Scilab Textbook Companion for A Textbook of Production Engineering by P. C. Sharma[1]

Created by
Mayank Sahu
B.E.
Mechanical Engineering
M. I. T. S Gwalior, M.P
College Teacher
None
Cross-Checked by
Bhavani Jalkrish

November 3, 2014

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

Book Description

Title: A Textbook of Production Engineering

Author: P. C. Sharma

Publisher: S. Chanda & Company, New Delhi

Edition: 11

Year: 2008

ISBN: 9788121901116

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 2

Press Tool Design

Scilab code Exa 2.1 find total pressure and dimensions

```
clc
D = 50 // Diameter of washer in mm
t = 4 // thickness of material in mm
d = 24 // diameter of hole in mm
p = 360 // shear strength of material in N/mm^2
F1 = %pi*D*t*p // blanking pressure in N
F2 = %pi*d*t*p // piercing pressure in N
F = F1 + F2 // total pressure in N
d1 = d + 0.4 // piercing die diameter in mm
d2 = D - 0.4 // blank punch diameter in mm
c = 0.8*F // press capacity in N
printf("\n Blanking pressure = %d kN\n Piercing pressure = %0.3f KN\n Total pressure required = %0.1f KN" ,F1/1000,F2/1000,F/1000)
printf("\n piercing punch diameter = %0.2f cm\n blanking punch diametre = %0.2f cm \n press capacity = %0.2f KN\n", d1/10 , d2/10 , c/1000)
// Answers vary due to round off error
```

Scilab code Exa 2.2 To find number of draws

```
clc
h = 10 // height of cup in cm
d = 5 // diameter of cup in cm
D = sqrt(d^2 + 4*d*h) // blank diameter in cm
// height daimeter ratio is 2 . Therefore from table
    2.9 total reductions are 3
r1 = 0.45*D // first reduction is 45%
d1 = D - r1 // diameter at first draw in cm
r2 = d1*0.25 // second reduction is 25%
d2 = d1 - r2 // diameter at second draw in cm
r3 = d2*0.2 // third reduction is 20%
d3 = d2 - r3 // diameter at third draw in cm
printf("\n Diameter at first draw = %0.2f cm\n
   Diameter at second draw = %0.2f cm\n Diameter at
    third draw =%0.3f cm" , d1 , d2 , d3)
```

Scilab code Exa 2.3 To calculate bending force

```
clc
K = 1.20 // die-opening factor
L = 37.5 // Length of strip in cm
T = 2.5 // thickness of strip in mm
sigma_ut = 630 // tensile strength in N/mm^2
W = 16*T // width of die opening in mm
F = (K*L*10*sigma_ut*T^2)/W // bending force in N
printf("\n bending force = %0.1f KN" , F/1000)
```

Scilab code Exa 2.4 find blanking force and work done

```
clc
b = 25 // width of blank in mm
```

```
l = 30 // length of blank in mm
tau = 450 // ultimate shear stress of material in N/
   mm^2
t = 1.5 // thickness of metal strip in mm
p = 2*(l + b) // perimeter of blank in mm
f = p*t*tau // blanking force in N
Pt = 0.25*t // punch travel in mm
w = f*Pt // work done in Nmm
printf("\n blanking force  = %0.2f KN\n work done  =
     %0.2f Nm", f/1000,w/1000)
```

Scilab code Exa 2.5 To find elastic recovery of material

```
clc
t = 1.5 // thickness in mm
c = 0.05*t // clearance in mm
D = 25.4 // outside diameter in mm
D_o = D - 0.05 // blank die opening in mm
B_s = D_o - 2*c // blanking punch size in mm
d = 12.7 // internal diameter in mm
P_s = d + 0.05 // piercing punch size in mm
D_s = P_s + 2*c // piercing die size in mm
printf("\n clearance = %0.3f mm\n blank die opening
    size = %0.2f mm ",c ,D_o)
printf("\n blanking punch size = %0.2f mm\n piercing
     punch size = %0.2f mm\n piercing die size = %0.2
    f mm",B_s,P_s,D_s )
```

Scilab code Exa 2.6 To find cutting forces

```
clc
D = 25.4 // outside diameter in mm
d = 12.7 // internal diameter in mm
```

```
t = 1.5 // thickness in mm
tau = 280 // ultimate shearing strength in N/mm^2
F = %pi*(D + d)*t*tau // total cutting force in N
F_s = %pi*D*t*tau // cutting force when punches are
    staggered in N
k = 0.6 // penetration
i = 1 // shear of punch in mm
F_p = (t*k*F)/(k*t +i)// cutting force when both
    punches act together in N
printf ("\n shear force when both punch act at same
    time and no shear is applied = %0.2f kN" , F
    /1000)
printf("\n cutting force when punches are staggered
    = %0.1f kN", F_s/1000)
printf("\n cutting force when there is penetration
    and shear on punch = %0.1f kN",F_p/1000)
```

Scilab code Exa 2.7 To calculate amount of shear

```
clc
D = 60 // hole diameter in mm
tau = 450 // ultimate shear strength in N/mm^2
t = 2.5 // thickness in mm
F = %pi*D*t*tau // maximum punch force in N
w = F*0.4*t // work done in Nm
f = F/2 // punching force in N
k = 0.4 // penetration percentage
i = k*t*(F-f)/f // shear on punch in mm
printf("\n Amount of shear on punch = %d mm" , i)
```

Scilab code Exa 2.8 To find economy of material

```
clc
```

```
// from fig 2.27
w = 2.5 // cm
t = 3.2 // strip thickness in mm
h = 10 // cm
a = t + 0.015*h*10 // back scrap and front scrap in
   mm
b = t // scrap bridge in mm
W = h + (2*a)/10 // width of strip in cm
W = ceil(W) // cm
s = w + b/10 // length of one piece of stock in cm
L = 240 // length of strip in cm
N = (L-b)/s // number of parts
y = L - (N*s + b/10)  // scrap remaining at the end
   in mm
printf("\n Value of back scrap = %0.1f mm\n Value of
    scrap bridge = %0.1f mm " , a , b  )
printf("\n Width of strip = %0.2f cm\n Length of one
    piece of stock needed to produce one part = %0.2
   f cm" , W , s)
printf("\n Number of parts = %0.1f blanks\n Scrap
   remaining at the end = %0.2f mm" , N , y)
// Answers vary due to round off error
```

Scilab code Exa 2.9 Calculations for designing drawing die

```
clc
// from figure 2.73
t = 0.8 // thickness in mm
d = 50 // shell diameter in mm
r = 1.6 // radius of bottom corner in mm
h = 50 // height in mm
D = sqrt(d^2 + 4*d*h) // shell blank size in mm
el = 6.4 // extra length required to add in shell
   blank size
D = D + el // mm
```

```
pr = 100*(1-(d/D)) // percentage reduction
ratio = h/d
n = 2 // number of draws
R1 = 45 // first reduction
D1 = D - R1*D/100 // diameter at first reduction in mm
R2 = 100*(1-(d/D1)) // second reduction
PR = 4*t // punch radius in mm
PR = ceil(PR)
DR = 6 // die radius in mm
DC1 = 0.87 // die clearance for first draw in mm
DC2 = 0.88 // die clearance for second draw in mm
PD2 = d - 2*t // punch diameter for second draw in mm
DD2 = PD2 + 2*DC2 // Die opening diameter for second draw in mm
PD1 = D1 - 2*t // punch diameter for first draw in mm
DD1 = D1 + 2*DC1 // Die opening diameter for first draw in mm
// Drawing pressure
c = 0.65 // constant
sigma = 427 // N/mm^2
F = %pi*d*t*sigma*(D/d-c) // Drawing pressure in mm
Bhp = 30.8 // blanking holding pressure in kN
pc = 150 // press capacity in kN
printf("\n (i) size of blank = %0.2f mm \n (ii) Percentage reduction = %0.1f percent \n (iii) Number of draws = %d \n (iv) Radius on punch = %d mm \n     Die Radius = %d mm \n (v) Die clearance for first draw = %0.2f mm \n     die clearance for second draw = %0.2f mm" , D , pr,n,PR,DR,DC1,DC2)
printf("\n    Punch diameter for second draw = %0.1f mm \n    Die opening diameter for second draw = %0.2f mm \n    Punch diameter for first draw  = %0.3f mm \n     Die opening diameter for first draw = %0.3f mm\n (vi) Drawing pressure = %0.2f mm \n (
```

```
    vii) Blank holding pressure = %d kN \n (viii)
    Press capacity = %d kN" , PD2 , DD2 ,PD1 ,DD1 , F
    /1000 ,Bhp , pc)
// Answers vary due to round off error
```

Scilab code Exa 2.10 Determine developed length

```
clc
// from figure 2.74
l1 = 76 - ( 2.3 + 0.90) // length1 in mm
l2 = 115 - (2.3 + 0.90) // length2 in mm
t = 2.3 // mm
r = 0.90 // inner radius in mm
k = t/3 // mm
B = 0.5*%pi*(r + k) // bending allowance in mm
d = l1 + l2 + B // developed length in mm
printf("\n Developed length = %0.2f mm" , d)
// Answers vary due to round off error
```

Scilab code Exa 2.11 To calculate bending force

```
clc
t = 3.2 // thickness of blank in mm
l = 900 // bending length in mm
sigma = 400 // ultimate tensile strength in N/mm^2
k = 0.67 // die opening factor
r1 = 9.5 // punch radius in mm
r2 = 9.5 // die edge radius in mm
c = 3.2 // clearance in mm
w = r1 + r2 + c // width between contact points
    batween die and punch in mm
F= (k*l*sigma*t^2)/w // bending force in N
F=floor(F/10)*10 // N
```

```
printf("\n bending force = %0.2f kN" ,F/1000)
```

Scilab code Exa 2.12 To calculate bending force

```
clc
k = 1.33 // die opening factor
l = 1200 // bend length in mm
sigma = 455 // ultimate tensile strength in N/mm^2
t = 1.6 // blank thickness in mm
w = 8*t // width of die opening in mm
F = k*l*sigma*t^2/w // bending force in N
printf("\n bending force  = %0.2f kN", F/1000)
```

Scilab code Exa 2.13 calculate capacity of double bending die

```
clc
c = 1.25 // clearance in mm
r1 = 3 // die radius in mm
r2 = 1.5 // punch radius in mm
sigma = 315 // ultimate tensile strength in MPa
t = 1 // thickness in mm
l = 50 // width at bend in mm
w = r1 + r2 +c // width between contact points on
    die and punch in mm
F = 0.67*l*sigma*t^2/w // bending force in N
F_p = 0.67*sigma*l*t // pad force in N
sigma_c = 560 // setting pressure in MPa
b1 = 2 // beads on punch
b = b1*r1 // mm
F_b = sigma_c*l*b // bottoming force in N
F_o = F_p + F_b // Force required when bottoming is
    used in N
```

```
F_n = F +F_p // Force required when bottoming is not
    used in N
printf("\n Force required when bottoming is used =
    %0.1f tonnes" ,F_o/(9.81*1000))
printf("\n Force required when bottoming is not used
    = %0.3f tonnes" , F_n/(9.81*1000))
```

Scilab code Exa 2.14 To calculate cutting force

```
clc
w = 2 // width in mm
t = 5 // thickness in mm
theta=6 // shear in degrees
tau = 382.5 // ultimate shear stress in MPa
F = w*t*tau*1000 // cutting force in N
l = t/sin(theta*%pi/180) // length to be cut in mm
F_i = l*t*tau // cutting force in N
printf("\n cutting force with parallel cutting edges
    = %0.3f MN\n cutting force when shear is
    considered = %0.2f kN ",F/10^6 ,F_i/1000)
```

Scilab code Exa 2.15 Determine blank and punch diameter

```
clc
d1 = 105 // inside diameter in mm
h = 90 // depth in mm
t = 1 // thickness in mm
D = sqrt(d1^2+4*d1*h) // blank diameter in mm
tr = t*100/D // thickness ratio
// from table safe drawing ratio is 1.82
r = 1.82 // draw ratio
d2 = D/r // diameter for first draw in mm
d = 130 // Let diameter for first draw in mm
```

```
ratio1 = D/d // D/d for first draw
ratio2 = d/d1 // D/d for second draw
h1 =((D)^2-(d)^2)/(4*d) // Depth of first draw in mm
sigma = 415 // N/mm^2
c = 0.65 // constant
pc = %pi*d*t*sigma(D/d - c) // press capacity in kN
printf("\n Blank diameter = %d mm \n Thickness ratio = %0.3f \n Diameter for first draw = %d mm \n Depth of first draw = %0.2f mm \n Press capacity = %0.2f kN" ,D,tr,d2,h1,pc/1000)
// Answers vary due to round off error
```

Scilab code Exa 2.16 To find drawing operations and force

```
clc
d = 80 // diameter in mm
h = 250 // height in mm
D = sqrt((d^2+4*d*h))/10 // blank diameter in cm
D1 = 0.5*D // diameter after first draw in cm
// let reduction be 40% in second draw
D2 = D1-0.4*D1 // diameter after scond draw in cm
R = (1 - (d/(10*D2)))*100 // percentage reduction for third draw
l1 = ((D)^2-(D1)^2)/(4*D1) // height of cup after first draw in cm
l2 = ((D)^2-(D2)^2)/(4*D2) // height of cup after first draw in cm
l3 = ((D)^2-(d/10)^2)/(4*d/10) // height of cup after first draw in cm
t = 3 // mm
sigma = 250 // N/mm^2
C = 0.66
F = %pi*d/10*t*sigma*((D*10/d)-C) // drawing force in kN
printf("\n Diameter after first draw = %0.1f \n
```

```
Diameter after second draw = %0.2f \n Percentage reduction after third draw = %d percent",D1,D2,R)
printf("\n Height of cup after first draw = %0.2f cm \n Height of cup after second draw = %0.2f cm\n Height of cup after third draw = %0.2f cm", l1,l2,l3)
printf("\n Drawing force = %0.3f kN",F/1000)
// Answers vary due to round off error
```

Scilab code Exa 2.17 Determine developed length

```
clc
// from figure 2.75 (a)
r1 = 30 // radius in mm
t = 10 // thickness in mm
h1 = 300 // height in mm
ir1 = r1-t // inner radius of bends in mm
L1 = h1-(ir1+t) // mm
alpha1 = 90 // degree
r2 = 2*t // mm
k = 0.33*t // mm
L2 = alpha1*2*%pi*(r2+k)/360 // mm
w = 200 // mm
L3 = w-2*(t+ir1)// mm
L4 = L2 //mm
h2 = 100 // mm
L5 = h2 -(t+ir1) // mm
r3 = 150 //mm
ir2 = r3 - t // inner radius in mm
alpha2 = 180 // degree
L6 = alpha2*2*%pi*(ir2+k)/360 // mm
dl = L1+L2+L3+L4+L5+L6 // Total developed length in mm
printf("\n Total developed length = %0.2f mm" , dl)
// Answers vary due to round off error
```

