm
tau=150;// MPa


Pd=P/4;// kN
C=P*e;// kN.mm (Couple)
// As C.G. lies at 45mm from top rivet
l1=45;l4=45;//mm
l2=15;l3=15;//mm
//(F1/l1)*(2*l1*l4+2*l2*l3) = C
F1= C*1000/(2*l1*l4+2*l2*l3)*l1;//N
R1=sqrt(Pd**2+F1**2);// N
//R1=%pi/4*d0**2*tau
d0=sqrt(R1/(%pi/4*tau));//mm
printf('Diameter of rivets - \n d0 = %.3f mm',d0)
printf('\n Use d0 = 13.5 mm & d = 12 mm')
```

Chapter 6

Shafts

Scilab code Exa 6.1 Shaft Diameter using AMSE code

```
// exa 6.1 Pg 168
clc;clear;close;

// Given Data
Sut=650;// MPa
Syt=380;// MPa
F1BYF2 = 2.5;// ratio of tensions
Fmax=2.5;// kN
da=200;// mm
db=400;// mm
L=1*1000;//mm
Km=1.5;// fatigue factor
Kt=1;// shock factor


tau_d1=0.30*Syt;// MPa
tau_d2=0.18*Sut;// MPa
tau_d=min(tau_d1, tau_d2);// MPa (taking minimum
    value)
tau_d=0.75*tau_d;//MPa (Accounting keyway effect)

```

```
// Pulley A
F1=2500;// N
F2=1000;// N
T=(F1-F2)*da/2;// N.mm
Fa=F1+F2;// N (resultant pull Downwards)

// Pulley B
// F3 & F4 are tension in belt (assumed)
//T=(F3-F4)*db/2
SUB_F3F4 = 2*T/db;// N (where SUB_F3F4 = F3-F4) --eqn(1)
F3BYF4=F1BYF2;// ratio of tensions --eqn(2)
F4 = SUB_F3F4/(F3BYF4-1);// N (using above 2 equations)
F3=F3BYF4*F4;// N
Fb=F3+F4;// N (resultant pull right side( -->))

// BENDING MOMENTS -
Mav=Fa*L/4;// N.mm (vertical force)
Mc=Fb*da;// N.mm
Mah=Mc/2;// N.mm (vertical force)
M = sqrt(Mav**2+Mah**2);// N.mm (resultant bending moment at A)
d=((16/%pi/tau_d)*sqrt((Km*M)**2+(Kt*T)**2))**(1/3);// mm

printf('shaft diameter = %.2f mm. Use diameter = 45 mm.',d)
```

Scilab code Exa 6.2 Diameter of the shaft

```
// exa 6.2 Pg 170
clc;clear;close;

// Given Data
```

```
Tmax=400;// N.m
Tmin=140;// N.m
Mmax=500;// N.m
Mmin=250;// N.m
Sut=540;// MPa
Syt=400;// MPa
n=2;// factor of safety
Kf=1.25;// given

Se_dash=0.4*Sut;// Mpa
Se=Se_dash/Kf;//MPa
Sys=0.577*Syt;// MPa
Ses=0.577*Se;// MPa
Mm=(Mmax+Mmin)/2;// N.m
Ma=(Mmax-Mmin)/2;// N.m
Tm=(Tmax+Tmin)/2;// N.m
Ta=(Tmax-Tmin)/2;// N.m
// Max. Distortion energy theory - Syt/n = 32/%pi/d
    **3*sqrt((Mm+Ma*(Syt/Se)**2)+0.75*(Tm+Ta*(Sys/Ses
    ))**2)
d = (32/%pi*sqrt((Mm+Ma*(Syt/Se))**2+0.75*(Tm+Ta*(
    Sys/Ses))**2)*1000/(Syt/n))**(1/3) ; // mm
printf('shaft diameter = %.2f mm. Use %.f mm.',d,d)
```

Scilab code Exa 6.3 Diameter of the shaft

```
// exa 6.3 Pg 171
clc;clear;close;

// Given Data
P=5;// kW
N=1000;// rpm
Syt=300;// N/mm.sq.
n=2;// factor of safety
v=0.25;// Poisson's ratio
```

```

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

Scilab code Exa 6.4 `Shaft Diameter using AMSE code`

```
// exa 6.4 Pg 171
clc;clear;close;

// Given Data
Sut=700;// MPa
Syt=460;// Mpa
F1BYF2=3;// ratio of tensions
dg=300;// mm
dp=400;// mm
P=25;// kW
N=600;// rpm
alfa=20;// degree
Km=1.5;// fatigue factor
Kt=1.5;// shock factor

tau_d1=0.30*Syt;// MPa
tau_d2=0.18*Sut;// MPa
tau_d=min(tau_d1, tau_d2);// MPa (taking minimum
    value)
tau_d=0.75*tau_d;//MPa (Accounting keyway effect)

```

```
// Pulley D
// P= 2*%pi*N*T/60
T=P/(2*%pi*N/(60*1000));// N.m
// (F1-F2)*dp/2=T
SUB_F1F2 = T*2/dp;// N (where SUB_F1F2 = F1-F2)
F2 = SUB_F1F2/(F1BYF2-1) ;// N (putting value of ratio)
F1=F1BYF2*F2;// N
F=F1+F2;// N
// Gear B
Ft=T*2/dg;// N
Fr=Ft*tand(alfa);// N

// Bearing Reactions

//Vertical forces
//RA*2*dg+Fr*dg=F*dg;
RA=(F*dg-Fr*dg)/(2*dg);// N (downwards)
RC=RA+Fr+F;// N (upwards)
MA=0;MB_v=-RA*dg;// N.mm
MC=-F*dg;// N.mm
//Horizontal forces
MB_h=Ft*2*dg/4;// N.mm
//Resultant B.M at B
MB=sqrt(MB_v**2+MB_h**2);// N.mm
Mmax=MC;//N.mm
T=T*1000;// N.mm
// d**3=16/%pi/tau_d*sqrt((Km*M)**2+(Kt*T)**2)
d=(16/%pi/tau_d*sqrt((Km*Mmax*1000)**2+(Kt*T)**2))**(1/3)
printf('shaft diameter(using ASME Code) = %.1f mm. Use diameter = %.f mm.',d,d)
```

Scilab code Exa 6.5 Diameter of the shaft

```
// exa 6.5 Pg 174
clc;clear;close;

// Given Data
L=1000;// mm
alfa=20;// degree
dg=500;// mm
L1=250;// mm
L2=300;// mm
dp=600;// mm
Wp=2000;// N
F1=2.5*1000;// N
F1BYF2=3;// ratio of tensions
tau_d=42;// MPa

F2=F1/F1BYF2;// N
T=(F1-F2)*dp/2;// N.mm
Ftg=2*T/dg;// N
Frg=Ftg*tand(alfa);// N
F=F1+F2;// N

// Vertical Loads
RAV=(Ftg*(L1+dg)+Wp*L2)/L;// N
RBV=Ftg+Wp-RAV;// N
MCV=RAV*L1;//N.mm
MDV=RBV*L2;// N.mm
// Horizontal Loads
RAH=(Frg*(L1+dg)+F*L2)/L;//N
RBH=Frg+F-RAH;// N
MCH=RAH*L1;// N.mm
MDH=RBH*L2;// N.mm
MD=sqrt(MDV**2+MDH**2);// N.mm
Mmax=MD;//N.mm
Te=MCV+MDV;// N.mm
// d**3 = 16*Te/%pi/tau_d
d = (16*Te/%pi/tau_d)**(1/3);//mm
printf('shaft diameter = %.1f mm.',d)
```

Scilab code Exa 6.6 Diameter of the shaft

```
// exa 6.6 Pg 176
clc;clear;close;

// Given Data
Tmax=400;// N.mm
Tmin=200;// N.mm
Mmax=500;// N.mm
Mmin=250;// N.mm
Sut=540;// MPa
Syt=420;// MPa
n=2;// factor of safety

Se=0.35*Sut;// MPa

Mm=(Mmax+Mmin)/2;// N.m
Ma=(Mmax-Mmin)/2;// N.m
Tm=(Tmax+Tmin)/2;// N.m
Ta=(Tmax-Tmin)/2;// N.m
Sys=0.5*Syt// MPa
Ses=0.5*Se;// MPa
// Max. Distortion energy theory - Syt/n = 32/%pi/d**3*sqrt((Mm+Ma*(Syt/Se)**2)+0.75*(Tm+Ta*(Sys/Ses))**2)
d = (32/%pi*sqrt((Mm+Ma*(Syt/Se))**2+0.75*(Tm+Ta*(Sys/Ses))**2)*1000/(Syt/n))**(1/3) ; // mm
printf('shaft diameter = %.f mm.',d)
```

Scilab code Exa 6.7 Diameter of the bar

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

```
clc;clear;close;

// Given Data
Wmax=40;// kN
Wmin=20;// kN
L=500;// mm
Se_dash=350;// MPa
Sut=650;// MPa
Syt=500;// MPa
n=1.5;// factor of safety
ka=0.9; // surface finish factor
kb=0.85;// size factor
ke=1;// load factor
Kf=1;// fatigue strength factor

Wm=1/2*(Wmax+Wmin);// N
Wa=1/2*(Wmax-Wmin);// N
Se=ka*kb*ke*Se_dash;//MPa
Mm=Wm*L/1000/4;// kN.m
Ma=Wa*L/1000/4;// kN.m
//sigma_m=32*Mm/%pi/d**3; & sigma_a=32*Ma/%pi/d**3
//soderburg failure criteria - 1/n=sigma_m/Syt+Kf*sigma_a/Se
//d=((32/%pi*n/1000)*(Mm/Syt+Kf*Ma/Se))*(1/3)
d=((32/%pi/1000)*(Mm/Syt+Kf*Ma/Se)*n)**(1/3)*1000;// mm
printf('shaft diameter = %.f mm.',d)
```