Chapter 4

Cost Estimating

Scilab code Exa 4.1 To calculate total cost and SP

```
clc
d_m = 5500 // cost of direct material in Rs
d_l = 3000 // manufacturing wages in Rs
// factory overhead is 100% 0f manufacturing wages
f_o = (100*d_l)/100 // factory overheads in Rs
FC = d_m + d_l + f_o // factory cost in Rs
nm_o = 15*FC/100 // non-manufacturing overheads in
   Rs
tc = FC+nm_o // total cost in Rs
p = 12*tc/100 // profit in Rs
sp = tc+p // selling price in Rs
printf("\n Total cost = Rs %d\n Selling price = Rs
   %d" , tc,sp)
```

Scilab code Exa 4.2 To find selling price

```
clc
// given
```

```
OS_RM = 20000 // opening stock of raw materials in
   Rs
CS_RM = 30000 // closing stock of raw materials in
   Rs
TP_RM = 170000 // total purchase in year in Rs
OS_FG = 10000 // opening stock of finished goods in
   Rs
CS_FG = 15000 // closing stock of finished goods in
   Rs
sales = 489500 // sales of finished goods in Rs
D_W = 120000 // direct wages in Rs
F_E1 = 120000 // factory expenses in Rs
NM_E = 50000 //  non-manufacturing expenses in Rs
DMC = OS_RM + TP_RM - CS_RM // direct material cost
FC = DMC + D_W + F_E1 // factory cost
TC = FC + NM_E // total cost
FG_S = OS_FG + TC - CS_FG // cost of finished goods
   sold in Rs
P = sales - FG_S // profit in Rs
F_E2 = (F_E1)/D_W*100 // factory expenses in percent
NM_C = (NM_E)/FC*100 // non-manufacturing expenses
   to factory cost
P_C = (P/FG_S)*100 // profit to cost of sales
dm = 20000 // direct material in Rs
dw = 30000 // direct wages in Rs
fe = dw // factory expenses
fc = dm+dw+fe // factory cost in Rs
nme = NM_C*fc/100 // non-manufacturing expenses in
   Rs
tc = fc+nme // total cost in Rs
p = (P_C*tc)/100 // profit in Rs
sp = tc+p // selling price in Rs
printf("\n Selling price = Rs %d" , sp)
```

Scilab code Exa 4.3 To find factory cost

```
clc
d = 38 // diameter of bar in mm
l = 25 // length of bar in mm
p = 8.6 // density gm/cm^3
g = 9.81 // acceleration due to gravity in m/s^2
w = (%pi*d^2*l*p*g)/(4*10^6) // weight of material
    in N
mc = w*1.625 // material cost in Rs
lc = (2*90)/60 // labour cost in Rs
fo = 0.5*lc // factory overheads in Rs
fc = mc + lc + fo // factory cost in Rs
printf("\n factory cost = Rs %0.2f ", fc )
// Answers vary due to round off error
```

Scilab code Exa 4.4 find production cost and time taken

```
clc
sp = 65 // selling price in Rs
profit = 0.2*sp // profit in Rs
tc = sp - profit // total cost in Rs
P = (sp - profit)/1.4 // production cost in Rs
DM = 15 // cost of direct material in Rs
W = (P - DM)/ 1.4 // direct labour cost in Rs
tt = W/2 // time taken in hours
printf("\n Time taken = %0.3f Hours" , tt )
// Answers vary due to round off error
```

Scilab code Exa 4.5 To find profit

```
clc
mp = 6000 // market price of machine in Rs
d = 0.2*mp // discount in Rs
sp = mp - d // selling price of factory in Rs
```

```
mc = 400 // material cost in Rs
lc = 1600 // labour cost in Rs
fo = 800 // factory overheads in Rs
F = mc + lc + fo // factory cost in Rs
se = 0.5*F // selling expenses in Rs
profit = sp - (F + se ) // Rs
printf("\n profit =  Rs %d" , profit)
```

Scilab code Exa 4.6 To find lot size and time

```
clc
a = 1500 // requirements of components
s = 30 // cost of each set up in Rs
k = 0.2 // charge factor
c = 5 // cost of each part in Rs
N = 5*sqrt(a*s)/(k*c) // economic lot size
printf("\n Economic lot size = %d pieces", N)
S = (N*s)/a // time for each set up in hours
printf("\n Time for each set up = %0.2f hours", S )
// Answers vary due to round off error
```

Scilab code Exa 4.7 To find time to change cutter

```
clc
Tc = 2 // time taken by cutter per cycle in minutes
Tk = 10 // time taken to change cutter in minutes
T = 240 // tool life in minutes
t = (Tc*Tk)/T // time to change the cutter in min.
printf("\n Unit time to change the cutter = %0.3f
    min" , t )
// Error in textbook
```

Scilab code Exa 4.8 To find tool change time

```
clc
Tk = 360 // time taken by tool to cut before
   sharpening in min.
Tc = 20 // time taken to change the tool in min.
T = 4320 // time taken before it is discarded in min
   .
t = (Tc*Tk)/T //  tool change time per cycle in min.
printf("\n Unit tool change time per cycle = %0.2f
   min" , t )
```

Scilab code Exa 4.9 To calculate measuring time allowance

```
clc
Tc = 10 // time taken to check hole in secs
F = 2 // frequency of checking dimension
tc = Tc*F // time taken to check one piece in secs
N = 200 // number of pieces
Tc = tc*(N + 1) // Total time in sec
printf("\n Total time taken to check dimensions = %d
    min" , Tc/60)
```

Scilab code Exa 4.10 To find direct labour cost

```
clc
forgings = 40
setup = 4
Tc = 12 // machining time in min. per forging
```

```
nmt = 21 // non-machining in min. per forging
st = 45 // set up time per set up
ts = 5 // total sharpening in min. per forging
f = 20 // fatigue in percent
f = f/100
pn = 5 // personal needs in percent
pn = pn/100
Tk = 10 // tool chanhe time in min.
T = 8 // tool life in hours
ct = 15 // checking time with 5 checks in 15 secs
R = 1.4 // performance factor
dlc = 5 // direct labour cost in Rs per hour
tt = Tc+nmt // machining and non-machining time in min.
ft = f*tt // fatigue time in min.
pnt = pn*tt // personal needs in min.
t = (Tc*Tk)/(T*60) // total sharpening time in min. per forging
mct = (ts*ct)/60 // measuring and checking time in min.per forging
su = Tc + nmt+ pnt + ft + t + mct // sum of times in min.
tf = su*forgings // time for 40 forgings in min.
tst = st*setup // total set up time in min.
Te = tf+tst // total estimated time in min.
Ta = Te*R // total actual time in min.
lc = (Ta*dlc)/60 // direct labour cost in Rs
printf("\n Direct labour cost = Rs %0.1f" , lc)
```

Scilab code Exa 4.11 To find machining time

```
clc
// from figure 4.4
v = 100 // cutting speed in m/min
D = 50 // mm
```

```
l1 = 76 // mm
f = 7.5 // feed in mm/rev.
// Case 1 , time to turn 38 mm diameter by 76 mm length of cut
N1 = (1000*v)/(%pi*D)// r.p.m
tm1 = l1*10/(f*N1) // min.
// Case 2 , time to turn 25 mm by 38 mm length
N2 = (1000*v)/(%pi*38) // r.p.m
l2 = 38 // mm
tm2 = l2*10/(f*N2) // min
tt = tm1 + tm2 // total time in min
printf("\n Total time = %0.2f min." , tt)
```

Scilab code Exa 4.12 To find time to turn relief

```
clc
// from figure 4.5
v = 60 //  cutting speed m/min.
f = 0.375 // feed in mm/rev
D = 38 // mm
N = (1000*60)/(%pi*D) // rev/min
l = 32 // mm
Tm = l/(f*N) // min
printf("\n Time to turn external relief = %0.2f min." , Tm )
```

Scilab code Exa 4.13 calculate time to face on lathe

```
clc
// from figure 4.11
l = 7.5 // cm
Dave = (25+ 10)/2 // average diameter in cm
v = 27 // cutting speed in m/min
```

```
f = 0.8 // feed in mm/rev
N = (1000*v)/(%pi*Dave*10) // r.p.m.
tm = l*10/(f*N) // min.
printf("\n The machining time to face on lathe = %0
    .2f min." , tm)
```

Scilab code Exa 4.14 To find time to drill hole

```
clc
D = 12.7 // diameter in mm
d = 50 // depth in mm
v = 75 // cutting speed in m/min.
f = 0.175 // feed in mm/rev
l = d + 2*0.29*D // length of drill travel in mm
N = (1000*v)/(%pi*D) // r.p.m.
tm = l/(f*N) // min
printf("\n Time taken to drill hole = %0.3f min." ,
    tm)
```

Scilab code Exa 4.15 To find time to complete cut

```
clc
k = 1/4 // return time to cutting ratio
l = 900 + 2*75 // length of stroke in mm
v = 6 // cutting stroke in m/min
f = 2 // feed mm/stroke
w = 600 // breadth in mm
N = (v*1000)/(l*1.25) // r.p.m
N = round(N)
time =  w/(f*N) // min
printf("\n Time required for shaper to complete one
    cut = %d min" ,time )
```

Scilab code Exa 4.16 To find time to broach

```
clc
l = 70 // length of stroke in cm
cs = 11 // cutting speed in m/min
rs = 24 // return speed in m/min
tm = (l/(100*cs)) + (l/(100*rs)) // min
printf("\n Time taken to broach a four spline brass
    = %0.4f min" , tm)
// Answers vary due to round off error
```

Scilab code Exa 4.17 find feed cutter travel and time

```
clc
v = 50 // cutting speed in m/min
D = 150 // diameter of face cutter in mm
N = (1000*v)/(%pi*D) // r.p.m.
f = 0.25 // feed mm/tooth
n = 10 //number of tooth
tf = N*f*n // table feed in mm/min
l = 200 // length of work piece in mm
d = 25 // depth of slot in mm
tot = sqrt(D*d - d^2) // total overtravel in mm
tct = l + tot // total cutter travel in mm
time = tct/tf // min.
printf("\n Table feed = %d mm/min. \n Total cutter
    travel = %0.1f mm\n Time required to machine the
    slot = %0.3f min." , tf , tct ,time )
```

Scilab code Exa 4.18 To find cutting time

```
clc
D = 63.5 // diameter of plain milling cutter in mm
w = 30 // width of block in mm
l = 180 // length of block in mm
f = 0.125 // feed in mm/tooth
n = 6 // no. of teeth
N = 1500 // spindle speed in r.p.m
tot = (D - sqrt(D^2 - w^2))/2 // total over travel
    in mm
tct = l + tot // total cutter travel in mm
Tm = tct/(f*n*N) // cutting time in min
printf(" Cutting time = %0.3f min." , Tm)
// Answers vary due to round off error
```

Scilab code Exa 4.19 To find milling time

```
clc
// from figure 4.17
d = 19 // depth of cut in mm
D1 = 5 // diameter of round bar in cm
v = 50 // cutting speed in m/min
n = 8 // number of teeth
f = 0.2 // feed in mm/tooth
l = 2*sqrt(d*D1*10 - d^2) // length of chord in mm
D2 = 10 // daimeter of cutter in cm
overrun = sqrt(D2*10*d+D1*10*d-d^2) - sqrt(D1*10*d-d
    ^2)// mm
tt = l + overrun // table travel in mm
N = (1000*v)/(%pi*D2*10) // r.p.m
tm = tt/(f*n*N) // time in min.
printf("\n The milling time = %0.2f min." , tm )
```

Scilab code Exa 4.20 To find time to grind shaft

```
clc
w = 50 // width of grinding wheep in mm
f = w/2 // feed in mm
t = 0.25 // toatal stock in mm
d = 0.025 // depth of cut in mm
n = t/d // number of cuts
v = 15 // cutting speed in m/min
D = 38 // diameter in mm
N = (1000*v)/(%pi*D) // r.p.m.
l = 200 // length of part in mm
Tm = (l*10)/(f*N) // min.
printf("Time required to grind the shaft = %0.2f min
    ." , Tm)
```

Scilab code Exa 4.21 To find time to cut threads

```
clc
v = 6 // cutting speed in m/min
n = 5 // number of cuts
D = 44 // diameter in mm
N = (1000*v)/(%pi*D) // r.p.m
f = 0.5 // feed in cm
l = 8.9 // length of cut in cm
Tm = (l*n)/(f*N) // time in min
printf("\n Time to cut the threads = %0.2f min" , Tm
    )
// Answers vary due to round off error
```

Scilab code Exa 4.22 find time to produce one piece

```
clc
vt = 40 // cutting speed for turning in m/min
vs = 8 // cutting speed for cutting and knurling in
    m/min
ft = 0.4 // feed for turning in mm/rev.
ff = 0.2 // feed for forming in mm/rev
d1 = 25 // diameter in mm
l1 = 50 // mm
N1 = 1000*vt/(%pi*d1) // spindle speed in rev./min.
time1 = l1/(ft*N1) // min.
tt = 2*time1 // total time in min.
d2 = 15 // mm
N2 = 1000*vt/(%pi*d2)// rev/min.
l2 = 30 // mm
time2 = l2/(ft*N2) // min.
eft = 0.15 // end forming time in min.
d3 = 10 // mm
N3 = 1000*vs/(%pi*d3) // rev./min.
l3 = 15 // mm
f = 1.5 // feed in min.
time3 = l3/(f*N3) // min.
N4 = 1000*vs/(%pi*d1) // rev./min.
l4 = 10 // mm
time4 = l4/(ft*N4) // min.
time5 = 0.15 // time for chamfering in min.
Dave = d1/2 // mm
N5 = 1000*vt/(%pi*Dave) // r.p.m.
time6 = Dave/(N5*ff) // min,
tmt = tt+time2+time3+time4+time5+time6+eft // total
    machining time in min.
t = 0.05 // min.
ht = time5+6*time6+4*t+3*t // handling time in min.
tot = ht+tmt // total handling time in min.
ct = 15*tot/100 // contingency in min.
tct = tot+ct // total cycle time in min.
st = 60 // set up time for turret lathe
```

```
p = 100 // total pieces
stp = st/p // set up time per piece in min.
tpt = tct+stp // Total production timr per piece in min.
printf("\n Total production timr per piece = %0.2f min" , tpt)
//Answers vary due to round off error
```

Chapter 5

Economics of tooling

Scilab code Exa 5.1 To find value of machine tool

```
clc
Co = 250000 // original value of machine tool in Rs
Cs = 25000 // salvage value in Rs
n = 20 //useful life in years
d = (Co-Cs)/n // depreciation per year in Rs
v1 = Co - 10*d // value of machine tool at the end
    of 10 years in Rs
s = Co - Cs // sum at the end of useful life in Rs
i = 8/100 // annual interst rate
D = (s*i)/((1 + i)^n-1) // annual deposit
a  = D*((1+i)^10-1)/i //amount at the end of 10
    years in Rs
v2 = Co - a // value at the end of 10 years
printf("\n Value of machine at the end of 10 years
    through straight line depreciation method = Rs %d
    " , v1)
printf("\n Value of machine at the end of 10 years
    through sinking fund method = Rs %d" , v2)
// Answers vary due to round off error
```

Scilab code Exa 5.2 To find annual investment

```
clc
p = 200000 // present worth in Rs
i = 10 // annual interest rate
i = 10/100
n = 20 // number of years
a1 = (p*i)/((1+i)^n-1) // annual investment using sinking fund factor in Rs
a2 = (p*i*(i+1)^n)/((i+1)^n-1)// annual investment using capital recovery factor in Rs
printf("\nAnnual investment using sinking fund factor = Rs %d /- per year" , a1)
printf("\nAnnual investment using capital recovery factor = Rs %d /- per year" , a2)
// Answers vary due to round off error
```

Scilab code Exa 5.3 find project is economical or not

```
clc
// cash in flows
a = 21240 // annual revenue in Rs
i = 10 // annual interest rate
i = 10/100
n = 5 // perod in years
f1 = 8000 // salvage value in Rs
p1 =  (a*((i+1)^n-1))/(i*(i+1)^5)// annual revenue in Rs
p2 = f1/(i+1)^5 //present worth in Rs
t1 = p1 + p2 // total cash in flows in Rs
// cash out flows
I = 40000 // investment in Rs
```

```
f2 = 12000 // annual payment in Rs
p3 = (f2*((1+i)^5-1))/(i*(1+i)^5) // annual payments in Rs
t2 = I + p3 // total cash out flows in Rs
printf("\nTotal cash in flows = Rs %0.2f\nTotal cash out flows = Rs %0.2f",t1 ,t2)
disp("Since cash out flows are more than cash in flows therefore project is not economical")
// Answers vary due to round off error
```

Scilab code Exa 5.4 selection of economical machine

```
clc
//Machine A
f1 = 2000 // annual benefit from better production quality in Rs
i = 10 // interest rate
i = 10/100
f2 = 12000 // salvage value in Rs
f3 = 8000 // operating and maintenance cost in Rs
I1 = 100000 // initial cost in Rs
n = 5 // years
p1 = (f1*((1+i)^n-1))/(i*(i+1)^n)
p2 = f2/(1+i)^n
c1 = p1 + p2 // cash in flows in Rs
p3 = (f3*((1+i)^n-1))/(i*(i+1)^n)
c2 = I1 + p3 // cash out flows in Rs
Pa = c1 - c2 // net P.W. in Rs
//Machine B
I2 = 60000 // initial cost in Rs
f4 = 16000 // operating and maintenance cost in Rs
f5 = 14000 // reconditioning at the end of third year in Rs
p4 = (16000*((1+i)^5-1))/(i*(1+i)^5)
p5 = f5/(1+i)^5
```

```
Pb = -I2 - p4 - p5 // net P.W. in Rs
printf("\n Net P.W. of Machine A= Rs %0.2f\n Net P.W.
    of Machine B = Rs%0.2f" , Pa ,Pb)
disp("It is clear that Net P.W of Machine A is less
    nagative as compared to that of Machine B ,
    therefore Machine A is economcal.")
// Answers vary due to round off error
```

Scilab code Exa 5.5 selection of machine

```
clc
//machine A
c1 = 20000 // manual cost in Rs
c2 = 40000 // operating cost in Rs
n1 = 2 // machine life in years
i = 10 // interest rate
i = 10/100
crf1 = ((1+i)^n1-1)/(i*(i+1)^n1) // capital recovery
     factor
pw1 = c1+c2*crf1 // present worth in Rs
// machine B
c3 = 50000 // manual cost in Rs
c4 = 30000 // operating cost in Rs
n2 = 4 // machine life in years
i = 10/100 // interest rate
crf2 = ((1+i)^n2-1)/(i*(i+1)^n2) // capital recovery
     factor
pw2 = c3+c4*crf2 // present worth in Rs for 4 years
pw3 = (pw2*crf1)/crf2 // present worth in Rs for 2
    years
printf("\n P.W. of expenses for A = Rs %d\n P.W. of
    expenses for B = Rs %0.2f" ,pw1,pw3)
disp("As the expenses of machine B are less , so
    this is economical")
// Answers vary due to round off error
```

Scilab code Exa 5.6 selection of economical machine

```
clc
//Machine A
i = 8 // // interest rate
i = i/100 // interest rate
n1 = 10 // economic life in years
CRF1 = i*(1+0.08)^n1/((1+i)^n1-1) // capital recovery factor
p1 = 46000 // first cost in Rs
s1 = 8000 // salvage value in Rs
o1 = 10000 // operating charges in Rs
AC1 = (p1-s1)*CRF1 + s1*i + o1 // annual cost in Rs
//Machine B
n2 = 15 // economic life in years
CRF2 = i*(1+0.08)^n2/((1+i)^n2-1) // capital recovery factor
p2 = 60000 // first cost in Rs
s2 = 10000 // salvage value in Rs
o2 = 9200 // operating charges in Rs
AC2 = (p2-s2)*CRF2 + s2*i + o2 // annual cost in Rs
printf("\n Annual cost of machine A = Rs %0.2f\n Annual cost of Machine B = Rs %0.2f" ,AC1, AC2 )
disp("Machine B will be economical")
// Error in textbook
```

Scilab code Exa 5.7 find ERR and economicality of project

```
clc
a = 100000 // Ej(p/f,e%,j) in Rs
n = 5 // life in years
```

```
e = 20 // M.A.R.R.
e = e/100 // M.A.R.R.
i = e
A = 32000 // savings in Rs
s = 20000 // salvage value in Rs
b = ((A*(((i+1)^n)-1)/i)+s)/a // (F/p,I,5)
i2 = (b)^(1/n)-1 // internal rate of return
printf("\n ERR = %0.4f\n Internal rate of return =
    %0.2f percent" , b , i2*100)
disp("Since Internal rate of return is > M.A.R.R ,
    therefore project is feasible")
```

Scilab code Exa 5.9 find ERR and economicality of project

```
clc
e = 20 // M,A.R.R.
i = e // interest rste
i = i/100
n = 5 // life in years
s = 32000 // annual net savings in Rs
p = 100000 // present worth in Rs
S = 20000 // salvage value in Rs
a = (p-S)*(i/((1+i)^n-1)) // (p-s)(A/F,e%,n)
E = (s-a)/p // E.R.R.R
printf("\n ERRR = %0.2f percent", E*100)
disp("Since E.R.R.R is > M.A.R.R. therefore project
    is feasible.")
```

Scilab code Exa 5.10 To determine acceptance of machine

```
clc
// machine A
r_e1 = 9600 //cash flow in Rs
```

```
p1 = 46000 // intial cost in Rs
s = 0 // salvage value
e = 8 // M.A.R.R
e = e/100
i = 8 // investment rate
i = i/100
n = 6 // life in years
x = i/((1+i)^n-1)
ERRR1 = (r_e1 - (p1-s)*x)/p1
//machine B
r_e2 = 7200 //cash flow in Rs
p2 = 32000 // intial cost in Rs
ERRR2 = (r_e2 - (p2-s)*x)/p2
printf("\n ERRR1 = %0.2f percent \n ERRR2 = %0.2f
   percent" , ERRR1*100 ,ERRR2*100)
disp("Only machine B is accepteble")
```

Scilab code Exa 5.11 find investment cost and unamortized value

```
clc
pmv = 15000 // present market value in Rs
ss = 6000 // sum needed to make it serviceable in Rs
ic = ss + pmv //  investment cost in Rs
pbv = 30000 // present book value in Rs
sv = 15000 // salvage value in Rs
ui = pbv - sv // unamortized investment in Rs
printf("\n Investment cost = Rs %d\n Unamortized
   investment = Rs %d" , ic , ui)
```

Scilab code Exa 5.13 To make decision of machines replacement

```
clc
// Existing machine
```

```
pmp = 100000 // present market price in Rs
io = 50000 // immediate overhauling in Rs
asl = 5 // additional service life in years
aoc = 50000 // annual operating cost in Rs
svo = 10000 // salvage value after overhauling in Rs
pc = io + pmp // present cost in Rs
i = 10 // interest rate
i = 10/100
crf1 = (i*(1+i)^asl)/((1+i)^asl - 1) // capital
    recovery factor
AC1 = (pc - svo)*crf1 + svo*i + aoc // average cost
    in Rs
// proposed machine
n = 10 // expected economic life in years
ic = 300000 // initial cost in Rs
sv = 100000 // salvage value in Rs
o = 30000 // annual operating cost in Rs
crf2 = (i*(1+i)^10)/((1+i)^10 - 1)
AC2 = (ic - sv)*crf2 + sv*i + o // average cost in
    Rs
printf("Existing machine = Rs %0.3f \n Proposed
    machine = Rs %0.2f" , AC1 , AC2)
disp("Since the equivalent annual cost of proposed
    machine is less than that of the existing machine
    , therefore, the replacement is justified.")
// Answers vary due to round off error
```

Scilab code Exa 5.15 Determine economic repair life

```
clc
c = 20000 // first cost of machine in Rs
s = 1000 // scrap value in machine in Rs
b = 180 // annual increase in cost of repairs in Rs
n = sqrt(2*(c-s)/b) // years
printf("\n Number of years of economic repair life =
```

```
%0.2f years" , n)
```

Scilab code Exa 5.16 find time to pay for itself

```
clc
Cn = 72000 // cost of new machine installed and
    tooled in Rs
Co = 28000 // cost of new machime installed and
    tooled in Rs
p = 16 // hourly pieces
Nn = 2200*p // estimated annual production on new
    machine
Ko = 17200 // present book value of old machine in
    Rs
So = 6400 // scrap value of old machine in Rs
Sn = 8000 // probable scrap value of old machine in
    at the end of its useful life Rs
oco = 2.5 // opreator cost per hour
mco= 48 // machine cost
ro = 10 // production rate per hour
ocn = 2 // opreator cost per hour
mcn= 62 // machine cost
rn = 16 // production rate per hour
Po = (oco+mco)/ro // labour and machine cost per
    unit on old machine in Rs
Pn = (ocn+mcn)/rn // labour and machine cost per
    unit on new machine in Rs
i = 6 // interest on investment
i = i/100
t = 6 // annual taxes
t = t/100
d = 10 // annual allowance for depreciation
d = d/100
m = 3 // annual allowance for maintenance
m = m/100
```

```
n = ((Cn-Sn)+(Ko-So))/((Nn*(Po-Pn)) - Cn*(i+t+d+m))
printf("\n The number of years in which the new
    machine will pay for itself = %0.3f years" , n)
```

Scilab code Exa 5.17 selection of machine for job

```
clc
C = 80000 // cost of new machine installed and
    tooled in Rs
nel = 2 // number of engine lathes
c = 32000*nel // first cost of engine lathe
N = 4000 // annual production of turret lathe
n = 3800 // annual production in engine lathe
nhp1 = 4 // hp motor
L = 2256*nhp1 // annual labour cost of turret lathe
w = 5 // wage in per hour
time = 2300 // hours
l = time*nel*w // labour cost of engine lathe
nhp2 = 2.5 // hp motor
pr = 0.35 // power rate in kwh
p = (nel*nhp2*746*time*pr)/1000 // power cost
P = (nhp1*746*time*pr)/1000 // power cost
F = 480 //saving
I  = 6/100 // interest rate
T = 4/100 // tax rate
D = 10/100 // allowance for depreciation in engine
    lathe
M = 6/100 // allowance for maintenance in engine
    lathe
B = 55/100 // labour burden in engine lathe
i  = 6/100 // interest rate
t = 4/100 // tax rate
d = 10/100 // allowance for depreciation in turret
    lathe
m = 6/100 // allowance for maintenance in turret
```

```
    lathe
X = (L + B*L + P +C*(I+T+D+M) - F)/N
x = (l+l*B + p + c*(i+t+d+m))/n
printf("\n Unit production cost on turret lathe = Rs
    %0.2f per piece\n Unit production cost on engine
    lathe = Rs %0.2f per piece" , X , x)
disp("Turret lathe will be more economical than two
    engine lathe")
// Answers vary due to round off error
```

Scilab code Exa 5.18 Calculate maximum investment on turret lathe

```
clc
X = 9.16 // production cost on turret lathe
N = 4000 // annual requirement
c = X*N // cost for 4000 pieces on turret lathe
n = 3800 // production of engine lathe
l = 23000 // labour cost
p = 3002 // power cost
i = 6 // interest rate
i = i/100
t = 4 // tax rate
t = t/100
d = 10 // allowance for depreciation in turret lathe
d = d/100
m = 6 // allowance for maintenance in turret lathe
m = m/100
b = 55/100 //labour burden
a = i+t+d+m
tc = 64000 // first cost of engine lathe
c1 =(N*(l*(1+b)+p))/n+(tc*a) // cost for engine
    lathe
s = c1-c // savings
amt = s/a // amount invested in turret lathe over
    the cost of engine lathe
```

```
printf("\n Amount invested in turret lathe over the
    cost of engine lathe = Rs %d" , amt)
// Answers vary due to round off error
```