Scilab code Exa 6.8 Required shaft diameter

```
// exa 6.8 Pg 178
clc;clear;close;

// Given Data
Tmax=300;// N.mm
```

```
Tmin=-100;// N.mm
Mmax=400;// N.mm
Mmin=-200;// N.mm
n=1.5;// factor of safety
Sut=500;// MPa
Syt=420;// MPa
sigma_d=280;// MPa
ka=0.62; // surface finish factor
kb=0.85;// size factor
keb=1;// load factor for bending
kes=0.58;// load factor for torsion
Kfb=1;// fatigue strength factor for bending
Kfs=1;// fatigue strength factor for torsion

Se_dash=0.5*Sut;// MPa
Se=ka*kb*keb*Se_dash;// MPa
Ses_dash=0.5*Se_dash;// MPa
Ses=ka*kb*kes*Ses_dash;// MPa
Sys=0.5*Syt;// MPa
Mm=(Mmax+Mmin)/2;// N.m
Ma=(Mmax-Mmin)/2;// N.m
Tm=(Tmax+Tmin)/2;// N.m
Ta=(Tmax-Tmin)/2;// N.m

// tau_d/n = (16/%pi/d**3)*sqrt((Mm+Ma*(Syt/Se)**2)+(Tm+Ta*(Sys/Ses))**2)
tau_d=sigma_d/2;// MPa
d = ((16/%pi)*sqrt((Mm+Ma*(Syt/Se)**2)+(Tm+Ta*(Sys/Ses))**2)/(tau_d*10**6/n))**(1/3)*1000;// mm
printf('shaft diameter = %.2f mm.',d)
// 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

```
// exa 7.1 Pg 195
clc;clear;close;

// Given Data
P=20;// kW
N=240;// rpm
tau_s=45;// MPa
tau_b=30;// MPa
sigma_b=60;// MPa
sigma_cs=2*tau_s;// MPa
tau_ci=15;// MPa
//Tmax=1.25*Tm
mu=0.15;// coefficient of friction

//SHAFT DIAMETER
// P= 2*%pi*N*Tm/60/1000
Tm=P/(2*%pi*N/60/1000);// N.m
Tmax=1.25*Tm;// N.m
// %pi*d**3*tau_s/16= Tmax
d=(Tmax/(%pi*tau_s/16)*1000)**(1/3);// mm
printf('shaft diameter = %.2f mm. Use d = 50 mm.',d)
```

```
d=50;// mm

// HUB DIAMETER
// Tmax=%pi/16*((d1**4-d**4)/d1)*tau_h
tau_h=tau_ci;// MPa
//d1*(Tmax/(%pi/16)/tau_h)-d1**4=d**4 -- eqn(1)
Tmax=Tmax*1000;// N.mm
p=[1 0 0 -Tmax/(%pi*tau_h/16) -d**4] ;// polynomial coefficients from eqn(1)
d1=roots(p);// roots of poly
d1=d1(1);// mm (taking +ve value)
d1=100;// mm (empirically adopted)
t1=(d1-d)/2;// mm (thickness of hub)
printf('\n thickness of hub = %.f mm',t1)
d4=d+t1;// mm (diameter of recess in flanges)
printf('\n diameter of recess in flanges = %.f mm',d4)
d3=4*d;// mm (outside diameter of protecting flange)
printf('\n outside diameter of protecting flange = %.f mm',d3)

// Hub length
b=d/4;// mm (width of key)
l=1.5*d;// mm (length of key)
printf('\n width of key = %.1f mm. Use b = 15 mm',b)
b=15;// mm
printf('\n length of key = %.f mm.',l)
t=b;// mm (thickness for square key)
printf('\n thickness for square key = %.f mm',t)
printf('\n Hub length = %.f mm',l)

//Number of bolts
n=floor(4*d/150+3);// no. of bolts
printf('\n Number of bolts = %.f',n)

// Bolt diameter
r2=1.5*d;// mm
F=Tmax/r2/n;// N
```

```
//%pi/4*db**2*tau_b=F
db=sqrt(F/(%pi/4*tau_b));// mm
printf('\n Bolt diameter = %.2f mm. Use db=12 mm',db)
bolt_dia=db;//mm

// Bolt diameter based on Tensile load
r3=d3/2;// mm
r4=d4/2;// mm
rf=2/3*((r3**3-r4**3)/(r3**2-r4**2));// mm
//Tmax=n*mu*Pi*rf;// N
Pi=Tmax/(n*mu*rf);// N
// Pi=%pi/4*db**2*sigma_t
sigma_t=sigma_b;// MPa
db=sqrt(Pi/(%pi/4*sigma_t));// mm
printf('\n Bolt diameter (based on Tensile load) = %.1f mm. Use db=15 mm',db)
db=15;// mm (adopted)

// Flange thickness
t2=0.5*t1+6;// mm (empirically)
printf('\n Flange thickness = %.1f mm. Use t=20 mm',t2)
t2=20;// mm (adopted)
//F=n*db*t2*sigma_c
sigma_ci=F/n/db/t2;// MPa
//2*%pi*d1**2*tau*t2/4=Tmax
tau=Tmax/(2*%pi*d1**2*t2/4);// MPa
printf('\n permissible bearing stress in flange = %.2f MPa < 30 MPa',sigma_ci)
printf('\n shearing of the flange at the junction with hub = %.2f MPa < 15 MPa.',tau)
printf(' Values are acceptable.')

// Check for crushing of bolt
//n*db*t2*sigma_cb*d2/2=Tmax
d2=d1+d;// mm
db=bolt_dia;//mm
```

```
sigma_cb=Tmax/(n*db*t2*d2/2);// MPa
printf('\n permissible crushing strength of bolts =
   %.1f MPa < 60 MPa.',sigma_cb)
printf(' Hence design is safe.')

// Thickness of protecting flange
t3=0.5*t2;// mm
printf('\n Thickness of protecting flange = %.f mm',
    t3)
// Hub overlap
ho=3;// mm (min)
printf('\n Hub overlap = %.f mm (min)',ho)
//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

```
// exa 7.2 Pg 200
clc;clear;close;

// Given Data
P=30;// kW
N=750;// rpm
//Tmax=1.2*Tm;// MPa
tau_s=35;// MPa
tau_b=35;// MPa
tau_k=35;// MPa
sigma_cs=70;// MPa
sigma_ck=70;// MPa
sigma_cb=70;// MPa
tau_ci=15;// MPa
pb=0.8;// MPa

//sigma_cs=2*tau_s;// MPa
```

```

//Tmax=1.5*Tm
mu=0.15;// coefficient of friction

//SHAFT DIAMETER
// P= 2*%pi*N*Tm/60/1000
Tm=P/(2*%pi*N/60/1000);// N.m
Tmax=1.2*Tm;// N.m
// %pi*d**3*tau_s/16= Tmax
d=(Tmax/(%pi*tau_s/16)*1000)**(1/3);// mm
printf('shaft diameter = %.2f mm. Use d = 42 mm.',d)
d=42;// mm

// HUB DIAMETER
// Tmax=%pi/16*((d1**4-d**4)/d1)*tau_h
tau_h=tau_ci;// MPa
//d1*(Tmax/(%pi/16)/tau_h)-d1**4=d**4 -- eqn(1)
Tmax=Tmax*1000;// N.mm
p=[1 0 0 -Tmax/(%pi*tau_h/16) -d**4] ;// polynomial coefficients from eqn(1)
d1=roots(p);// roots of poly
d1=d1(1);// mm (taking +ve value)
d1=2*d;// mm (empirically adopted)
t1=(d1-d)/2;// mm (thickness of hub)
printf('\n thickness of hub = %.f mm',t1)
//d4=d+t1;// mm (diameter of recess in flanges)
//printf('\n diameter of recess in flanges = %.f mm',d4)
d3=4*d;// mm (outside diameter of protecting flange)
printf('\n outside diameter of protecting flange = %.f mm. Use 170 mm',d3)
d3=170;// mm (adopted)

//Key size & Hub length
b=d/4;// mm (width of key)
l=1.5*d;// mm (length of key)
printf('\n width of key = %.1f mm. Use b = 12 mm',b)
b=12;// mm
```

```
printf('\n length of key = %.f mm.',l)
t=b;// mm (thickness for square key)
printf('\n thickness for square key = %.f mm',t)
printf('\n Hub length = %.f mm',l)

//Number of bolts
n=(0.04*d+3);// no. of bolts
printf('\n Number of bolts = %.2f. Use n=6',n)
n=6;// adopted

// Bolt diameter
db=0.5*d/sqrt(n);// mm
printf('\n Bolt diameter = %.2f mm. Use db=20 mm for
    design purpose',db)
db=20;//mm (adopted)
bolt_dia=db;//mm
dsb=24;// mm(shank diameter of bolt for design)

// Outer diameter of rubber bush
trb=2;// mm (thickness of rubber bush)
tbb=6;// mm (thickness of brass bush)
d3=dsb+2*trb+2*tbb;// mm
d2=d1+d3+2*tbb;// mm (pitch circle diameter of bolts
    )
printf('\n pitch circle diameter of bolts = %.f mm '
    ,d2)

// Check of shear in bolt
F=2*Tmax/n/d2;// N
//%pi/4*db*2*tau=F
tau=F/(%pi/4*db**2);//MPa
printf('\n Permissible shear stress in bolts = %.2f
    MPa < 35 MPa. Hence design is safe.', tau)

// Length of brush
pb=0.8;// MPa(bearing pressure of brush)
//F=l2*d3*pb;
l2=F/d3/pb;// mm
```

```
printf('\n length of bush = %.f mm',l2)

// Check for pin in bending
c=5;// mm (clearance between two flanges)
l3=(l2-c)/2+c;//mm
//M=%pi/32*db**3*sigma_b & M=F*l3
sigma_b = F*l3/(%pi/32*db**3);// MPa
printf('\n Bending stress in pin = %.1f MPa',sigma_b
   )

// Maximum shear stress in pin
tau_max=sqrt((sigma_b/2)**2+tau**2);//MPa
printf('\n Maximum shear stress in pin = %.2f MPa <
   35 MPa. Hence design is safe.',tau_max)

// Flange thickness
t2=0.5*t1+6;// mm (empirically)
printf('\n Flange thickness = %.1f mm. Use t=18 mm',
   t2)
t2=18;// mm (adopted)
tau=Tmax/(2*%pi*d1**2*t2/4);// MPa
printf('\n shearing of the flange at the junction
   with hub = %.2f MPa < 15 MPa.',tau)
printf(' Values are acceptable.')

//Note - Answer in llast part is calculated wrong in
    the textbook(error in calculation).
```