Scilab code Exa 5.19 To find years for new machine

```
clc
Cn = 60000 // cost of new machine
Sn = 5000 // scrap value of new machine
So = 1000 // scrap value of old machine
Nn = 200000 //annual production
I = 10 // interest rate
I = I/100
M = 7 // allowance for maintenane
M = M/100
T = 6 // annual taxes
T = T/100
D = 1/10 // allowance for depreciation
lco = 300 // labour charges for old machine
m = 12 // months
rco = 15000 // running charges for old machine
pro = 50000 // production rate for old machine
lcn = 500 // labour charges for new machine
rcn = 10000 // running charges for old machine
prn = 200000// production rate f
Po = (lco*m + rco)/pro // labour and machine cost on
     old machine
Pn = (lcn*m + rcn)/prn // labour and machine cost on
     new machine
n =((Cn-Sn)-So)/((Nn*(Po-Pn))-Cn*(I+T+D+M)) //years
printf("\n Years in which new machine will pay for
    itself = %0.2f years" , n)
```

Scilab code Exa 5.20 To find cost and pieces

```
clc
a = 1.50 //saving in labour
b=55/100 // burden applied on labour
T = 4/100 // allowance for taxes
M = 5/100 // allowance for maintenance
I = 8/100 // interest rate
D = 50/100 // allowance for depreciation
H = 2 // years to amortize the investment
S = 50 // yearly cost for set up
C = 3000 // first cost
N1 = (C*(I+T+M+D)+S)/(a*(1+b)) // annual production when 1 run is made
r = 5 // number of runs
N2 = (C*(I+T+M+D)+S*r)/(a*(1+b)) // annual production when 1 run is made
D1 = 100/100 // allowance for depreciation
N3 = (C*(I+T+M+D1)+S)/(a*(1+b)) // production when D = 100
n1 = 1530 // pieces
C1 = (n1*(a*(1+b))-S)/(I+T+M+D1) // economical investment
n2 = 950 // pieces
a1 = 2 // labour cost
r1 = 6 // number of runs
S1 = r1*S // yearly cost
V = n2*a1*(1+b)-C*(I+T+M+D)-S1 //profit
printf("\n Number of pieces when one run is made and cost is Rs 3000 = %d pieces",N1)
printf("\n Annual production when 5 runs are made per year = %d pieces" , N2)
printf("\nAnnual production when fixture pay for itself = %d pieces" , N3)
printf("\nEconomical investment when 1530 pieces for single run with savings Rs 1.50 per piece = Rs %d",C1)
printf("\nAnnual profit when 950 pieces made per
```

```
    year in 6 runs and saving in labour cost Rs 2 per
     piece = Rs %d per year" , V)
//  'Answers vary due to round off error'
```

Scilab code Exa 5.21 To find number of components

```
clc
a = 0.125 //saving in labour cost per unit
b = 0.4 // overhead applied on direct labour saved
D = 1/2 // allowance for depreciation
C = 2400 // first cost
I = 6/100 // interest rate
T = 4/100 // allowance for taxes
M = 10/100 // allowance for maintenance
S = 80 // cost of set up
N = (C*(I+T+D+M)+S)/(a*(1+b)) // pieces per year
t = N*2 // total number of pieces
printf("\n Total number of pieces produced = %d" , t
    )
// Answers vary due to round off error
```

Scilab code Exa 5.22 To find number of components

```
clc
a = 0.125 // saving in labour cost per unit
b = 0.4 // overhead applied on direct labour saved
D = 1/2 // allowance for depreciation
C = 2400 // first cost
I = 6/100 // interst rate
T = 4/100 // allowance for taxes
M = 10/100 // allowance for maintenance
n = 6 // number of baches
S = 80 // cost of set up
```

```
s1 = S*n // total set up cost
N = (C*(I+T+D+M)+s1)/(a*(1+b)) // pieces
t = N*2 // total number of pieces
printf("\n Total number of pieces produced = %d" , t
    )
// Answers vary due to round off error
```

Scilab code Exa 5.23 To find time and profit

```
clc
C1 = 2000 // first cost  small tool in Rs
N = 5000 // parts per year
n = 5 // number of batches
S = 50*n // cost of set up
a = 0.15 // saving in labour cost per unit
b = 50/100 // burden applied on direct labour saved
I = 10/100 // interest rate
T = 5/100 // allowance for tax
M = 10/100 // allowance for maintenance
H = C1/((N*a*(1+b))-(C1*(I+T+M))-S) // years
C2 = 1600 // cost of fixture
D = 1/H // allowance for depreciation
V = N*a*(1+b)-C2*(I+T+D+M)-S // profit
printf("\n Number of years taken by fixture of Rs
    2000 = %0.2f years\n profit made when fixture of
    Rs 1600 = Rs %d" , H ,V)
```

Scilab code Exa 5.24 To find minimum number of components

```
clc
c1 = 3 //  machine cost per component using existing
     euipment in Rs
c2 = 1 // machine cost using fixture in Rs
```

```
s = c1 - c2 // saving in machine cost per piece
f= 1000 // cost of fixture in Rs
N = f/2 // components
printf ("\n Minimum number of components to be
    produced if cost of fixture to be recovered = %d"
    ,N)
```

Scilab code Exa 5.25 To calculate number of pieces

```
clc
C = 1000 // cost of fixture
Co = 700 // cost of old fixture
Cs = 250 // scrap value
a = 10 //saving per piece in paisa
a = a/100
b = 30 // overhead applied on direct labour saved
b = b/100
I = 8 // interest rate
I = I/100
M = 3 // allowance for maintenance
M = M/100
T = 12 // allowance for tax
T = T/100
H = 3/2 // amortization
D = 1/H // allowance for depreciation
N = (C*(I+T+D+M)+(Co-Cs)*I)/(a*(1+b)) // pieces per
    year
printf("\n Number of pieces which must be produced
    to break even so that fixture may pay for itself
    = %d pieces per year" , N)
// Answers vary due to round off error
```

Scilab code Exa 5.26 To find cost for new fixture

```
clc
N = 9000 // number of pieces
Co = 700 // cost of old fixture
Cs = 250 // scrap value
a = 10 //saving per piece in paisa
a = a/100
b = 30 // overhead applied on direct labour saved
b = b/100
I = 8 // interest rate
I = I/100
M = 3 // allowance for maintenance
M = M/100
T = 12 // allowance for tax
T = T/100
H = 3/2 // amortization
D = 1/H // allowance for depreciation
C = (N*a*(1+b)-(Co-Cs)*I)/(I+T+D+M) // cost in Rs
printf("\n Cost for new fixture = Rs %d" , C)
// Answers vary due to round off error
```

Scilab code Exa 5.27 find time to amortize fixture

```
clc
n = 6500 // yearly production
c = 1350 // cost of fixture
a = 10 //saving per piece in paisa
a = a/100
b = 30 // overhead applied on direct labour saved
b = b/100
I = 8 // interest rate
I = I/100
M = 3 // allowance for maintenance
M = M/100
T = 12 // allowance for tax
T = T/100
```

```
co = 700 // cost of old fixture
cs = 250 // scrap value
H = (c)/((n*a*(1+b))-I*(co-cs)-c*(I+T+M)) // amotization in years
printf("\n Time taken to amortize the fixture = %0.1f years" , H)
```

Scilab code Exa 5.28 To find profit

```
clc
n = 9000 // production of pieces per year
c = 1000 // fixture costs
Co = 700 // cost of old fixture
Cs = 250 // scrap value
a = 10 //saving per piece in paisa
a = a/100
b = 30 // overhead applied on direct labour saved
b = b/100
I = 8 // interest rate
I = I/100
M = 3 // allowance for maintenance
M = M/100
T = 12 // allowance for tax
T = T/100
h = 1.5 // amortization
D = 1/h // allowance for depreciation
V = (n*a*(1+b))-(c*(I+T+D+M))-((Co-Cs)*I) // profit
printf("\n profit = Rs %d " , V)
// Answers vary due to round off error
```

Scilab code Exa 5.29 To find BEP Cost and Components

```
clc
```

```
fc1 = 100000 // fixed cost in Rs
vc1 = 100 // variable cost in Rs per unit
sp = 200 // selling price in Rs per unit
q1 = fc1/(sp-vc1) // quantity of production at break
    even point
fc2 = 125000 // fixed cost in Rs
vc2 = 90 // variable cost in Rs per unit
q2 = fc2/(sp-vc2) // quantity of production at break
    even point
p = 20000 // profit in Rs
q3 = (fc1 + p)/(sp-vc1) // quantity of production at
    profit of Rs 20000
printf("\n Break even point = %d pieces \n If fixed
    cost is 125000 and variable cost is Rs 90 per
    unit then break even point = %d pieces\n Number
    of components to get profit of Rs 20000 = %d
    pieces" , q1 , q2 ,q3)
```

Scilab code Exa 5.30 To find break even point

```
clc
fc1 = 12000 // fixed cost for machine A in Rs
fc2 = 48000 // fixed cost for machine B in Rs
n1 = 6 // unit production cost in Rs per piece for
    machine A
n2 = 1.2 // unit production cost in Rs per piece for
    machine B
q = (fc2-fc1)/(n1-n2) // break even point
printf("\n Break even point = %d pieces" , q)
```

Scilab code Exa 5.31 To find break even quantity

```
clc
```

```
// capstan lathe
tc1 = 300 // total cost in Rs
mc1 = 2.5 // material cost per piece in Rs
olc1 = 5 // operation labour cost per hour in Rs
ct1 = 5 // cycle time per piece in min.
slc1 = 20 // setting up labour cost in Rs per hour
st1 = 1 // setting up time in hour
mo1 = 300/100 // machine over heads of operation
    labour cost
o1 = mo1*olc1 // overheads of capstan lathe in Rs
    per hour
fc1 = tc1 + slc1*st1 + o1*st1 // fixed cost of
    capstan lathe in Rs
vc1 = mc1 + (olc1*ct1)/60 + (o1*ct1)/60 // variable
    cost in Rs
// Automatic (single spindle)
tc2 = 300 // total cost in Rs
cc2 = 1500 // cost of cams in Rs
mc2 = 2.5 // material cost per piece in Rs
olc2 = 2 // operation labour cost per hour in Rs
ct2 = 1 // cycle time per piece in min.
slc2 = 20 // setting up labour cost in Rs per hour
st2 = 8 // setting up time in hour
mo2 = 1000/100 // machine over heads of operation
    labour cost
o2 = mo2*olc2 // overheads of single spindle in Rs
    per hour
fc2 = tc2 + cc2 + slc2*st2 + o2*st2 // fixed cost of
     single spindle in Rs
vc2 = mc2 + (olc2*ct2)/60 + (slc2)/60 // variable
    cost in Rs
q = (fc2-fc1)/(vc1-vc2) // break even quantity
printf("\n Break even quantity for a component which
     can be produced on either the capstan lathe or
    single spindle automatic = %d pieces" , q)
```

Scilab code Exa 5.32 To do break even analysis

```
clc
// Engine lathe
t = 12 // time/piece in min.
l = 7 // overhead cost/hr
o = 4 // direct labour cost/hr
s = 2 // set up time in hour
sr = 8 // set up rate per
// turret lathe
T = 5 // time/piece in min.
L = 5 // overhead cost/hr
O = 8 // direct labour cost/hr
S = 8 // set up time in hour
SR = 8 // set up rate per
q = 60*(S*SR-s*sr)/(t*(l+o)-T*(L+O)) // break even
    point
q = round(q)
printf("\n Break even point = %d pieces" , q)
```

Scilab code Exa 5.33 To calculate minimum number of pieces

```
clc
fc1 = 80000 // fixed cost for turret lathe in Rs
fc2 = 32000 // fixed cost for engine lathe in Rs
n1 = 16 // production of pieces per year in turret
    lathe
n2 = 10 // production of pieces per year in engine
    lathe
vc1 = 2 //operators cost in turret lathe
vc2 = 2.5 // operators cost in engine lathe
Q=poly(0,'Q')
```

```
Q=roots((fc1+1/n1*vc1*Q)-(fc2+2.5*Q/10))
printf("\n Break even point = %d pieces" , Q)
```

Scilab code Exa 5.34 To determine the point

```
clc
st1 = 15 // set up time for engine lathe in min.
ut1 = 15 // unit time for engine lathe in min.
st2 = 90 // set up time for automatic lathe in min.
ut2 = 1.5 // unit time for engine lathe in min.
q = (st2-st1)/(ut1-ut2) // quantity of production
printf("\n The point at which the automatic lathe will be justified = %0.2f " , q)
// Answers vary due to round off error
```

Scilab code Exa 5.35 To find quantity of pieces

```
clc
// Automatic lathe
p = 30 // number of pieces produced per hour
l = 4 // labour rate per hour in Rs
d = 4.50 // hourly depreciation rate per machine in hour
s = 4 // set up time in hour
// turret lathe
P = 10 // number of pieces produced per hour
L = 4 // labour rate per hour in Rs
D = 1.50 // hourly depreciation rate per machine in hour
S = 2 // set up time in hour
q = (P*p*(S*L+S*D-s*l-s*d))/(P*(l+d)-p*(L+D)) // quantity of pieces at break even point
```

```
printf("\n Quantity of pieces at Break even point =
   %d pieces" , q)
```

Scilab code Exa 5.36 To determine quantity of production

```
clc
Pa = 8.4 // unit tool process cost for method A in
   Rs
Pb = 14.8 // unit tool process cost for method B in
   Rs
Ta = 6480 //total tool cost for method A in Rs
Tb = 1616 //total tool  cost for method B in Rs
q = (Ta-Tb)/(Pb-Pa) // break even point
printf("\n Quantity of production at break even
   point = %d pieces" , q)
```

Scilab code Exa 5.37 find preference between machines and production

```
clc
// machine A
ic1 = 50000 // initial cost
hoc1 = 10 // hourly operating charges
pp1= 5 // pieces produced per hour
i = 15 // interest rate
i = i/100
oh = 2000 // operating hours
fc1 = ic1*i // fixed cost
vc1 = oh*hoc1 // variable cost
tc1 = fc1+vc1 // total charges
ao1 = oh*pp1 // annual output
c1 = tc1/ao1 // cost per unit
// machine B
ic2 = 80000 // initial cost
```

```
hoc2 = 8 // hourly operating charges
pp2= 8 // pieces produced per hour
fc2 = ic2*i // fixed cost
vc2 = oh*hoc2 // variable cost
tc2 = fc2+vc2 // total charges
ao2 = oh*pp2 // annual output
c2 = tc2/ao2 // cost per unit
printf("\n (i) Cost per unit for machine A = Rs %0.2
    f\n Cost per unit machine B = Rs %0.2f",c1,c2)
disp("machine B will be preferred")
// machine A
ao3 = 4000 // annual output
oc3 = ao3*hoc1/pp1 // operating charges
tc3 = oc3+fc1 // total annual charge
c3 = tc3/ao3 // cost/piece
// machine B
ao4 = 4000 // annual output
oc4 = ao4*hoc2/pp2 // operating charges
tc4 = oc4+fc2 // total annual charge
c4 = tc4/ao4// cost/piece
printf("\n (ii) Cost per unit for machine A = Rs %0
    .2f\n Cost per unit machine B = Rs %0.2f",c3,c4)
disp("machine A will be preferred")
A = hoc1/pp1 // operating cost per piece on machine
    A
B = hoc2/pp2 // operating cost per piece on machine
    B
Q = fc2 - fc1 // annual production
printf("\n(iii) Annual production to make cost per
    piece equal for two machines = %d pieces" , Q )
```