Scilab code Exa 7.3 Length of hub

```
// exa 7.3 Pg 204
clc;clear;close;

// Given Data
n=8;// no. of spline
```

```
d=52;// mm
D=60;// mm
pm=6;// MPa
mu=0.06;// coefficient of friction
N=320;// rpm
P=20;// kW

T=60*10**3*P/2/%pi/N;// N.m
l=8*T*10**3/pm/n/(D**2-d**2);// mm
printf('length of hub = %.f mm',l)
Rm=(D+d)/4;// mm
F=T*10**3/Rm;// N
Ff=mu*F;//N (Force of friction)
printf('\n Force required to shift the connection =
   %.1f N',Ff)
```

Scilab code Exa 7.4 Design the rectangular key

```
// exa 7.4 Pg 204
clc;clear;close;

// Given Data
d=75;// mm
tau=50;// MPa
sigma_c=75;// MPa
printf('for key to be equally strong in shear &
   crushing - \n')
b=d/4;// mm
printf(' b= %.2f mm. Use b=20 mm. ',b)
b=20;//mm
//2*b/t=sigma_c/tau for key to be equally strong in
   shear & crushing
t=2*b/(sigma_c/tau);// mm
printf('\n t=%.2f mm. Use t=27 mm',t)
l= %pi*d**2/8/b;// mm (for key to be equally strong
```

```
    in shear as shaft)
printf('for key to be equally strong in shear as
    shaft - \n')
printf(' l=%.1f mm. Use l=115 mm',l)
```

Scilab code Exa 7.6 Suitable dimensions for shaft

```
// exa 7.6 Pg 205
clc;clear;close;

// Given Data
P=135;// kW
N=120;// rpm
tau_s=55;// MPa
tau_b=45;// MPa
tau_ci=175;// MPa
sigma_ci=75;// MPa

//sigma_cs=2*tau_s;// MPa

//Tmax=1.5*Tm
mu=0.15;// coefficient of friction

//SHAFT DIAMETER
// P= 2*%pi*N*Tm/60/1000
Tm=P/(2*%pi*N/60/1000);// N.m
// %pi*d**3*tau_s/16= Tm
d=(Tm/(%pi*tau_s/16)*1000)**(1/3);// mm
d=ceil(d)
printf('shaft diameter = %.2f mm.',d)
Tmax=Tm;// N.m

// HUB DIAMETER
// Tmax=%pi/16*((d1**4-d**4)/d1)*tau_h
tau_h=tau_ci;// MPa
```

```
//d1*(Tmax/(%pi/16)/tau_h)-d1**4=d**4 -- eqn(1)
Tmax=Tmax*1000;// N.mm
p=[1 0 0 -Tmax/(%pi*tau_h/16) -d**4] ;// polynomial coefficients from eqn(1)
d1=roots(p);// roots of poly
d1=d1(1);// mm (taking +ve value)
d1=2*d;// mm (empirically adopted)
t1=(d1-d)/2;// mm (thickness of hub)
printf('\n thickness of hub = %.f mm',t1)
d4=d+t1;// mm (diameter of recess in flanges)
printf('\n diameter of recess in flanges = %.f mm',d4)
d3=4*d;// mm (outside diameter of protecting flange)
printf('\n outside diameter of protecting flange = %.f mm.',d3)

//Key size & Hub length
b=d/4;// mm (width of key)
l=1.5*d;// mm (length of key)
printf('\n width of key = %.1f mm.',b)
printf('\n length of key = %.f mm.',l)
t=b;// mm (thickness for square key)
printf('\n thickness for square key = %.f mm',t)
printf('\n Hub length = %.f mm',l)

//Number of bolts
n=ceil(4*d/150+3);// no. of bolts
printf('\n Number of bolts = %.2f.',n)

// Bolt diameter
r2=1.5*d;// mm
F=Tm*1000/r2/n;//N
//(%pi/4)*db**2*tau_b=F
db=sqrt(F/((%pi/4)*tau_b));// mm
printf('\n Bolt diameter = %.2f mm. Use db=20 mm for design purpose',db)
db=20;// mm (adopted for design)
bolt_dia=db;//mm
```

```

// Flange thickness
t2=0.5*t1+6;// mm (empirically)
printf('\n Flange thickness = %.1f mm. Use t=20 mm',t2)
//F=n*db*t2*sigma_c
sigma_ci=F/n/db/t2;// MPa
//2*%pi*d1**2*tau*t2/4=Tmax
tau=Tmax/(2*%pi*d1**2*t2/4);// MPa
printf('\n permissible bearing stress in flange = %.2f MPa < 75 MPa',sigma_ci)
printf('\n shearing of the flange at the junction with hub = %.2f MPa < 175 MPa.',tau)
printf(' Values are acceptable.')

// Check for crushing of bolt
//n*db*t2*sigma_cb*d2/2=Tmax
d2=d1+d;// mm
db=bolt_dia;//mm
sigma_cb=Tmax/(n*db*t2*d2/2);// MPa
printf('\n permissible crushing strength of bolts = %.2f MPa < 60 MPa.',sigma_cb)
printf(' Hence design is safe.')
// Thickness of protecting flange
t3=0.5*t2;// mm
printf('\n Thickness of protecting flange = %.f mm',t3)
// Hub overlap
ho=3;// mm (min)
printf('\n Hub overlap = %.f mm (min)',ho)
```

Scilab code Exa 7.7 Design the rectangular key

```
// exa 7.7 Pg 208
clc;clear;close;
```

```

// Given Data
d=50;// mm
tau=42;// MPa
sigma_c=72;// MPa
printf('for key to be equally strong in shear & crushing - \n')
b=d/4;// mm
printf(' b= %.2f mm. Use b=15 mm.',b)
b=15;//mm
//2*b/t=sigma_c/tau for key to be equally strong in shear & crushing
t=2*b/(sigma_c/tau);// mm
printf('\n t=%.2f mm. Use t=20 mm',t)
l= %pi*d**2/8/b;// mm (for key to be equally strong in shear as shaft)
printf('\n for key to be equally strong in shear as shaft - \n')
printf(' l=%.2f mm. Use l=70 mm',l)
```

Scilab code Exa 7.8 `Design the rectangular key`

```
// exa 7.8 Pg 208
clc;clear;close;

// Given Data
d=25;// mm
N=550;// rpm
P=12;// kW
sigma_yt=400;// N/mm.sq.
sigma_yc=400;// N/mm.sq.
n=2.5;// factor of safety

// P= 2*%pi*N*T/(60*10**3)
T=P/(2*%pi*N/(60*10**3));// N.m
```

```
tau=0.5*sigma_yt;// MPa
tau_d=tau/n;// N/mm.sq.
printf('design shear stress = %.f N/mm.sq.',tau_d)
sigma_d=sigma_yc/n;// N/mm.sq.
printf('\n design crushing strength = %.f N/mm.sq.',sigma_d)
b=d/4;//mm
printf('\n width of key = %.f mm. Use 7mm',b)
b=ceil(d/4);// mm
t=b;// mm
printf('\n thickness of key = %.f mm.',t)
l_s=2*T*10**3/(d*b*tau_d);// mm (length of key based on shear failure)
printf('\n length of key based on shear failure = %.2f mm or %.f mm',l_s, l_s)
l_c=4*T*10**3/(d*t*sigma_d);// mm (length of key based on crushing failure)
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

```
// exa 7.9 Pg 209
clc;clear;close;

// Given Data
d=36;// mm
P=15;// kW
N=720;// rpm
//Tmax=1.25*Tm
sigma_yt=245;// MPa (for C20 steel)
n=3;// factor of safety
sigma=82;// MPa (Design tensile stress)

tau=0.577*sigma;// MPa (shear stress)
```

```
sigma_u=200;// MPa (for FG 200 cast Iron)
n2=5;// factor of safety (for FG 200 cast Iron)
tau2=20;// MPa shear stress (for FG 200 cast Iron)