Scilab code Exa 5.38 To find BEP and various sales

```
clc
as = 80000 // annual sales in Rs
```

```
vc = 64000 // variable expenses in Rs
c = 16000 // contribution in Rs
fc = 24000 // fixed expenses in Rs
l = 8000 // losses in Rs
p = 9000 // profit in Rs
s1 = fc + vc // sales at B.E.P in Rs
s2 = (fc + vc + p)/0.945 // sales at net income of
    Rs9000 and corporate tax rate being 5.5%
q = 10000 // quantity of units
sp = (fc+vc)/q // selling price per unit in Rs
printf("\n Sales at break even point = %d units" ,
    s1 )
printf("\n Sales at net income of Rs9000 and
    corporate tax rate being 5.5 = Rs %0.2f\n Sales
    per unit if B.E.P brought down to 10000 units =
    Rs %0.2f per unit" ,s2 , sp)
```

Scilab code Exa 5.39 To determine break even point

```
clc
fc = 55000 // fixed cost in Rs
vc = 45 // variable cost per piece in Rs
sp = 100 // selling price per piece in Rs
p = (vc/sp)*100 // percentage of variable cost to
pm = 100 - p // profit margin
bep = ((55000/55)*100)/100 // Break even point
printf("\n Break even point = %d pieces" , bep)
```

Scilab code Exa 5.40 To calculate economic lot size

```
clc
f1 = 335 // fixed cost in Rs for capstan lathe
k = 0.25 // stock carrying factor in paise per piece
```

```
k = k/100
N1 = sqrt(f1/k) // pieces for capstan lathe
a1 = 4.16 // variable cost per piece for capstan
    lathe
tc1 = a1+f1/N1+k*N1 // total cost for capstan lathe
f2 = 2120 // fixed cost in Rs for turret lathe
N2 = sqrt(f2/k) // pieces for turret lathe
a2 = 2.863 // variable cost per piece for turret
    lathe
tc2 = a2+f2/N2+k*N2 // total cost for turret lathe
printf("\n Total cost per piece for capstan lathe =
    Rs %0.2f\n Total cost per piece for turret lathe
    = Rs %0.2f" , tc1 , tc2)
// Answers vary due to round off error
```

Scilab code Exa 5.41 To find EOQ and total cost

```
clc
R=500 // cost of ordering in Rs per order
A=12000 //annual consumption units
C=3.00 // unit cost of item
K=3 // unit storage cost
I1=0.2 // interest rate
function y=f(N)
function G=f2(N)
G=C*A+I1*C*N/2+K*N/2+A*R/N // total cost per year
endfunction
y=derivative(f2,N)
endfunction
funcprot(0)
N=fsolve(2000,f)
O = A/N // number of orders
N1 = 2400 // units
tc = C*A + I1*C*N1/2 + K*N1/2 + A*R/N1 // total cost
     in Rs
```

```
I2 = (2*R*A)/(C*N1^2)
printf("\n Economic order quantity = %d units\n Totl
    cost = Rs %d per year\n I = %0.4f",N1,tc,I2)
disp(" It is clear that inventory cost will get
   increased very greatly")
```

Scilab code Exa 5.42 Determine optimum lot size

```
clc
A = 40000 // number of units per year
I = 25 // carrying cost in percent
I = I/100
C1 = 8 // cost for 0 < N < 1000 per unit in Rs
C2 = 7.5 // cost for 1000 < N < 10000 per unit in Rs
C3 = 7.25 // cost for N >= 10000 per unit in Rs
R = 250 // ordering cost per order in Rs
N = 10000 // units
N1 = sqrt(2*R*A/(I*C3)) // optimal quantity for
   lowest curve
G1= C3*A+(A*R)/N+I*C3*N/2 // total cost in Rs
N2 = sqrt(2*R*A/(I*C2)) // optimal quantity for
   higher curve
G2= C2*A+(A*R)/N2+I*C2*N2/2 // total cost in Rs
N3 = sqrt(2*R*A/(I*C1)) // optimal quantity for
   highest curve
G3 = C1*A+(A*R)+1 // total cost in Rs
printf("\n Total cost for lowest cost curve = Rs %0
   .2f\n Total cost for next higher curve = Rs %0.2f
   \n Total cost for highest curve = Rs %0.2f " , G1
   ,G2,G3)
disp("Comparing all total cost lowest is Rs
   300,062.50 for an order quantity of 10,000.")
disp("N = 10,000 and No. of orders = 4")
```

Scilab code Exa 5.43 To find most economical lot size

```
clc
c = 50000 // components
R=500 // cost of ordering in Rs per order
A=12000 //annual consumption units
C=3.00 // unit cost of item
K=1.50 // unit storage cost
I=0.2 // interest rate
function y=f(N)
    function G=f2(N)
        G=0.02*N+1500000/N
    endfunction
y=derivative(f2,N)
endfunction
funcprot(0)
N=fsolve(2000,f)
l = c/N // number of lots
l = ceil(l)
ls = c/l // lot size
printf("\n The lot size = %d components",ls)
```

Chapter 9

Limits Tolerences and Fits

Scilab code Exa 9.1 To find allowance and tolerence

```
clc
h1 = 37.52 // high limit of hole in mm
h2 = 37.50 // low limit of hole in mm
s1 = 37.47 // high limit of shaft in mm
s2 = 37.45 // low limit of shaft in mm
ht = h1-h2 // hole tolerence in mm
st = s1-s2 // shaft tolerence in mm
a = h2-s1 // allowance in mm
printf("\n Hole tolerence = %0.2f mm\n Shaft
    tolerence = %0.2f mm\n Allowance = %0.2f mm" ,ht
    ,st ,a)
```

Scilab code Exa 9.2 Determine dimensions of shaft and hole

```
clc
t = 0.075 // tolerence in mm
h2 = 75 // low limit of hole in mm
a = 0.10 // allowance in mm
```

```
h1 = h2+t  // high limit of hole in mm
s1 = h2-a // high limit of shaft in mm
s2 = s1-t // low limit of shaft in mm
printf("\n High limit of hole = %0.3f mm\n High
    limit of shaft = %0.2f mm\n Low limit of shaft =
    %0.3f mm" ,h1 ,s1 , s2)
```

Scilab code Exa 9.3 Determine dimensions of hole and shaft

```
clc
t = 0.225 // tolerence in mm
h2 = 75 // low limit of hole in mm
a = 0.0375 // interference in mm
h1 = h2+t  // high limit of hole in mm
s2 = h1+a // low limit of shaft in mm
s1 = s2+t // high limit of shaft in mm
printf("\n High limit of hole = %0.3f mm\n Low limit
     of shaft = %0.4f mm\n High limit of shaft = %0.4
    f mm" ,h1 ,s2 , s1)
```

Scilab code Exa 9.4 Calculate fundamental deviations and tolerences

```
clc
s1 = 50 // diameter of step1 in mm
s2 = 80 // diameter of step2 in mm
d = (s1*s2)^(1/2) // mm
i = (0.45*(d)^(1/3)+0.001*d)/10^3 // mm
t1 = 25*i // tolerence for hole in mm
t2 = 16*i // tolerence for shaft in mm
a1 = 0 // fundamental deviation for hole in mm
a2 = 5.5*(d)^0.41 // fundamental deviation for shaft
     in microns
a2 = a2/10^4 // mm
```

```
h1 = 60 // low limit of hole in mm
h2 = h1+t1 // high limit of tolerence in mm
s1 = h1 - t2 // high limit of shaft in mm
s2 = s1-t2 // low limit of shaft in mm
printf("\n Tolerence for hole = %0.3f mm\n Tolerence
     for shaft = %0.3f mm" , t1 ,t2)
printf("\n Fundamental deviation for hole = %0.2f mm
    \n Fundamental deviation for shaft = %0.3f mm" ,
    a1 , a2 )
printf("\n Low limit of hole = %d mm\n High limit of
     hole = %0.3f mm\n High limit of shaft = %0.2f mm
    \n Low limit of hole = %0.2f mm" ,h1 ,h2 , s1 ,
    s2)
// Answers vary due to round off error
```

Scilab code Exa 9.5 Find tolerences limits and clearance

```
clc
b = 30 // basic size in mm
s1 = 0.005 // maximum limit of shaft in mm
s2 = 0.018 // minimum limit of shaft in mm
h1 = 0.020 // maximum limit of hole in mm
h2 = 0.0 // minimum limit of hole in mm
t1 = s2-s1 // shaft tolerence in mm
t2 = h1-h2 // hole tolerence in mm
Sh = b-s1 // high limit of shaft in mm
Sl = b-s2 // low limit of shaft in mm
Hh = b+h1 // high limit of hole in mm
Hl = b+h2 // low limit of hole in mm
c1 = Hh-Sl // maximum clearance in mm
c2 = Hl-Sh // minimum clearance in mm
printf("\n Basic size = %d mm\n Shaft tolerence = %0
    .3f mm\n Hole tolerence = %0.3f mm",b,t1,t2)
printf("\n High limit of shaft = %0.3f mm\n Low
    limit of shaft = %0.3f mm\n High limit of hole =
```

```
    %0.3f mm \n Low limit of hole = %0.3f mm",Sh,Sl,
    Hh,Hl)
printf("\n Maximum clearance = %0.3f mm\n Minimum
    clearance = %0.3f mm",c1,c2)
```

Scilab code Exa 9.6 Determine limits of shaft and hole

```
clc
minc = 0.01 // minimum clearance in mm
bs = 25 // basic size in mm
maxc = 0.02 // maximum clearance in mm
x=poly(0,'x')
y=1.5*x
x=roots(y+0.01+x-0.02)
y=horner(y,x)
// hole basis system
low_h1 = bs // low limit of hole in mm
high_h1 = bs+y // high limit of hole in mm
u_s = low_h1-minc // upper limit of shaft in mm
low_s1 = u_s-x // lower limit of shaft in mm
// shaft basis system
high_s = bs // high limit of shaft in mm
low_s2 = bs-x // low limit of shaft in mm
low_h2 = bs+minc // low limit of hole in mm
high_h2 = low_h2+y // high limit of hole in mm
printf("Hole basis system \n Lower limit of hole =
    %d mm\n Higher limit of hole = %0.3f mm\n Higher
    limit of shaft = %0.3f mm \n Lower limit of shaft
     = %0.3f mm", low_h1,high_h1,u_s,low_s1)
printf("\n Shaft basis system \n high limit of shaft
     = %0.3f mm\n lower limit of shaft = %0.3f mm\n
    lower limit of hole = %0.3f mm\n upper limit of
    hole = %0.3f mm" , high_s,low_s2,low_h2,high_h2)
```

Scilab code Exa 9.7 Determine dimensions of shaft and hole

```
clc
bs = 100 // basic size in mm
s1 = 120// diameter of step1 in mm
s2 = 80 // diameter of step2 in mm
d = (s1*s2)^(1/2) // mm
d = ceil(d)
i = (0.45*(d)^(1/3)+0.001*d)/10^3 // mm
t1 = 16*i // tolerence for hole in mm
t2 = 25*i // tolerence for shaft in mm
G = (2.5*(d)^0.34)/10^3 // fundamental deviation for
    hole in mm
e = (11*(d)^0.11)/10^3 // fundamental deviation for
    shaft in microns
// Hole
LLh = bs+G // lower limit of hole in mm
HLh = LLh+t1 // higher limit of hole in mm
// shaft
ULs = bs-e // upper limit of shaft in mm
LLs = ULs-t2 // lower limit of shaft in mm
printf("\n lower limit of hole = %0.3f mm\n higher
    limit of hole = %0.3f mm\n upper limit of shaft =
     %0.3f mm\n lower limit of shaft = %0.3f mm" ,
    LLh,HLh,ULs,LLs)
// Error in textbook
```

Scilab code Exa 9.8 Determine size of bearing and journal

```
clc
tb = 0.005 // tolerence on bearing in mm
tj = 0.004 // tolerence on journal in mm
```

```
a = 0.002 // allowance in mm
//hole-basis system
b = 100 // basic size in mm
Bl = b // lower limit of bearing in mm
Bh = Bl+tb // higher limit of bearing in mm
Jh = Bl-a // higher limit of journal in mm
Jl1 = Jh - tj // lower limit of journal in
// shaft-basis system
Ju = b // upper limit of journal in mm
Jl2 = Ju-tj // lower limit of journal in mm
Bl = Ju+a // lower limit of bearing in mm
Bu = Bl+tb // upper limit of bearing in mm
printf("\n Hole basis system \n Lower limit of
    journal = %d mm\n Higher limit of bearing = %0.3f
     mm\n Higher limit of journal = %0.3f mm \n Lower
     limit of journal = %0.3f mm", Bl,Bh,Jh,Jl1)
printf("\n shaft basis system \n upper limit of
    journal = %0.3f mm\n lower limit of journal = %0
    .3f mm\n lower limit of bearing = %0.3f mm\n
    upper limit of bearing = %0.3f mm" , Ju,Jl2,Bl,Bu
    )
```

Scilab code Exa 9.9 Determine size of two mating parts

```
clc
// Hole-basis system
b = 100 // basic size in mm
i1 = 0.12 // maximum interference in mm
i2 = 0.05 // minimum interfernce in mm
t = (i1-i2)/2 // tolerence in mm
Sh = b+i1 // upper limit of shaft in mm
Hl = b // lower limit of hole in mm
Hh = b+t // higher limit of hole in mm
Sl1 = Sh-t //lower limit of shaft in mm
// shaft-basis system
```

```
Su = b // upper limit of shaft in mm
Sl2 = b-t // lower limit of shaft in mm
Hl1 = b-i1 // lower limit of hole in mm
Hu = Hl1+t // higher limit of hole in mm
printf("\n Hole basis system \n upper limit of shaft
    = %0.3f mm\n lower limit of hole = %0.3f mm\n
    higher limit of hole = %0.3f mm\n lower limit of
    shaft = %0.3f mm" , Sh,Hl,Hh,Sl1)
printf("\n Shaft basis system \n upper limit of
    shaft = %0.3f mm\n lower limit of shaft = %0.3f
    mm\n lower limit of hole = %0.3f mm\n upper limit
    of hole = %0.3f mm" , Su,Sl2,Hl1,Hu)
```