// Max. torque transmitted
//P=2*%pi*N*Tm/(60*10**3)
Tm=P/(2*%pi*N/(60*10**3))*1000;// N.mm
Tmax=1.25*Tm;// N.mm
printf('\n Maximum transmitted torque = %.f N.mm',Tmax)

// Hub diameter
tau_h=20;// MPa (permissible shear stress in hub)
//Tmax=(%pi/16)*(d1**4-d**4)/d1*tau_h    ...eqn(1)
d1=2*d;//mm (empirically)
tau_h=Tmax*1000/((%pi/16)*(d1**4-d**4)/d1);// MPa
t1=(d1-d)/2;// mm (thickness of hub)
printf('\n Hub diameter = %.f mm',d1)
printf('\n Thickness of hub = %.f mm',t1)
d4=d+t1;// mm
printf('\n Diameter of recess in flanges = %.f mm',d4)
d3=4*d;//mm
printf('\n Outside diameter of protecting flange = %.f mm',d3)

//Hub length
b=d/4;// mm (width of key)
l=1.5*d;// mm (length of key)
printf('\n width of key = %.1f mm.',b)
printf('\n length of key = %.f mm.',l)
t=b;// mm (thickness for square key)
printf('\n thickness for square key = %.f mm',t)
printf('\n Hub length = %.f mm',l)

//Number of bolts
n=ceil(4*d/150+3);// no. of bolts
printf('\n Number of bolts = %.2f.',n)
```

```
// Bolt diameter
r2=1.5*d;// mm
F=Tmax/r2/n;//N
//(%pi/4)*db**2*tau_b=F
db=sqrt(F/((%pi/4)*tau));// mm
printf('\n Bolt diameter = %.2f mm. Use db=6 mm for
    design purpose',db)
db=6;// mm (adopted for design)
bolt_dia=db;//mm

// Flange thickness
t2=0.5*t1+6;// mm (empirically)
printf('\n Flange thickness = %.1f mm. Use t=20 mm',
    t2)
//F=n*db*t2*sigma_c
sigma_ci=F/n/db/t2;// MPa
//2*%pi*d1**2*tau*t2/4=Tmax
tau=Tmax/(2*%pi*d1**2*t2/4);// MPa
printf('\n permissible bearing stress in flange = %
    .2f MPa < 40 MPa',sigma_ci)
printf('\n shearing of the flange at the junction
    with hub = %.2f MPa < 20 MPa.',tau)
printf(' Values are acceptable.')

// Check for crushing of bolt
//n*db*t2*sigma_cb*d2/2=Tmax
d2=d1+d;// mm
db=bolt_dia;//mm
sigma_cb=Tmax/(n*db*t2*d2/2);// MPa
printf('\n permissible crushing strength of bolts =
    %.2f MPa < 82 MPa.',sigma_cb)
printf(' Hence design is safe.')
// Thickness of protecting flange
t3=0.5*t2;// mm
printf('\n Thickness of protecting flange = %.f mm',
    t3)
// Hub overlap
```

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

Scilab code Exa 7.10 Diameter of bolts

```
// exa 7.10 Pg 212
clc;clear;close;

// Given Data
d=35;// mm
d2=125;// mm
n=6;// factor of safety
T=800;// N.m
N=350;// rpm
tau_s=63;// MPa
tau_b=56;// MPa
tau_CI=10;// MPa
tau_k=46;// MPa

// Diameter of bolts:
F=2*T*10**3/d2/n;// N
//%pi/4*db**2*tau_b=F
db=sqrt(F/(%pi/4*tau_b));// mm
printf('\n (i) Diameter of bolts = %.2f mm. Use 8 mm
    .',db)

// Flange thickness
d1=2*d;// mm
//T=%pi/2*d1**2*t2*tau_CI
t2=T*1000/(%pi/2*d1**2*tau_CI);// mm
printf('\n (ii) Flange thickness = %.1f mm. Use t2 =
    12 mm',t2)
t2=12;// mm

//Key dimensions
```

```
b=10;// mm (width of key)
t=7;// mm (from tables)
//T=l*b*tau_k*d/2
l=T*10**3/(b*tau_k*d/2);// mm
l=ceil(l);// mm
printf('\n (iii) Length of key = %.f mm\n\t\td=%.f
   mm\n\t\tb=%.f mm',l,d,b)

// Hub length
lh=l;// mm (length of hub)
printf('\n (iv) Hub length = %.f mm',l)
tau_c=T*10**3/(%pi/16*(d1**4-d**4)/d1);// N/mm.sq.
printf('\n shear stress in hub = %.2f N/mm.sq.',
   tau_c)
printf('It is nearly equal to %.f N/mm.sq.',tau_CI)
printf('\n hence design parameters are fine.')

// Power transmitted
P=2*%pi*N*T/60/10**3;// kW
printf('\n (v) Power transmitted = %.2f kW',P)
```

Chapter 8

Mechanical Springs

Scilab code Exa 8.1 Design a close coiled helical compression spring

```
// exa 8.1 Pg 227
clc;clear;close;

// Given Data
Fmin=250;// N
Fmax=300;// N
del=8;// mm
C=8;// spring index
tau_d=420;// MPa
G=84;// GPa

// 1. Wahl's correction factor
Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction
    factor
printf("\n Wahl''s correction factor = %.3f ",Kw)
// 2. Wire diameter
// tau_d=Kw*8*Fmax*C/%pi/d**2
d=sqrt(Kw*8*Fmax*C/%pi/tau_d);// mm
printf('\n Wire diameter = %.2f mm. Use 4.25 mm.',d)
d=4.25;// mm
// 3. Mean coil diameter
```

```
Dm=8*d;// mm
printf('\n Mean coil diameter = %.f mm.',Dm)
// 4. Stiffness of spring
k=(Fmax-Fmin)/del;// N/mm
// 5. no. of active turns
n = G*10**3*d/8/C**3/k ;// no. of active turns
printf('\n no. of active turns = %.f',n)
// 6. total no. of turns for squared and ground ends
nt=n+2;// total no. of turns for squared and ground ends
printf('\n total no. of turns for squared and ground ends = %.f',nt)
// 7. Free length of spring
//lf=l_s+del_max+clashallowance(=0.15*del_max)
del_max=del*Fmax/(Fmax-Fmin);//mm
l_s=nt*d;// mm
lf=l_s+del_max+0.15*del_max;// mm
printf('\n Free length of spring = %.1f mm Use 124 mm',lf)
lf=124;//mm
// 8. Pitch of coils
p=lf/(nt-1);//mm
printf('\n Pitch of coils = %.2f mm',p)
// 9. Check for buckling
printf('\n Check for buckling - ')
m=lf/Dm;// > 2.6 provided guide
printf('\n ratio lf/Dm = %.3f > 2.6. So, Providing guide is necessary.',m)
kl_1=0.22;// for hinged ends
kl_2=0.62;// for fixed ends
Fcr_1=k*kl_1*lf;//N (for hinged ends)
Fcr_2=k*kl_2*lf;//N (for fixed ends)
printf('\n Critical load for buckling - ')
printf('\n Fcr = %.1f N for hinged ends < Fmax',Fcr_1)
printf('\n Fcr = %.1f N for fixed ends > Fmax',Fcr_2)
printf('\n From above two calculatio , it can be seen
```

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

Scilab code Exa 8.2 Factor of safety

```
// exa 8.2 Pg 228
clc;clear;close;

// Given Data
Fmin=60;// N
Fmax=140;// N
d=3;// mm
Dm=18;// mm
Sut=1430;// MPa

C=Dm/d;// spring index
Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction factor
Ks=1+0.5/C;// Shear Stress factor
Fm=(Fmax+Fmin)/2;// N
Fa=(Fmax-Fmin)/2;// N
tau_m=Ks*(8*Fm*C)/(%pi*d**2);// MPa
tau_a=Kw*(8*Fa*C)/(%pi*d**2);// MPa
Ses_dash=0.22*Sut;// MPa
Sys=0.45*Sut;// MPa
//tau_m/Sys+tua_a/Ses_dash*(2-Ses_dash/Sys)=1/n
n=1/(tau_m/Sys+tau_a/Ses_dash*(2-Ses_dash/Sys));// factor of safety
printf('\n factor of safety = %.2f',n)
```

Scilab code Exa 8.3 Initial tortional shear stress

```
// exa 8.3 Pg 229
clc;clear;close;

// Given Data
Fi=40;// N
d=3;// mm
C=6;// spring index
n=15;// factor of safety
tau=650;// N/mm.sq.
G=84;// kN/mm.sq.

// Wahl's correction factor
Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction
    factor
printf("\n Wahl''s correction factor = %.4f ",Kw)

// Initial tortional shear stress
tau_i=Kw*(8*Fi*C)/(%pi*d**2);// MPa
printf('\n Initial tortional shear stress = %.2f MPa
    ',tau_i)
k=G*10**3*d/(8*C**3*n);// spring stiffness
printf('\n spring stiffness = %.2f N/mm',k)
// Spring load to cause yielding
//tau=Kw*(8*Fi*C)/(%pi*d**2)
F=tau/(Kw*(8*C)/(%pi*d**2))
printf('\n Spring load to cause yielding = %.1f N',F
    )
```

Scilab code Exa 8.4 Diameter of spring wire

```
// exa 8.4 Pg 230
clc;clear;close;

// Given Data
Fmin=500;// N
Fmax=1200;// N
C=6;// spring index
n=1.5;// factor of safety
Sys=760;// MPa
Ses_dash=350;// MPa
del=25;// mm
G=82;// kN/mm.sq.

Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction
    factor
Ks=1+0.5/C;// Shear stress factor
Fm=(Fmax+Fmin)/2;// N
Fa=(Fmax-Fmin)/2;// N
tau_m_into_d_sq=Ks*(8*Fm*C)/(%pi);// where
    tau_m_into_d_sq = tau_m*d**2
tau_a_into_d_sq=Kw*(8*Fa*C)/(%pi);// where
    tau_a_into_d_sq = tau_a*d**2

//(tau_m-tau_a)/Sys+2*tua_a/Ses_dash=1/n
d=sqrt(n)*sqrt((tau_m_into_d_sq-tau_a_into_d_sq)/Sys
    +2*tau_a_into_d_sq/Ses_dash);// mm
printf('\n diameter of spring wire = %.2f mm or %.f
    mm',d, ceil(d))
d=ceil(d);// mm
Dm=C*d;// mm
printf('\n Mean coil diameter = %.f mm', Dm)
//del=8*Fmax*Ci**3/(G*d)
i=(del/(8*Fmax*C**3/(G*10**3*d)));// no. of active
    coils
i=ceil(i);// no. of active coils
printf('\n no. of active coils = %.f',i)
nt=i+2;// no. of active coils (for square & ground
    ends)
```

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

Scilab code Exa 8.5 Design a spring for boiler safety

```
// exa 8.5 Pg 231
clc;clear;close;

// Given Data
p=125;// MPa
dv=60;// mm
del1=40;// mm
del2=20;// mm
tau_max=600;// MPa
G=85;// kN/mm.sq.
C=6;// spring index

Fv=(%pi/4)*dv**2*p/100;// N (Force on the valve)
del_max=del1+del2;// mm (Max. deflection)
Fmax=Fv*dv/del1;// N (Max. force)
Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction factor
// tau = 8*Fmax*C*Kw/(%pi*d**2)
d=sqrt((8*Fmax*C*Kw/(%pi))/tau_max);// mm (Diameter of spring wire)
Dm=6*d;// mm (Mean coil diameter)
n=G*10**3*d*del_max/(8*Fmax*C**3);// no. of turns
n = ceil(n);// no. of turns
nt=n+2;// total no. of turns
lf=nt*d+1.15*del_max;// mm (Free length)
p=lf/(nt-1);// mm (Pitch of coil)
printf('\n Force on the valve = %.1f N',Fv)
printf('\n Maximum deflection = %.f mm', del_max)
printf('\n Maximum force = %.1f N', Fmax)
printf('\n Wahl''s correction factor = %.4f ',Kw)
```

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

Scilab code Exa 8.7 Design the spring

```
// exa 8.7 Pg 232
clc;clear;close;

// Given Data
dv=30;// mm
Wv=10;// N
Wl=25;// N
lf=100;// mm
del1=20;// mm
p=3.5;// N/mm.sq.
valve_lift=2;// mm
C=6;// spring index
tau=500;// N/mm.sq.
G=0.84*10**5;// N/mm.sq.

W=(%pi/4)*dv**2*p;// N (load on the valve at
    operating condition)
W1=W-Wv;//N (Net load on the valve at operating
    condition)
//W1*100=Wl*150+S1*200+P*300 // taking momens about
    the fulcrum
//S1*200+P*300=W1*100-Wl*150 ...eqn(1)
valve_lift=20*100/200;// mm //from figure (when
    spring is extended by 20 mm)
spring_extension=2*200/100;// mm // from figure (
```

```
when valve is lifted 2 mm)
valve_load=W*12/10;// N // (when valve is lifted 2 mm)
W2=valve_load-Wv;// N // (when valve is lifted 2 mm)
del2=del1+4;// mm (when valve is lifted)
//S2=S1*del2/del1;// spring force when valve is lifted
//S1*del2/del1-s2=0 ... eqn(1)
//W2*100=Wl*150+S2*200+P*300 // taking momens about the fulcrum
//S2*200+P*300 =W2*100-Wl*150 ... eqn(2)
//S1*200+P*300=W1*100-Wl*150 ...eqn(3)
// solving above 3 eqn. by matrix method
A=[del2/del1 -1 0;200 0 300;0 200 300];
B=[0;W1*100-Wl*150;W2*100-Wl*150];
X=A**-1*B;// solution matrix
S1=X(1);// N
S2=X(2);// N
printf('\n Spring force when valve is lifted = %.1f N',S2)
printf('\n\n Design of spring - ')
k=(S2-S1)/(del2-del1);// N/mm (Spring stiffness)
printf('\n Spring stiffness = %.2f N/mm',k)
Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction factor
printf('\n Wahl''s correction factor = %.4f',Kw)
// tau=Kw*8*S2*C/%pi/d**2 max. shear stress
d=sqrt(Kw*8*S2*C/%pi/tau);// mm (spring diameter)
printf('\n spring diameter = %.2f mm or %.f mm',d,d)
d=ceil(d);// mm
// k=G*d/(8*C**3*n) (Spring stiffness)
n=G*d/(8*C**3*k);// no. of active coils
printf('\n no. of active coils = %.2f. Use n=7',n)
n=ceil(n);// rounding
nt=n+1;// total no. of active coils
printf('\n total no. of active coils = %.f',nt)
p=lf/(n-1);// mm (pitch of coils)
printf('\n pitch of coils = %.2f mm',p)
```

Scilab code Exa 8.8 Design a spring for a balance

```
// exa 8.8 Pg 234
clc;clear;close;

// Given Data
Fmin=0;// N
Fmax=1000;// N
del=80;// mm
Do=25;// mm
n=30;// no. of turns
G=85;// kN/mm.sq.

k=(Fmax-Fmin)/del;// N/mm (Spring stiffness)
printf('\n Spring stiffness = %.1f N/mm',k)
// k=G*d/(8*C**3*n) (Spring stiffness)
C_cube_BY_d=G*10**3/(k*8*n);//

function [C,d]=hitntrial(c3d,Do)
    for C=5:-0.1:4.5
        d=C**3/(c3d);
        Doo=d*C+C;
        if Doo<Do
            break;
        end;
    end

endfunction;

[C,d]=hitntrial(C_cube_BY_d,Do)
printf('\n By hit and trial method and using value
    of C^3/d -')
printf('\n value of Spring Index, C = %.1f ',C)
printf('\n value of wire diameter, d = %.1f mm',d)
```

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

Scilab code Exa 8.10 Wire diameter of spring

```
// exa 8.10 Pg 235
clc;clear;close;

// Given Data
Fmin=600;// N
Fmax=1000;// N
C=6;// spring index
n=1.5;// factor of safety
Sys=700;// N/mm.sq.
Ses_dash=350;// N/mm.sq.

Kw=(4*C-1)/(4*C-4)+0.615/C;// Wahl's correction
    factor
Ks=1+0.5/C;// Shear Stress factor
Fm=(Fmax+Fmin)/2;// N
Fa=(Fmax-Fmin)/2;// N
tau_m_into_d_sq=Ks*(8*Fm*C)/(%pi);// where
```

```
tau_m_into_d_sq = tau_m*d**2
tau_a_into_d_sq=Kw*(8*Fa*C)/(%pi);// where tau_a_into_d_sq = tau_a*d**2

//(tau_m-tau_a)/Sys+2*tua_a/Ses_dash=1/n
d=sqrt(n)*sqrt((tau_m_into_d_sq-tau_a_into_d_sq)/Sys+2*tau_a_into_d_sq/Ses_dash);// mm
printf('wire diameter of spring = %.2f mm',d)
```

Scilab code Exa 8.11 Design a helical spring

```
// exa 8.11 Pg 236
clc;clear;close;

// Given Data
dv=100;//mm
C=5.5;// spring index
pi=1;// N/mm.sq.
p=1.075;// N/mm.sq.
del=6;// mm
tau_max=400;// N/mm.sq.
G=80;// kN/mm.sq.