Scilab code Exa 9.10 Determine size of hole and shaft

```
clc
aa = 0.04 // average allowance in mm
a = 0.012 // allowance in mm
Max = aa+a // maximum allowance in mm
Min = aa-a // minimum allowance in mm
t = (Max-Min)/3 // tolerence in mm
ts = t // tolerence in shat in mm
th = 2*t // tolerence in hole in mm
b = 100 // basic size in mm
Hl = b // lower limit of hole in mm
Hu = b+th // upper limit of hole in mm
Su = b-0.028 // upper limit of shaft in mm
Sl = Su-ts // lower limit of shaft in mm
printf("\n lower limit of hole = %d mm\n upper limit
    of hole = %0.3f mm\n upper limit of shaft = %0.3
    f mm\n lower limit of shaft = %0.3f mm" ,Hl,Hu,Su
    ,Sl)
```

Chapter 11

Surface finish

Scilab code Exa 11.1 Calculate CLA value

```
clc
v = 15000 // vertical magnification
h = 100 // horizontal magnification
l = 0.8 // sampling length in mm
a1 = 160 // area above datum line in mm^2
a2 = 90 // area above datum line in mm^2
a3 = 180 // area above datum line in mm^2
a4 = 50 // area above datum line in mm^2
a5 = 95 // area below datum line in mm^2
a6 = 65 // area below datum line in mm^2
a7 = 170 // area below datum line in mm^2
a8 = 150 // area below datum line in mm^2
a = (a1+a2+a3+a4+a5+a6+a7+a8)/(v*h)
CLA= a/l
printf("\n C.L.A value = %0.2f*10^-6 m " , CLA*1000)
```

Scilab code Exa 11.2 Calculate average and rms value

```
clc
// from figure 11.23
y1 = 0.15 // mu_m
y2 = 0.25 // mu_m
y3 = 0.35 // mu_m
y4 = 0.25 // mu_m
y5 = 0.30 // mu_m
y6 = 0.15 // mu_m
y7 = 0.10 // mu_m
y8 = 0.30 // mu_m
y9 = 0.35 // mu_m
y10 = 0.10 // mu_m
y1sqr = y1^2 // mu_m
y2sqr = y2^2// mu_m
y3sqr = y3^2 // mu_m
y4sqr = y4^2 // mu_m
y5sqr = y5^2 // mu_m
y6sqr = y6^2 // mu_m
y7sqr = y7^2 // mu_m
y8sqr = y8^2 // mu_m
y9sqr = y9^2 // mu_m
y10sqr = y10^2 // mu_m
n = 10
yn = (y1+y2+y3+y4+y5+y6+y7+y8+y9+y10)/n //
   arithmetic average in mu_m
rms = sqrt((y1sqr+y2sqr+y3sqr+y4sqr+y5sqr+y6sqr+
   y7sqr+y8sqr+y9sqr+y10sqr)/n) // r.m.s value in
   mu_m
printf("\n The arithmetic average = %0.2f*10^-6 m \n
    The r.m.s. value = %0.3f*10^-6 m",yn,rms)
```

Chapter 13

Analysis of metal forming processes

Scilab code Exa 13.1 To find drawing load

```
clc
sigma_0 = 240 // N/mm^2
d1 = 5 // initial wire diameter in mm
d0 = 5.5 // final wire diameter in mm
x = d1/d0 // mm
alpha = 8 // angle of contact
alpha = alpha*%pi/180
mu = 0.1 // coefficient of friction
B = mu*cotg(alpha)
sigma_d  = (sigma_0*(1+B)*(1-(x)^(2*B)))/B // N/mm^2
l = 3 // die land in mm
mu = 0.1 // coefficient of friction
r1 = d1/2 // mm
sigma_t = sigma_0 - (sigma_0 - sigma_d)/exp((2*mu*l)
   /r1) // N/mm^2
dl = sigma_t*%pi*(r1)^2 // drawing load in N
printf("\n Total drawing load = %0.1f N" , dl)
// Answers vary due to round off error
```

Scilab code Exa 13.2 Calculate drawing force

```
clc
alpha = 15 // angle of contact
alpha = alpha*%pi/180
bita = 0 // degree
mu = 0.1 // coefficient of friction
mu1 = mu
mu2 = mu
h1 = 1.75 // mm
h0 = 2.5 // mm
B = (mu1+mu2)/(tan(alpha)-tan(bita))
y1 = (1+B)*(1-(h1/h0)^B)/B  //sigma_d/sigma_0 for
    plug mandrels in N/mm^2
z = 1/((h0/h1)-1)
y2 = log10(z)// sigma_d/sigma_0 for movable mandrels
     in N/mm^2
printf("\n The pipe drawing force force on plug
    mandrels = %0.3f \n The pipe drawing forcw on
    mandrels = %0.3f",y1,y2)
disp("Use of movable mandrel substantially reduces
    drawing force")
```

Scilab code Exa 13.3 find neutral section slips and pressure

```
clc
h0 = 25 // thickness of plate in mm
h1 = 20 // mm
delta_h = h0-h1 // mm
sigma = 100 // maximum pressure in N/mm^2
D = 500 // rolled diameter in mm
r = D/2 // rolled radius in mm
```

```
alpha = acos(1-(delta_h/D)) // angle of contact in
    radians
mu = tan(alpha) // coefficient of friction
Ho = 2*sqrt(r/h1)*atan(sqrt(r/h1)*mu)
Hn = (Ho - (log(h0/h1))/mu)/2
theta = sqrt(h1/r)*tan(sqrt(h1/r)*(Hn/2)) // radian
hn = h1 + r*theta^2 // neutral section in mm
x = hn/h0
bs = (1-x)*100 // backward slip
y = hn/h1
fs = (y-1)*100 // forward slip
sigma0 = 2*sigma/sqrt(3)
pn = sigma0*hn*exp(mu*Hn)/h1 //N/mm^2
printf("\n Neutral section = %0.1f mm" , hn)
printf("\n Backward slip = %0.1f percent\n Forward
    slip = %0.1f percent" , bs,fs)
printf("\n Maximum pressure = %0.1f N/mm^2" , pn)
 // 'Answers vary due to round off error'
```

Scilab code Exa 13.4 To determine maximum force

```
clc
Do = 250 // diameter in mm
ho = 250 // hieght in mm
delta_h = 100 // mm
h = 150 // mm
sigma0 = 55 // N/mm^2
d = Do*sqrt(ho/(ho-delta_h)) // diameter in mm
mu = 0.42 // coefficient of friction
R = 162.5 // mm
pa = sigma0/2*(h/(mu*R))^2*(%e^(2*mu*R/h)-2*mu*R/h
    -1) // N/mm^2
p = pa*%pi*(R)^2 // force in kN
printf("\n Force = %d kN",p/1000)
```

Scilab code Exa 13.5 Determine sticking radius and total load

```
clc
d = 150 // diameter in mm
h = 10 // thickness in mm
R = d/2 // radius in mm
mu = 0.2 // coefficient of friction
sigma_0 = 200 // N/mm^2
Rs = R - (h/(2*mu))*log(1/(sqrt(3)*mu)) // sticking
    radius in mm
Ps = sigma_0*exp(2*mu*(R-Rs)/h) // pressure at
    sticking radius in N/mm^2
function y=f(r)
    y=2*%pi*r*sigma_0*exp(2*mu/h*(R-r))
endfunction
L_sld = intg(48.5,75,f)
L_sld = L_sld/1000 // load on sliding portion in kN
Pc = Ps + (2*sigma_0*Rs)/(h*sqrt(3)) // pressure at
    centre in N/mm^2
L_sp = (Pc+Ps)*%pi*(Rs)^2/(2*1000) // load on
    sticking portion in kN
F_l = L_sld + L_sp // total forging load in kN
printf("\n Sticking radius = %0.1f mm \n Total
    forging load = %0.3f MN",Rs ,F_l/1000)
// 'Answers vary due to round off error'
```

Scilab code Exa 13.7 To find drawing load and power

```
clc
RA = 0.30
d = 12 // diameter in mm
alpha = 6  // angle of contact in degree
```

```
alpha = 6*%pi/180 // angle of contact in radian
mu = 0.10 // coefficient of friction
sigma_0 = 240  // N/mm^2
B = mu*cotg(alpha)
x = 1 - RA
sigma_d  = (sigma_0*(1+B)*(1-(x)^B))/B // N/mm^2
r1  = sqrt(x)*(d/2) // mm
l = sigma_d*%pi*(r1)^2 // load in kN
ita = 98 // efficiency
ita = ita/100
s = 2.3 // drawing speed in m/s
P = (l*s)/ita // kW
printf("\n Drawing load = %0.2f kN\n Power of motor
   = %0.2f kW" , l/1000 ,P/1000 )
// 'Answers vary due to round off error'
```

Scilab code Exa 13.8 calculate drawing load and power rating

```
clc
mu1 = 0.15 // coefficient of friction
mu2 = 0.18 // coefficient of fricton
alpha = 14 // angle of contact in degree
alpha = alpha*%pi/180
bita = 10 // semi-cone angle in degree
bita = bita*%pi/180
sigma_0 = 1.40 // kN/mm^2
h0 = 1.5 //mm
h1 = 1 // mm
B = (mu1+mu2)/(tan(alpha)+tan(bita))
sigmad = (sigma_0*(1+B)*(1-(h1/h0)^B))/B // drawing
    stress in kN/mm^2
d1 = 11 // outside diameter in mm
t = 1 // thickness in mm
d2 = d1-2*t // mm
a = (%pi*((d1)^2-(d2)^2))/4 // area in mm^2
```

```
l = sigmad*a // load in kN
s = 0.65 // drawing speed in m/s
w = l*s // work in kJ/s
p = w // power in kW
printf("\n Drawing load = %0.3f kN\n Power rating of
    motor = %0.2f kW" , l , p)
// 'Answers vary due to round off error'
```

Scilab code Exa 13.9 To calculate forging loads

```
clc
sigma_0 = 50 // pressure at start in MPa
B = 0.9 // width in m
h1 = 0.2 // thickness in m
b = 0.3 // tool bite in m
// At commencement of forging
FL = sigma_0*B*b*(1+(b/(4*h1))) // forging load in
   MN
// At completion of forging
h2 = 0.1 // thickness in m
sigma_0c = 150 // pressure at completion in MPa
FL2 = sigma_0c*B*b*(1+(b/(4*h2))) // forging load in
    MN
printf("\n Forging load at start of forging = %0.4f
   MN\n Forging load at completion of forging = %0.3
   f MN" , FL , FL2)
```

Scilab code Exa 13.10 Determine extrusion load

```
clc
sigma_0 = 250 // N/mm^2
d1 = 5 // initial wire diameter in mm
d0 = 15 // final wire diameter in mm
```

```
r0 = d0/2
r1 = d1/2
x = (r0/r1)^2 // mm
alpha = 45 // angle of contact
alpha = alpha*%pi/180
mu = 0.1 // coefficient of friction
B = mu*cotg(alpha)
sigma_x0  = (sigma_0*(1+B)*(1-(x)^B))/B // N/mm^2
sigma_x0 = -sigma_x0
l = 37.5 // length 0f billet in mm
tau1 = sigma_0/2 // Mpa
Pe = sigma_x0 + (4*tau1*l)/d0 // extrusion pressure
    in Mpa
el = Pe*%pi*(r0)^2 // extrusion load in MN
printf("\n Extrusion load = %d MN" , el/10000)
```

Scilab code Exa 13.11 To find roll pressures

```
clc
h0 = 4.05 // thickness of plate in mm
h1 = 3.55 // mm
D = 500 // rolled diameter in mm
r = D/2 // rolled radius in mm
mu = 0.04 // coefficient of friction
sigma = 210 // N/mm^2
delta_h = h0-h1 // mm
p = 2*sigma/sqrt(3) // N/mm^2
alpha = acos(1-(delta_h/D)) // angle of contact
Ho = 2*sqrt(r/h1)*atan(sqrt(r/h1)*alpha)
Hn1 = (Ho - (log(h0/h1))/mu)/2
theta = sqrt(h1/r)*tan(sqrt(h1/r)*(Hn1/2)) //
    radians
hn = h1 + 2*r*(1-cos(theta)) // mm
pn1 = p*hn*exp(mu*Hn1)/h1 // roll pressure in N/mm^2
// b) roll pressure when coefficient of friction is
```

```
0.4
mu2 = 0.4 // coefficient of friction
Hn2 = (Ho - (log(h0/h1))/mu2)/2
theta = sqrt(h1/r)*tan(sqrt(h1/r)*(Hn2/2)) // radians
hn2 = h1 + r*theta^2 // mm
pn2 = (p*hn2*exp(mu2*Hn2))/h1 // roll pressure in N/mm^2
// c) if tension is applied of 35 N/mm^2
sigma_f = 35 // front tension in N/mm^2
pn3 = (p-sigma_f)*hn*exp(mu*Hn1)/h1 // roll ressure in N/mm^2
printf("\n (a) Roll pressure at enter and exit = %0.1f N/mm^2\n     Roll pressure at neutral plane = %0.2f N/mm^2" ,p , pn1)
printf("\n (b) Roll pressure at neutral point when co-efficient of friction is 0.40 = %0.2f N/mm^2", pn2)
printf("\n (c) Roll pressure when 35 N/mm^2 tension is applied at neutral point = %0.2f N/mm^2" , pn3)
// 'Answers vary due to round off error'
```

Scilab code Exa 13.12 Determine neutral plane

```
clc
h1 = 6.35 // thickness in  mm
mu = 0.2 // coefficient of friction
r = 50 // rolled radius in cm
r = r*10 // mm
R = 30 // reduction in percent
h0 = h1*100/(100-R) // mm
delta_h = h0-h1 // mm
alpha = acos(1-(delta_h/(2*r))) // angle of contact
Ho = 2*sqrt(r/h1)*atan(sqrt(r/h1)*alpha)
```

```
Hn = (Ho - (log(h0/h1))/mu)/2
theta = sqrt(h1/r)*tan(sqrt(h1/r)*(Hn/2)) // neutral
    plane in radians
theta = theta*180/%pi // neutral plane in degrees
printf("\n Neutral plane = %0.2f degree", theta)
// 'Answers vary due to round off error'
```

Chapter 14

Theory of metal cutting

Scilab code Exa 14.1 calculate the tool life

```
clc
v1 = 18 // cutting speed in m/min
t1 = 3 // tool life in hours
n = 0.125 // exponent
c = v1*(t1*60)^n // constant
v2 = 24 // cutting speed in m/min
t = (c/v2)^(1/0.125) // tool life in min.
printf("Tool life = %d min." , t)
```

Scilab code Exa 14.2 Calculate the optimum cutting speed

```
clc
c_t = 8 // tool change time in min.
r_t = 5 // tool re-grind time in min.
mr_c = 5 // machine running cost per hour
d = 30 // total depreciation per re-grind in paisa
n = 0.25 // exponent
c = 150 // constant
```

```
c_c = mr_c*c_t/60 // total change cost in Rs
r_c = mr_c*r_t/60 // regrind cost in Rs
ct = c_c+r_c+d/100 // tooling cost in Rs
cm = mr_c/60 // machining cost in Rs
v = c*((cm*n)/(ct*(1-n)))^n // cutting speed in m/min.
printf("\n Cutting speed = %0.1f m/min." , v)
```

Scilab code Exa 14.3 To find different orthogonal cutting picture

```
clc
mu1 = 0.15 // coefficient of friction
mu2 = 0.18 // coefficient of fricton
alpha = 14 // angle of contact in degree
alpha = alpha*%pi/180
bita = 10 // semi-cone angle in degree
bita = bita*%pi/180
sigma_0 = 1.40 // kN/mm^2
h0 = 1.5 //mm
h1 = 1 // mm
B = (mu1+mu2)/(tan(alpha)+tan(bita))
sigmad = (sigma_0*(1+B)*(1-(h1/h0)^B))/B // drawing stress in kN/mm^2
d1 = 11 // outside diameter in mm
t = 1 // thickness in mm
d2 = d1-t // mm
a = (%pi*((d1)^2-(d2)^2))/4 // area in mm^2
l = sigmad*a // load in kN
s = 0.65 // drawing speed in m/s
w = l*s // work in kJ/s
p = w // power in kW
printf("\n Drawing load = %0.3f kN\n Power rating of motor = %0.2f kW" , l , p)
clc
t = 0.127 // uncut chip thickness in mm
```

```
b = 6.35 // width of cut in mm
v = 2 // cutting speed in m/s
alpha = 10 // rake angle in degrees
fc = 567 // cutting force in N
ft = 227 // thrust force in N
tc = 0.228 // chip thickness in mm
r = t/tc // chip thickness ratio
alpha = alpha*%pi/180 // rake angle in radians
phi = atan(r*cos(alpha)/(1-(r*sin(alpha)))) // shear angle
phi1 = phi*180/%pi // shear angle
printf("\n Shear angle = %0.2f degree" , phi1)
mu =((fc*sin(alpha)+ft*cos(alpha))/(fc*cos(alpha)-ft*sin(alpha))) //coefficient of friction
bita = atan(mu) // friction angle
bita = bita*180/(%pi)
printf("\n Friction angle = %0.2f degree", bita )
fs = fc*cos(phi)-ft*sin(phi) //shear force in N
taus = (fs*sin(phi))/(b*t) // shear stress
printf("\n Shear stress = %0.1f N/mm^2" , taus)
cp = fc*v/1000 // cutting power in kw
printf("\n Cutting power = %0.3f kw " , cp)
vc = v*r // chip velocity in m/s
printf("\n Chip velocity = %0.3f m/s",vc)
ss = cotg(phi) + tan(phi-alpha) // shear strain
printf("\n shear strain = %0.3f" , ss)
spl = t/sin(phi) // shear plane length
vs = v*cos(alpha)/cos(phi-alpha) // shear velocity
S = vs*10/spl // shear strain rate
S = S*10^3 // shear strain rate
printf("\n Shear strain rate = %.3f s^-1" , S)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.4 To find tool life

```
clc
v = 30 // cutting speed in m/min
feed = 0.3 // feed rate in mm/rev.
d = 2.5 // depth of cut in mm
t = 60 // tool life in min.
c = v*t^0.13*feed^0.77*d^0.37 // constant
printf("\n constant = %0.2f " , c)
v2 = v*1.2 // cutting speed in m/min
t2 = (c/(v2*feed^0.77*d^0.37)) // tool life when cutting speed increased by 20% in min.
t2 = t2^(1/0.13)
f2 = feed*1.2 // feed rate in mm/rev.
t3 = (c/(v*d^0.37*f2^0.77)) // tool life when feed rate increased by 20% in min.
t3 = t3^(1/0.13)
d2 = d*1.2 // depth of cut in mm
t4 = (c/(v*feed^0.77*d2^0.37)) // tool life when depth of cut increased by 20% in min.
t4 = t4^(1/0.13)
t5 = (c/(v2*d2^0.37*f2^0.77)) // tool lfe in min.
t5 = t5^(1/0.13)
printf("\n Tool life when cutting speed increased by 20 = %0.2f min." , t2)
printf("\n Tool life when feed rate increased by 20 = %0.2f min." , t3)
printf("\n Tool life when depth of cut increased by 20 = %0.2f min." , t4)
printf("\n Tool life when all taken together after increasing by 20 = %0.2f min." , t5)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.5 find force and coefficient of friction

```
clc
t = 0.25 // uncut chip thickness in mm
```

```
b = 2.5 // width of cut in mm
v = 2.5 // cutting speed in m/s
alpha = 10 // rake angle in degrees
fc = 1130 // cutting force in N
ft = 295 // thrust force in N
tc = 0.45 // chip thickness in mm
r = t/tc // chip thickness ratio
alpha = alpha*%pi/180 // rake angle in radians
phi = atan((r*cos(alpha))/(1-r*sin(alpha))) // shear
    angle
phi2 = phi*180/%pi // shear angle
fs = fc*cos(phi) - ft*sin(phi) //shear force in N
printf("\n Force of shear at shear plane = %0.2f N"
    , fs)
mu = atan((fc*sin(alpha)+ft*cos(alpha))/(fc*cos(
    alpha)-ft*sin(alpha))) //friction anglele
printf("\n Friction angle = %0.3f degree", mu )
// 'Answers vary due to round off error'
```

Scilab code Exa 14.6 To find terms of orthogonal cutting

```
clc
t = 0.2 // uncut chip thickness in mm
alpha = 15 // rake angle in degrees
tc = 0.62 // chip thickness in mm
r = t/tc // chip thickness ratio
crc = 1/r // chip reduction coefficient
printf("\n Cutting ratio = %0.3f\n Chip reduction co
    -efficient = %0.1f" , r , crc)
alpha = alpha*%pi/180 // rake angle in radians
phi = atan(r*cos(alpha)/(1-r*sin(alpha))) // shear
    angle
phi = phi*180/%pi // shear angle
printf("\n Shear angle = %0.2f degree" , phi)
ss = cotg(phi*%pi/180) + tan((phi*%pi)/180-(alpha*
```

```
%pi)/180) // shear strain
printf("\n shear strain = %0.3f" , ss)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.7 To solve tool life equation

```
clc
v1 = 25 // cutting speed in m/min
t1 = 90 // tool life in min.
v2 = 35 // cutting speed in m/min
t2 = 20 // tool life in min
n = log(v2/v1)/log(t1/t2) // exponent
C = v1*(t1)^n // constant
t = 60 // tool life in min.
v = C/(t)^n // cutting speed in m/min.
printf("\n n = %0.3f\n C = %0.1f\n Cutting speed = %0.2f m/min." , n , C,v)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.8 Determine normal and tangential force

```
clc
t = 0.5 // uncut chip thickness in mm
b = 3 // width of cut in mm
alpha = 15 // rake angle in degrees
alpha = alpha*%pi/180 // rake angle in radians
r = 0.383 // chip thickness ratio
mu = 0.7 // average coefficient of friction on tool face
bita = atan(mu) // friction angle
tau = 280 // yield stress in N/mm^2
phi = atan((r*cos(alpha))/(1-r*sin(alpha))) // shear angle
```

```
fc = (tau*b*t)/(sec(bita-alpha)*cos(phi+bita-alpha)*
    sin(phi)) // cutting force in N
ft = (fc*(mu-tan(alpha)))/(1+mu*tan(alpha)) //
    thrust force in N
F = fc*sin(alpha)+ft*cos(alpha) // tangential force
    on tool face in N
F = ceil(F)
N = fc*cos(alpha)-ft*sin(alpha) // normal force on
    tool face in N
printf("Tangential force on tool face = %d N\n
    normal force on tool face = %0.1f N" , F,N)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.9 To find cutting and thrust force

```
clc
t = 0.25 // uncut chip thickness in mm
b = 0.5 // width of cut in cm
v = 8.2 // cutting speed in m/min.
alpha = 20 // rake angle in degrees
alpha2 = alpha*%pi/180 // rake angle in radians
r = 0.351 // cutting ratio
phi = atan(r*cos(alpha2)/(1-r*sin(alpha2))) // shear
     angle in radians
phi2 = phi*180/%pi // shear angle in degrees
alpha2 = alpha*%pi/180 // rake angle in radians
bita = 35+alpha-phi2 // degrees
s = cotg(phi) + tan(phi-alpha2) // shear strain
e = s/sqrt(3) // natural strain
sigma = 784*(e)^0.15 // tensile property in N/mm^2
tau = sigma/sqrt(3) // yield shear stress in N/mm^2
As = (b*10*t)/sin(phi) // shear plane area in mm^2
Fs = tau*As // shear gorce in N
R = Fs/cos(phi+(bita*%pi/180)-alpha2)
Fc = R*cos((bita*%pi/180)-alpha2) // cutting force
```

```
in N
Ft = R*sin((bita*%pi/180)-alpha2) // thrust force in N
printf("\n Cutting force= %0.1f N\n Thrust force = %0.1f N" , Fc , Ft)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.10 find terms of orthogonal rake system

```
clc
f = 0.2 // feed in mm/rev.
t = 0.2 // uncut chip thickness in mm
alpha = 10 // rake angle in degrees
fc = 1600 // cutting force in N
ft = 850 // thrust force in N
tc = 0.39 // chip thickness in mm
r = t/tc // chip thickness ratio
d = 2 // depth of cut in mm
b = 2 // mm
alpha2 = alpha*%pi/180 // rake angle in radians
phi = atan(r*cos(alpha2)/(1-r*sin(alpha2))) // shear angle in radians
phi2 = phi*180/%pi // shear angle in degree
fs = fc*cos(phi)-ft*sin(phi) //shear force in N
fn = fc*sin(phi)+ft*cos(phi) // normal force in N
f = fc*sin(alpha2)+ft*cos(alpha2) // friction force in N
mu =((fc*tan(alpha2)+ft)/(fc-ft*tan(alpha2))) // kinetic coefficient of friction
s = fc/(b*t) // specific cutting energy in N/mm^2
printf("\n Shear force = %d N\n Normal force = %0.1f N\n Friction force = %0.1f N\n Kinetic coefficient of friction = %0.3f" , fs , fn ,f , mu)
printf("\n Specific cutting energy = %d N/mm^2" , s)
```

```
// 'Answers vary due to round off error'
```

Scilab code Exa 14.11 Calculate CLA

```
clc
cs = 20 // side cutting edge angle in degree
ce = 30 // end cutting edge angle in degree
f = 0.1 // feed in mm/rev.
r = 3 // nose radius in mm
cs2 = cs*%pi/180 // side cutting edge angle in
    radians
ce2 = ce*%pi/180 // end cutting edge angle in
    radians
h = (1-cos(ce2))*r + f*sin(ce2)*cos(ce2) - sqrt((2*f
    *r*(sin(ce2))^3)-((f^2)*(sin(ce2))^4))
Ra = h/4 // Centre line average roughness in mm
printf("\n Centre line average roughness = %0.2f
    *10^-6m" , Ra*10^3)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.12 Calculate back and side rake angle

```
clc
i = 0 // inclination angle in degree
alpha = 10 // orthogonal rake angle in degree
lemda = 75 // principal cutting edge angle in degree
alpha = alpha*%pi/180 // orthogonal rake angle in
    radian
lemda = lemda*%pi/180 // principal cutting edge
    angle in radian
alpha_b = atan(cos(lemda)*tan(alpha)+sin(lemda)*tan(
    i)) //back rake angle in radians
```

```
alpha_b = alpha_b*180/%pi // back rake angle in
    degree
alpha_s = atan(sin(lemda)*tan(alpha)-cos(lemda)*tan(
    i)) // side rake angle in radians
alpha_s = alpha_s*180/%pi // side rake angle in
    degree
printf("\n Back rake angle = %0.2f degree\n Side
    rake angle = %0.2f degree" , alpha_b,alpha_s)
```

Scilab code Exa 14.13 Calculate inclination and rake angle

```
clc
alphab = 8 // back rake in degree
alphas = 4 // side rake in degree
cs = 15 // side cutting edge angle in degree
lemda = 90 - cs // approach angle in degree
alphab = alphab*%pi/180 // back rake in radian
alphas = alphas*%pi/180 // side rake in radian
cs = cs*%pi/180 // side cutting edge angle in radian
lemda = lemda*%pi/180 // approach angle in radian
alpha = atan(tan(alphas)*sin(lemda)+tan(alphab)*cos(
    lemda)) // orthogonal rake angle in radian
alpha = alpha*180/%pi // orthogonal rake angle in
    degree
i = atan(sin(lemda)*tan(alphab)-cos(lemda)*tan(
    alphas)) // inclnation angle in radian
i = i*180/%pi // inclnation angle in degree
printf("\n Othogonal rake angle = %0.2f degree\n
    Inclination angle = %0.1f degree" , alpha , i)
```

Scilab code Exa 14.14 find different powers and resistance

```
clc
```

```
cs = 15 // side cutting edge angle in degree
v = 0.2 // cutting speed in m/s
f = 0.5 // feed rate in mm/rev.
d = 3.2 // depth of cut in mm
fc = 1593*(f)^0.85*(d)^0.98 // cutting force in N
pc = fc*v/1000 // cutting power in kw
ita_mt = 0.85 // efficiency of lathe
pm = pc/ita_mt // motor power in kw
a = f*d // area of uncut chio in mm^2
r = fc/a // specific cutting resistance in N/mm^2
p = pc/(a*v)// unit power in W/(mm^3)*s
printf("\n Cutting power = %0.3f kw\n Motor power =
    %0.2f kw\n Specific cutting resistance = %0.2f N/
    mm^2\n Unit power = %0.3f W/(mm^3)*s", pc,pm,r,p
    )
// 'Answers vary due to round off error'
```

Scilab code Exa 14.15 Calculate percentage increase in tool life