Fi=(%pi/4)*dv**2*pi;// N (initial tension in spring)
printf('\n initial tension in spring = %.f N', Fi)
F=(%pi/4)*dv**2*p;// N (maximum tension in spring)
printf('\n maximum tension in spring = %.f N', F)
k=(F-Fi)/del;// N/mm (stiffness of spring)
printf('\n stiffness of spring = %.2f N/mm',k)
//Tmax=F*Dm/2 where Dm=5.5*d
Tmax_BY_d=F*5.5/2;// calculation
//Tmax=(%pi/16)*d**3*tau_max
d=sqrt(Tmax_BY_d/((%pi/16)*tau_max));// mm
printf('\n diameter of spring = %.2f mm. Use 18 mm.',d)
```

```
d=ceil(d);// mm (rounding)
Dm=5.5*d;//mm
printf('\n mean coil diameter = %.f mm',Dm)
Do=Dm+d;//mm
printf('\n outside coil diameter = %.f mm',Do)
Di=Dm-d;// mm
printf('\n initial coil diameter = %.f mm',Di)
n=G*10**3*d*del/8/(F-Fi)/C**3;// no. of turns
printf('\n no. of turns = %.f',n)
nt=n+1;// total no. of turns
printf('\n total no. of turns(for extension spring) = %.f',nt)
gi=1;// mm (initial gap)
lf=nt*d+(nt-1)*gi;// mm
printf('\n free length of spring = %.f mm',lf)
p=lf/(nt-1);//mm
printf('\n pitch of coils = %.2f mm',p)
```

Scilab code Exa 8.12 Find the axial load

```
// exa 8.12 Pg 236
clc;clear;close;

// Given Data
d=6;//mm
Do=75;// mm
tau=350;// N/mm.sq.
G=84;// kN/mm.sq.

printf('\n (i) neglecting the effect of curvature')
dm=Do-d;// mm
C=dm/d;// spring index
Ks=1+0.5/C;// shear stress factor
//tau=Ks*(8*Fmax*C)/(%pi*d**2)
Fmax=tau/(Ks*(8*C)/(%pi*d**2));// N
```

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

Chapter 9

Power Screws

Scilab code Exa 9.1 Stress in the screw

```
// exa 9.1 Pg 256
clc;clear;close;

// Given Data
d=26;// mm
p=5;// mm
W=10;// kN
Do=50;// mm
Di=20;// mm
mu=0.2;// coefficient of thread friction
mu_c=0.15;// coefficient of collar friction
N=15;// rpm
pb=6;// MPa

dm=d-p/2;// mm
dc=d-p;// mm
t=p/2;//mm
l=2*p;// mm
alfa=atand(l/(%pi*dm));// degree
fi=atand(mu);// degree
Tf=W*dm/2*tand(alfa+fi);// N.mm
```

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

Scilab code Exa 9.2 Power required to drive the screw

```
// exa 9.2 Pg 257
clc;clear;close;

// Given Data
d=50;// mm
p=8;// mm
W=2;// kN
Do=100;// mm
Di=50;// mm
mu=0.15;// coefficient of thread friction
mu_c=0.10;// coefficient of collar friction
N=25;// rpm
two_beta=29;// degree

dm=d-p/2;// mm
dc=d-p;// mm
t=p/2;//mm
```

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

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

```
// exa 9.3 Pg 259
clc;clear;close;

// Given Data
d=10;// mm
p=3;// mm
mu=0.15;// coefficient of thread friction
mu_c=0.20;// coefficient of collar friction
dc=15;// mm
F=60;// N
W=4;// kN
two_beta=30;// degree
h=25;// mm
lf=150;// mm (screw free length)

dm=d-p/2;// mm
alfa=atand(p/(%pi*dm));// degree
mu_e=mu/cosd(two_beta/2);// virtual coefficient of
    friction
```

```
fi=atand(mu_e);// degree
Tf=W*10**3*dm/2*tand(alfa+fi);// N.mm
Tc=mu_c*W*10**3/2*dc;// N.mm
T=Tf+Tc;// N.mm
//F*l=T
l=T/F;// mm (Length of handle)
printf('\n (a) Length of handle = %.1f mm',l)

printf('\n\n (b) Maximum shear stress in screw')
printf('\n Section 1-1 : ')
dc=d-p;//mm
tau=16*T/(%pi*dc**3);// N/mm.sq.
M=F*lf;// N.mm
sigma_b=32*M/(%pi*dc**3);// N/mm.sq.
tau_max=sqrt((sigma_b/2)**2+tau**2);// MPa
printf('\n Maximum shear stress = %.2f MPa',tau_max)
printf('\n Section 2-2 : ')
sigma_c=4*W*10**3/(%pi*dc**2);// N/mm.sq. (Direct
    compressive stress)
tau2=16*Tc/(%pi*dc**3);//;// N/mm.sq. (Tortional
    shear stress)
tau_max=sqrt((sigma_c/2)**2+tau2**2);// MPa
printf('\n Maximum shear stress = %.2f MPa',tau_max)

//h=n*p;// height of nut
n=ceil(h/p);// no. of threads
t=p/2;// mm (thickness of threads)
pb=W*10**3/(%pi*dm*t*n);// MPa
printf('\n\n (b) Bearing pressure on threads = %.1f
    MPa',pb)
```

Scilab code Exa 9.4 Power required to drive the slide

```
// exa 9.4 Pg 260
clc;clear;close;
```

```
// Given Data
W=25;// kN
two_beta=29;// degree
v=0.96;// m/min
mu=0.14;// coefficient of thread friction
Di=30;// mm
Do=66;// mm
mu_c=0.15;// coefficient of collar friction
d=36;// mm
p=6;// mm
Sut=630;// MPa
Syt=380;// MPa

dm=d-p/2;// mm
dc=d-p;// mm
l=2*p;// mm
alfa=atand(l/(%pi*dm));// degree
mu_e=mu/cosd(two_beta/2);// virtual coefficient of friction
fi=atand(mu_e);// degree
Tf=W*10**3*dm/2*tand(alfa+fi);// N.mm
Tc=mu_c*W*10**3/4*(Do+Di);// N.mm
T=Tf+Tc;// N.mm
N=v*10**3/l;// rpm

P=2*%pi*N*T/(60*10**3)*10**-3;// kW
printf('\n Power required to drive the slide = %.2f kN',P)
sigma_c=4*W*10**3/(%pi*dc**2);// MPa
tau=16*T/(%pi*dc**3);// MPa
sigma1=1/2*(sigma_c+sqrt(sigma_c**2+4*tau**2));// MPa
tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
n_t=Syt/sigma1;// factor of safety in tension
printf('\n factor of safety in tension = %.2f ',n_t)
n_s=Syt/2/tau_max;// factor of safety in shear
printf('\n factor of safety in shear = %.2f ',n_s)
```

```
// Note- Answer in the textbook are not accurate.
```

Scilab code Exa 9.5 `Clamping force between jaws`

```
// exa 9.5 Pg 262
clc;clear;close;

// Given Data
d=12;// mm
dc=10;// mm
p=2;// mm
Do=10;//mm
mu=0.15;// coefficient of thread friction
mu_c=0.18;// coefficient of collar friction
F=100;// N
l=150;// mm

dm=dc+p/2;// mm
alfa=atand(p/(%pi*dm));// degree
fi=atand(mu);// degree
TfByW=dm/2*tand(alfa+fi);// where TfByW = Tf/W
TcByW=mu_c/3*Do;// where TcByW = Tc/W
TByW=TfByW+TcByW;// N.mm (total torque at B-B)
Tapplied=F*l;// N.mm (torque applied by the operator)
//putting T= Tapplied
W= Tapplied/TByW;// N
printf('\n (a) Clamping force between the jaws = %.f N',W)
eta=W*dm/2*tand(alfa)/Tapplied*100;// %
printf('\n (b) Efficiency of vice = %.2f %%',eta)
Tf=TfByW*W;// N.mm
printf('\n (c) Torque at A-A, Tf = %.1f N.mm & Torque at B-B = %.f N.mm',Tf,Tapplied)
// Note- Answer in the textbook are not accurate.
```

Scilab code Exa 9.6 Design a screw jack

```
// exa 9.6 Pg 267

clc;clear;close;

// Given Data
W=100;// kN
lift=400;// mm
sigma_ts=100;// MPa
sigma_cs=100;// MPa
tau_s=60;// MPa
tau_tn=50;// MPa
sigma_cn=45;// MPa
tau_n=40;// MPa
pb=15;// MPa
mu=0.2;// coefficient of thread friction
mu_c=0.15;// coefficient of collar friction

//sigma_cs=4*W/(%pi*dc**2)
dc=sqrt(4*W*10**3/(%pi*sigma_cs));// mm
printf('\n Screw Diameter-\n Core diameter of screw,
    dc=%.2f mm. Use dc=40 mm',dc)
dc=40;// mm
p=7;// mm (for normal series square threads)
d=dc+p;//mm
printf('\n outside diameter = %.f mm',d)
dm=dc+p/2;// mm
printf('\n mean diameter = %.1f mm',dm)
t=p/2;// mm
printf('\n thread thickness = %.1f mm',t)

printf('\n Maximum tensile & shear stress in screw -
    ')
```

```
sigma_c=4*W*1000/%pi/dc**2;// MPa
alfa=atand(p/(%pi*dm));// degree
fi=atand(mu);// degree
Tf=dm*W*10**3/2*tand(alfa+fi);// where TfByW = Tf/W
tau=16*Tf/(%pi*dc**3);// MPa
sigma12=(1/2)*(sigma_c+sqrt(sigma_c**2+4*tau**2));// MPa
printf('\n Maximum tensile stress = %.f MPa < %.f MPA. Hence design is safe.',sigma12,sigma_ts)
tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
printf('\n Maximum shear stress = %.2f MPa < %.f MPA. Hence design is safe.',tau_max,tau_s)

printf('\n Height of nut-')
n=W*10**3/(%pi/4)/pb/(d**2-dc**2);// no. of threads
n= ceil(n);// no. of threads (rounding)
h=n*p;// mm
printf('\n h=%.f mm',h)

printf('\n Check for stress in screw and nut')
tau_screw=W*10**3/(%pi*n*dc*t);// MPa
printf('\n shear stress in screw = %.2f MPa < %.f MPa',tau_screw,tau_s)
tau_nut=W*10**3/(%pi*n*d*t);// MPa
printf('\n shear stress in nut = %.2f MPa < %.f MPa',tau_nut,tau_n)
printf('\n These are within permissible limits. Hence design is safe.')

printf('\n Nut collar size-')
// %pi/4*(D1**2-d**2)*sigma_tn=W
D1=sqrt(W*10**3/(%pi/4)/tau_tn+d**2);// mm
printf('\n Inside diameter of collar = %.2f mm. Use D1=70 mm',D1)
D1=70;//mm (adopted for design)
// %pi/4*(D2**2-D1**2)*sigma_cn=W
D2=sqrt(W*10**3/(%pi/4)/sigma_cn+D1**2);// mm
printf('\n Outside diameter of collar = %.2f mm. Use
```

```
    D2=90 mm',D2)
D2=90;//mm (adopted for design)

// %pi*D1*tc*tau_n=W
tc=W*10**3/(%pi*D1*tau_n);// mm
printf('\n thickness of nut = %.2f mm. Use tc=12 mm.
    ',tc)
tc=12;// mm (adopted for design)

printf('\n Head Dimensions-')
D3=1.75*d;// mm
printf('\n Diameter of head on top of screw = %.2f
    mm. use D3=84 mm.',D3)
D3=84;// mm (adopted for design)
D4=D3/4;// mm
printf('\n pin diameter in the cup = %.f mm',D4)

printf('\n Torque required between cup and head-')
Tc=mu_c*W*10**3/3*((D3**3-D4**3)/(D3**2-D4**2));// N
    .mm
printf('\n Tc=%.f N.mm (acc. to uniform pressure
    theory)',Tc)
T=Tf+Tc;// N.mm
printf('\n Total Torque, T=%.f N.mm',T)

F=300;// N (as a normal person can apply 100-300 N)
l=T/F;//mm
printf('\n length of lever = %.f mm. Use 3300 mm',l)

M=F*l;// N.mm
dl=(32*M/%pi/sigma12)**(1/3);// mm
printf('\n Diameter of lever, dl=%.1f mm. Use dl=48
    mm.',dl)
dl=48;// mm (adopted for design)

H=2*dl;// mm
printf('\n Height of head, H=%.f mm',H)

```

```
printf('\n Check for screw in buckling-')
L=lift+0.5*h;// mm
K=dc/4;// mm
C=0.25;// spring index
sigma_y=200;// MPa
Ac=%pi/4*dc**2;//mm.sq.
Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(L/K)**2)/1000;// kN
printf('\n Buckling or critical load for screw, Wcr = %.f kN > 100kN',Wcr)

To=W*10**3*dm/2*tand(alfa);// N.mm
eta=To/T*100;// %
printf('\n Efficiency of screw = %.1f %%',eta)

printf('\n Body dimensions-')
D5=1.5*D2;// mm
t2=2*tc;// mm
t3=0.25*d;//mm
D6=2.25*D2;// mm
printf('\n Diameter of body at top, D5 = %.f mm', D5)
printf('\n Thickness of base, t2 = %.f mm', t2)
printf('\n Thickness of body, t3 = %.f mm', t3)
printf('\n Inside diameter of bottom, D6 = %.1f mm. Use D6=205 mm.', D6)
D6=205;// mm (adopted for design)
D7=1.75*D6;// mm
hb=lift+h+100;// mm
printf('\n Outside diameter at the bottom, D7 = %.2f mm. Use 360 mm.', D7)
printf('\n Height of body = %.f mm. Use 600mm',hb)
```

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

```
// exa 9.7 Pg 267

clc;clear;close;

// Given Data
two_beta=30;// degree
W=400*10**3;// N
d=100;// mm
p=12;// mm
mu=0.15;// coefficient of thread friction

dm=d-p/2;// mm
dc=d-p;// mm
l=2*p;// mm
alfa=atand(l/%pi/dm);// degree
mu_e=mu/cosd(two_beta/2);// virtual coefficient of friction
fi=atand(mu);// degree
Tf=W*dm/2*tand(alfa+fi);// N.mm (Frictional torque for raising load)
T=W*dm/4*tand(fi);// N.mm
To=W*dm/2*tand(alfa);// N.mm (Torque without friction)
eta1=To/Tf*100;// %
printf('\n Efficiency during raising the load = %.2f %%',eta1)
eta2=T/To*100;// %
printf('\n Efficiency during lowering the load = %.2f %%',eta2)
// Note - answer & solution is wrong in the textbook.
```

Scilab code Exa 9.9 Safe capacity of press

```
// exa 9.9 Pg 272
```

```
clc;clear;close;

// Given Data
d=70;// mm
mu=0.13;// coefficient of thread friction
mu_c=0.15;// coefficient of collar friction
Do=90;// mm
Di=26;// mm
L=450;// mm
// C-25 steel screw
sigma_t1=275;// MPa
sigma_c1=275;// MPa
tau1=137.5;// MPa
// Phosphor bronze nut
sigma_t2=100;// MPa
sigma_c2=90;// MPa
tau2=80;// MPa
pb=15;//MPa
n=2;// factor of safety
//screw
sigma_ts=137.5;// MPa
sigma_cs=137.5;// MPa
tau_s=68.75;// MPa
//Nut
sigma_tn=50;// MPa
sigma_cn=45;// MPa
tau_n=40;// MPa

p=10;// mm (for normal series square threads)
dc=d-p;//mm
dm=d-p/2;//mm
t=p/2;//mm
alfa=atand(p/%pi/dm);// degree
fi=atand(mu);// degree

K=dc/4;// mm
C=0.25;// spring index
```

```
sigma_y=275;// MPa
Ac=%pi/4*dc**2;//mm.sq.
Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(L/K)**2);// N
printf('\n (a) Safe Capacity of press or critical load for the screw = %.f N',Wcr)

n=Wcr/(%pi*dm*t*pb);// no. of threads
n=ceil(n);// rounding
h=n*p;// mm
printf('\n (b) Height of nut, h=%.f mm',h)

W=Wcr;// N
Tf=W*dm/2*tand(alfa+fi)/1000;// N.mm (Frictional torque)
Tc=mu_c*W/4*(Do+Di)/1000;// N.mm (Collar torque)
T=Tf+Tc;// N.mm
printf('\n (c) Necessary torsional moment or total torque = %.2f N.mm',T)
// Note - answer in the textbook is wrong.
```

Scilab code Exa 9.11 Determine the force

```
// exa 9.11 Pg 273

clc;clear;close;

// Given Data
d=26;// mm
L=0.25;//m
F=300;// N
mu=0.14;// coefficient of thread friction
p=5;// mm (for normal series)

dc=d-p;// mm
```

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

Scilab code Exa 9.13 Design a screw jack

```
// exa 9.13 Pg 274

clc;clear;close;

// Given Data
W=50;// kN
lift=200;// mm
gc=300;// mm (ground clearance)
pb=16;// MPa
mu=0.14;// coefficient of collar friction

//Screw C-35
Sut=288;// MPa
n=3;// factor of safety for screw
// Nut : phosphor-bronze
sigma_t=100;// MPa
sigma_c=90;// MPa
tau=80;// MPa
n2=3;// factor of safety for nut

sigma_ts=Sut/n;// MPa
```

```
sigma_cs=Sut/n;// MPa
tau_s=sigma_ts/2;// MPa
// sigma_cs=4*W/(%pi*dc**2)
dc= sqrt(4*W*10**3/(%pi*sigma_cs));// mm
printf('\n Screw diameter - \n Core diameter, dc = %.2f mm. Use 30 mm',dc)
dc=30;// mm (adopted for design)
p=6;// mm (for normal series square threads)
d=dc+p;//mm
printf('\n outside diameter = %.f mm',d)
dm=dc+p/2;// mm
printf('\n mean diameter = %.1f mm',dm)
t=p/2;// mm
printf('\n thread thickness = %.1f mm',t)

printf('\n Maximum tensile & shear tress in screw -')
sigma_c=4*W*1000/%pi/dc**2;// MPa
alfa=atand(p/(%pi*dm));// degree
fi=atand(mu);// degree
Tf=dm*W*10**3/2*tand(alfa+fi);// where TfByW = Tf/W
tau=16*Tf/(%pi*dc**3);// MPa
sigma12=(1/2)*(sigma_c+sqrt(sigma_c**2+4*tau**2));// MPa
printf('\n Maximum tensile stress = %.1f MPa < %.f MPA. Hence design is safe.',sigma12,sigma_ts)
tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
printf('\n Maximum shear stress = %.2f MPa < %.f MPA. Hence design is safe.',tau_max,tau_s)

printf('\n Height of nut-')
n=W*10**3/(%pi/4)/pb/(d**2-dc**2);// no. of threads
n= round(n);// no. of threads (rounding)
h=n*p;// mm
printf('\n h=%.f mm',h)

printf('\n Check for stress in screw and nut')
tau_screw=W*10**3/(%pi*n*dc*t);// MPa
```

```
printf('\n shear stress in screw = %.2f MPa\',tau_screw)
tau_nut=W*10**3/(%pi*n*d*t);// MPa
printf('\n shear stress in nut = %.2f MPa',tau_nut)
printf('\n These are within permissible limits. Hence design is safe.')

printf('\n Nut collar size-')
// %pi/4*(D1**2-d**2)*sigma_tn=W
D1=sqrt(W*10**3/(%pi/4)/(50)+d**2);// mm
printf('\n Inside diameter of collar = %.2f mm. Use D1=52 mm',D1)
D1=52;//mm (adopted for design)
// %pi/4*(D2**2-D1**2)*sigma_cn=W
D2=sqrt(W*10**3/(%pi/4)/45+D1**2);// mm
printf('\n Outside diameter of collar = %.1f mm. Use D2=65 mm',D2)
D2=65;//mm (adopted for design)

// %pi*D1*tc*tau_cn=W
tau_cn=40;// MPa
tc=W*10**3/(%pi*D1*tau_cn);// mm
printf('\n thickness of nut = %.2f mm. Use tc=8 mm.',tc)
tc=8;// mm (adopted for design)

printf('\n Head Dimensions-')
D3=1.75*d;// mm
printf('\n Diameter of head on top of screw = %.2f mm. use D3=64 mm.',D3)
D3=64;// mm (adopted for design)
D4=D3/4;// mm
printf('\n pin diameter in the cup = %.f mm',D4)

printf('\n Torque required between cup and head-')
Tc=mu*W*10**3/3*((D3**3-D4**3)/(D3**2-D4**2));// N.mm
printf('\n Tc=%.f N.mm (acc. to uniform pressure
```

```
theory)',Tc)
T=Tf+Tc;// N.mm
printf('\n Total Torque, T=%.f N.mm',T)

F=300;// N (as a normal person can apply 100-300 N)
l=T/F;//mm
printf('\n length of lever = %.f mm. Use 1075 mm',l)

M=F*l;// N.mm
dl=(32*M/%pi/sigma12)**(1/3);// mm
printf('\n Diameter of lever, dl=%.1f mm.',dl)

H=2*dl;// mm
printf('\n Height of head, H=%.f mm',H)

printf('\n Check for screw in buckling-')
L=lift+0.5*h;// mm
K=dc/4;// mm
C=0.25;// spring index
sigma_y=288;// MPa
Ac=%pi/4*dc**2;//mm.sq.
Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(
    L/K)**2)/1000;// kN
printf('\n Buckling or critical load for screw, Wcr
    = %.f kN > 50kN',Wcr)
printf('\n Hence design is safe.')
```

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

```
// exa 9.14 Pg 278

clc;clear;close;

// Given Data
d=32;// mm
```

```
p=5;// mm
W=12;// kN
D3=50;// mm
D4=20;// mm
mu=0.15;// coefficient of thread friction
mu_c=0.20;// coefficient of collar friction
N=24;// rpm
pb=6;// N/mm.sq.
tau_s=30;// MPa
tau_n=30;// MPa

dm=d-p/2;// mm
dc=d-p;// mm
t=p/2;// mm
l=2*p;//mm
alfa=atand(l/%pi/dm);// degree
fi=atand(mu);// degree
Tf=W*10**3*dm/2*tand(alfa+fi);// N.mm
Tc=mu_c*W*10**3/4*(D3+D4);// N.mm
T=Tf+Tc;// N.mm
printf('\n (i) Torque required to rotate the screw =
    %.f N.mm',T)

printf('\n (ii) Stresses induced in screw - ')
sigma_c=4*W*10**3/(%pi*dc**2);// N/mm.sq.
printf('\n Direct compressive stress = %.2f N/mm.sq'
   ,sigma_c)
tau=16*T/(%pi*dc**3);// N/mm.sq.
printf('\n Tortional shear stress = %.2f N/mm.sq',
   tau)
tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
printf('\n Maximum shear stress = %.2f MPa < %.f MPa
   ',tau_max,tau_s)
printf('\n Hence design is safe.')
n=W*10**3/(%pi*dm*t*pb);// no. of threads
n=ceil(n);// rounding
h=n*p;//mm
printf('\n (iii) Height of nut = %.f mm',h)
```

Scilab code Exa 9.15 Design a simple screw jack

```
// exa 9.15 Pg 279

clc;clear;close;

// Given Data
W=100;// kN
lift=260;// mm
pb=15;// N/mm.sq.
mu=0.15;// coefficient of thread friction
mu_c=0.20;// coefficient of collar friction
//Screw
Suts=800;// N/mm.sq.
sigma_ss=340;// N/mm.sq.
ns=4;// factor of safety
//Nut
Sutn=552;// N/mm.sq.
sigma_sn=260;// N/mm.sq.
nn=5;// factor of safety

sigma_ts=Suts/ns;// N/mm.sq.
sigma_cs=Suts/ns;// N/mm.sq.
tau_s=sigma_ss/ns;// N/mm.sq.
sigma_tn=Sutn/nn;// N/mm.sq.
sigma_cn=Sutn/nn;// N/mm.sq.
tau_n=sigma_sn/nn;// N/mm.sq.

//sigma_cs=4*W/(%pi*dc**2)
dc=sqrt(4*W*10**3/(%pi*sigma_cs));// mm
printf('\n Screw Diameter-\n Core diameter of screw, dc=%.2f mm. Use dc=33 mm',dc)
dc=33;// mm
p=7;// mm (for normal series square threads)
```

```
d=dc+p;//mm
printf('\n outside diameter = %.f mm',d)
dm=dc+p/2;// mm
printf('\n mean diameter = %.1f mm',dm)
t=p/2;// mm
printf('\n thread thickness = %.1f mm',t)

printf('\n Maximum stresses in screw -')
sigma_c=4*W*1000/%pi/dc**2;// MPa
alfa=atand(p/(%pi*dm));// degree
fi=atand(mu);// degree
Tf=dm*W*10**3/2*tand(alfa+fi);// where TfByW = Tf/W
tau=16*Tf/(%pi*dc**3);// MPa
sigma12=(1/2)*(sigma_c+sqrt(sigma_c**2+4*tau**2));// MPa
printf('\n Maximum tensile stress = %.1f N/mm.sq. < %.f N/mm.sq.. Hence design is safe.',sigma12,sigma_ts)
tau_max=sqrt((sigma_c/2)**2+tau**2);// MPa
printf('\n Maximum shear stress = %.2f N/mm.sq. < %.f N/mm.sq.. Hence design is safe.',tau_max,tau_s)

printf('\n Height of nut-')
n=W*10**3/(%pi/4)/pb/(d**2-dc**2);// no. of threads
n= ceil(n);// no. of threads (rounding)
h=n*p;// mm
printf('\n h=%.f mm. Use 120 mm.',h)
h=120;// mm

printf('\n Check for stress in screw and nut')
tau_screw=W*10**3/(%pi*n*dc*t);// MPa
printf('\n shear stress in screw = %.2f MPa < %.f MPa',tau_screw,tau_s)
tau_nut=W*10**3/(%pi*n*d*t);// MPa
printf('\n shear stress in nut = %.2f MPa < %.f MPa',tau_nut,tau_n)
printf('\n These are within permissible limits. Hence design is safe.')
```

```

printf('\n Nut collar size-')
// %pi/4*(D1**2-d**2)*sigma_tn=W
D1=sqrt(W*10**3/(%pi/4)/sigma_tn+d**2);// mm
printf('\n Inside diameter of collar = %.2f mm. Use D1=55 mm',D1)
D1=55;//mm (adopted for design)
// %pi/4*(D2**2-D1**2)*sigma_cn=W
D2=sqrt(W*10**3/(%pi/4)/sigma_cn+D1**2);// mm
printf('\n Outside diameter of collar = %.2f mm. Use D2=70 mm',D2)
D2=70;//mm (adopted for design)

// %pi*D1*tc*tau_n=W
tc=W*10**3/(%pi*D1*tau_n);// mm
printf('\n thickness of nut = %.f mm. Use tc=15 mm.',tc)
tc=15;// mm (adopted for design)

printf('\n Head Dimensions-')
D3=1.75*d;// mm
printf('\n Diameter of head on top of screw = %.2f mm.',D3)
D4=D3/4;// mm
printf('\n pin diameter in the cup = %.1f mm. Use 20 mm.',D4)
D4=20;// mm (adopted for design)

printf('\n Torque required between cup and head-')
Tc=mu_c*W*10**3/3*((D3**3-D4**3)/(D3**2-D4**2));// N.mm
printf('\n Tc=%.f N.mm (acc. to uniform pressure theory)',Tc)
T=Tf+Tc;// N.mm
printf('\n Total Torque, T=%.f N.mm',T)

F=300;// N (as a normal person can apply 100-300 N)
l=T/F;//mm
```

```
printf('\n length of lever = %.f mm or %.2f m',l,l/1000)

M=F*l;// N.mm
sigma_b=100;// N/mm.sq. (assumed)
dl=(32*M/%pi/sigma_b)**(1/3);// mm
printf('\n Diameter of lever, dl=%.1f mm. Use dl=45 mm.',dl)
dl=45;// mm (adopted for design)

H=2*dl;// mm
printf('\n Height of head, H=%.f mm',H)

printf('\n Check for screw in buckling-')
L=lift+0.5*h;// mm
K=dc/4;// mm
C=0.25;// spring index
sigma_y=200;// MPa
Ac=%pi/4*dc**2;//mm.sq.
Wcr=Ac*sigma_y*(1-(sigma_y/4/C/%pi**2/(200*10**3))*(L/K)**2)/1000;// kN
printf('\n Buckling or critical load for screw, Wcr = %.f kN > 100kN',Wcr)

To=W*10**3*dm/2*tand(alfa);// N.mm
eta=To/T*100;// %
printf('\n Efficiency of screw = %.2f %%',eta)

printf('\n Body dimensions-')
D5=1.5*D2;// mm
t2=2*tc;// mm
t3=0.25*d;//mm
D6=2.25*D2;// mm
printf('\n Diameter of body at top, D5 = %.f mm', D5)
printf('\n Thickness of base, t2 = %.f mm', t2)
printf('\n Thickness of body, t3 = %.f mm', t3)
printf('\n Inside diameter of bottom, D6 = %.1f mm.
```

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