```
clc
C = 400
n=0.5
a=2 // (T1/T2)^n
b=2^(1/n) // T2
i = (b-1)*100 // percentage increase
printf("\n Percentage increase = %d percent", i)
```

Scilab code Exa 14.16 To find percentage of total energy

```
clc
t = 0.127 // uncut chip thickness in mm
b = 6.35 // width of cut in mm
v = 1.20 // cutting speed in m/min.
```

```
alpha = 10 // rake angle in degrees
fc = 556.25 // cutting force in N
ft = 222.50 // thrust force in N
tc = 0.229 // chip thickness in mm
r = t/tc // chip thickness ratio
R = sqrt((fc^2)+(ft^2))
bita = (acos(fc/R)) + alpha*%pi/180 //
f = R*sin(bita) //
fe = f*r // friction energy
p = (f*r*100)/fc // percentage of fricton enrgy and total energy
printf("\n The percentage of total energy that goes into overcoming friction at tool chip interface = %0.2f percent" , p)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.17 To find power and different energies

```
clc
D = 300 // diameter in mm
r = 45 // rev/min.
d = 2 // depth of cut in mm
f = 0.3 // feed in mm/rev
fc = 1850 // cutting force in N
ff = 450 // feed force in N
V = 2.5*10^6 // metal removed in mm
v = (%pi*D*r)/(60*1000) // cutting velocity in m/s
pc = fc*v/1000 // cutting power in kW
fv = f*r/60*1000 // feed velocity in m/s
fp = fv*ff // feed power in W
mrr = d*f*v*60*1000 // mm^3/min.
ce = pc*1000*60/mrr // specific cutting energy in W*s/mm^2
E = ce*V/(3600*1000) // energy consumed
printf("\n Power consumption = %0.2f W\n Specific
```

```
cutting energy = %0.2f W*s/mm^3\n Energy consumed = %0.3f kWh" ,pc,ce,E)
// 'Answers vary due to round off error'
```

Scilab code Exa 14.18 Determine components of force and power

```
clc
D = 100 // diameter in mm
cs = 30 // side cutting edge angle in degree
lemda = 90 - cs // approach angle in degree
d = 2.5 // depth of cut in mm
f = 0.125 // feed in mm/rev.
N = 300 // turning speed of job in rev./min.
mu = 0.6 // coefficient of friction
tau = 400 // ultimate shear stress in Mpa
bita = atand(mu) // friction angle in radian
alphas = 10 // side rake angle
alphab = 6 // back rake angle
alpha = atand(tand(alphas)*sind(lemda)+tand(alphab)*cosd(lemda)) // orthogonal rake angle in degree
phi = 45 - ( bita - alpha) // shear angle
Fc = tau*d*f/(secd(bita-alpha)*cosd(phi+bita-alpha)*sind(phi)) // cutting force in N
Ft = Fc*tand(bita-alpha) // thrust component in N
Ff = Ft*sind(lemda) // feed force along axis of job in N
Rf = Ft*cosd(lemda) // radial force normal to axis of job in N
v = %pi*D*N/(1000*60) // velocity in m/s
p = Fc*v // power in watts
printf("\n Cutting force = %d N\n Thrust force = %0.3f N\n Feed force = %0.1f N\n Radial force = %0.3f N\n Cutting power = %d watts" , Fc,Ft,Ff,Rf,p)
// 'Answers vary due to round off error'
```

Chapter 15

Design and manufacture of cutting tools

Scilab code Exa 15.1 calculate horsepower at cutter and motor

```
clc
w = 10 // width of cut in cm
h = 0.32 // depth of cut in cm
n = 8 // number of teeth in cutter
ft = 0.033 // feed rate per tooth
N = 200 // cutter speed in rpm
ita = 60/100 // efficiency
f = n*ft*N // feed rate in cm/min.
mrr = w*h*f // metal removal rate in cm^3/min.
k = 8.2 // machinibility factor from table 15.3
hpc = mrr/k // horsepower at cutter
hpm = hpc/ita // horsepower at motor
printf("\n Horsepower at cutter = %0.2f\n Horsepower
    at motor = %0.2f" , hpc , hpm)
// 'Answers vary due to round off error'
```

Scilab code Exa 15.2 Determine broaching power and Design broach

```
clc
l = 35 // length of bore in mm
v = 0.15 // cutting speed in m/s
t1 = 0.01 // upper limit in mm
t2 = 0.05 // upper limit in mm
D = 32.25 // finished broach in mm
D1 = 32.25+t2 // mm
d = 32.75 // finish diameter in mm
d1 = 32.75 + t1 //finish diameter of hole in mm
s = 0.05 // mm
B = 1.30 // blunt broach factor
c = 45 // specific cutting force in N/mm^2
n = 3 // number of teeth cutting at a time
F = n*%pi*d1*s*c*B // force needed for broaching in N
bp = F*v/1000 // Broaching power in kw
// broach design
p = 1.75*sqrt(l) // pitch in mm
theta = 10 // face angle in degree
alha1 = 1.5 // relief angle for roughing in degree
alha2 = 1.0 // relief angle for finishing in degree
w = 0.3*p // width of land in mm
h = 0.4*p // depth of cutting teeth in mm
r = 0.3*p // tooth fillet radius in mm
T = (d1-D1)/2 // mm
n = T/s // number of cutting teeth
n = round(n)
l = (n+7)*p //length of toothed portion of broach in mm
printf("\n (i)Broaching power = %0.4f kW",bp)
disp("(ii) Broach Design")
printf(" (a) Pitch diameter = %0.2fmm\n (b) width of land = %0.2f mm \n    depth of cutting teeth = %0.2f mm\n    Tooth fillet radius= %0.2f mm", p,w,h,r)
printf("\n (c) Length of toothed portion of broach =
```

```
%d mm", l)
// 'Answers vary due to round off error'
```

Scilab code Exa 15.3 Estimate moment thrust force and power

```
clc
Hb = 200 // brinell hardness
d = 12.7 // diameter in mm
f = 0.254 // feed in mm/rev.
N = 100 // rpm
M = (Hb*(d)^2*f)/8 //moment in kgf-mm
k = 1.1 //material factor
p = (1.25*(d)^2*k*N*(0.056+1.5*f))/(10)^5 // power in kW
T1a = (1.7*M)/d // thrust force kgf
T1b = (3.5*M)/d // kgf
T1 = (T1a+T1b)/2 // average
w = 0.14*d // thickness in mm
T2a = (0.1*%pi*(w)^2*Hb)/4 // thrust force kgf
T2b = (0.2*%pi*(w)^2*Hb)/4 // thrust force kgf
T2 = (T2a+T2b)/2 // average
avg = T1+T2 // kgf
thrust = 1.16*k*d*(100*f)^0.85 // kgf
printf("\n Moment = %0.1f kgf-mm\n Power = %0.3f hp\n Average force = %d kgf\n Thrust force = %0.1f kgf" ,M, p ,avg , thrust)
// Error in textbook
```

Scilab code Exa 15.4 Design shell inserted blade reamer

```
clc
d = 55 // diameter in mm
ul = 0.035 // upper limit in mm
```

```
ll = 0.000 // lower limit in mm
Dmax = d+ul // maximum diameter of hole in mm
Dmin = d+ll // minimum diameter of hole in mm
IT = 0.035 // hole tolerence in mm
dmax = Dmax-0.15*IT // maximum diameter of reamer in
    mm
dmin = dmax-0.35*IT // minimum diameter of reamer in
    mm
l = ((d/4)+(d/3))/2 // length of guiding section in
   mm
Z = 1.5*sqrt(d)+2 // number of teeth
Z = ceil(Z)
printf("\n 1 Diameter of reamer \n Maximum diameter
    of reamer = %0.3f mm \n Minimum diameter of
    reamer = %0.3f mm \n 2 Back taper = 0.05 mm \n 3
    Values of various angles \n Rake angle = 5 degree
     \n Plan approach angle = 45 degree \n Circular
    land = 0.25 to 0.50 mm \n Secondary clearance
    angle = 10 degree \n 4 Length of reamer \n Length
     of fluted portion = 82.5 mm \n Length of reaming
     allowance = 0.18 mm \n Length of cutting section
     = 2.25 mm \n Length of guiding section = %d mm \
    n 5 Number of teeth = %d" , dmax,dmin,l,Z)
// Answer vary due to round off error
```

Scilab code Exa 15.5 To design single point cutting tool

```
clc
Pm = 10 // power of motor in kw
v = 40 // cutting speed in m/min.
ita = 70 // efficiency
ita = ita/100
Pc = Pm*ita
Fc = (Pc*1000*60)/v // cutting force
sigmab = 250 // stress in Mpa
```

```
B = sqrt((Fc*1.25*6)/(sigmab*1.6)) // width of shank in mm
h = 1.6*B // hieght of shank in mm
l = 1.25*h // shank overang in mm
printf("\n The width of shank = %0.1f mm\n Height of shank = %0.2f mm\n Shank overhang = %0.1f mm" , B,h,l)
// 'Answers vary due to round off error'
```

Scilab code Exa 15.8 find various terms for stainless steel

```
clc
l = 150 // length in mm
D = 12.70 // diameter in mm
dia = 12.19 // diameter on centre lathe in mm
N = 400 // spindle speed in rev./min
s = 203.20 // axial speed in mm/min.*####
v = (%pi*D*N)/(1000*60) // cutting velocity in m/s
d = (D-dia)/2 // depth of cut in mm
f = s/N // feed in mm/rev.
Dave = (D+dia)/2 // average diameter in mm
V = %pi*Dave*N
a = d*f // area of cut in mm^2
mrr = a*V // metal removal rate in mm^3/min.
T = l/(f*N) // machine timing in min.
c = 56 // constant from table
p = d*f*v*60*c // power in watts
omega = (2*%pi*N)/60 // rpm
t = p/omega // torque in Nm
Fc = (2*t*1000)/Dave // cutting force in N
printf("\n Cutting speed = %0.2f m/s\n MRR = %d mm^3/min.\n Time to cut = %0.2f min.\n Power = %0.1f watts\nCutting force = %d N" , v , mrr , T ,p , Fc)
// Answers are given wrong in book
```

Scilab code Exa 15.9 To find MRR power and torque

```
clc
f = 0.2 // feed in mm/rev.
N = 800 // spindle speed in rev./min.
d = 10 // doameter of hole in mm
mrr = %pi*(d^2)*f*N/4 // metal removal rate in mm^3/
    min.
mrr = mrr/60 // mm^3/s
p = 0.5*mrr // cutting power from table 14.2 in
    watts
omega = 2*%pi*N/60 // rpm
T = p/omega // torque in N.m
printf("\n MRR = %0.2f mm^3/s\n Cutting power = %0.3
    f watts\n Torque = %0.2f N.m" , mrr,p,T)
```

Scilab code Exa 15.10 find MRR power torque and time

```
clc
l = 300 // length in mm
w = 100 // width in mm
f = 0.25 // feed in mm/tooth
d = 3.2 // depth of cut in mm
D = 50 // cutter diameter in mm
n = 20 // number of cutter teeth
N = 100 // cutter speed in rev./min.
tf = f*n*N // table feed in mm/min.
mrr = w*d*tf // metal removal rate in mm^3/min.
mrr = mrr/60 // mm^3/s
p = 6*mrr // cutting power from table 14.2 in watts
omega = 2*%pi*N/60 // rpm
```

```
T = p/omega // torque in N.m
att = sqrt((D*d)-(d^2)) // added table travel in mm
t = (l+att)/tf // cutting time in min.
t = t*60 // s
printf("\n MRR = %0.2f mm^3/s\n Cutting power = %d
    watts\n Torque = %0.2f N.m\n Cutting time = %0.1
    fs" , mrr,p,T,t)
```

Chapter 16

Gear manufacture

Scilab code Exa 16.1 Calculate settings of gear tooth

```
clc
n = 34 // number of teeths
m = 5 // module in mm
w = m*n*sin (%pi/(n*2)) //tooth thickness in mm
h = m*(1+(n*(1 - cos(%pi/(n*2)))/2)) // chordal addendum in mm
printf("\n Tooth thickness = %0.3f mm\n Chordal addendum = %0.3f mm" ,w ,h)
// 'Answers vary due to round off error'
```

Chapter 17

Thread manufacturing

Scilab code Exa 17.1 Calculate best wire size

```
clc
d = 80 // outside diameter in mm
p = 6  // pitch diameter in mm
d = 0.5774*p // best wire size in mm
printf("\n Best wire size = %0.3f mm" , d)
```

Scilab code Exa 17.2 Calculate size and distances over wire

```
clc
D = 20 // diameter in mm
p = 2.5 // pitch diameter in mm
d = 0.5774*p // mm
W = D+3*d-1.5156*p// best wire size in mm
printf("\n Best wire size = %0.3f mm\n Distance over
    wires = %0.3f mm" ,d, W)
// Answer vary due to round off error
```

Chapter 21

Statical quality control

Scilab code Exa 21.1 Construct R and X chart

```
clc
clf()
n = 10 // number of samples
A2 = 0.577
D3 = 0
D4 = 2.115
// number of defectives
x1 = 11.274
x2 = 11.246
x3 = 11.204
x4 = 11.294
x5 = 11.252
x6 = 11.238
x7 = 11.230
x8 = 11.276
x9 = 11.208
x10 = 11.266
r1 = 0.15
r2 = 0.20
r3 = 0.33
r4 = 0.46
```

```
r5 = 0.10
r6 = 0.15
r7 = 0.20
r8 = 0.23
r9 = 0.50
r10 = 0.30
x = x1+x2+x3+x4+x5+x6+x7+x8+x9+x10
r = r1+r2+r3+r4+r5+r6+r7+r8+r9+r10
Xavg = x/n
Ravg = r/n
// for X chart
ucl1 = Xavg + A2*Ravg
lcl1 = Xavg - A2*Ravg
// for R chart
ucl2 = D4*Ravg
lcl2 = D3*Ravg
printf("\n control limits \n  For X charts \n UCL = %0.2f cm \n LCL = %0.2f cm\n For R charts \n UCl = %0.3f \n LCL = %0.3f" , ucl1,lcl1,ucl2,lcl2)
// X chart
x=[1,2,3,4,5,6,7,8,9,10];
y =[11.274,11.246,11.204,11.294,11.252,11.238,11.230,11.276,11.208,
plot(x,y)
xtitle("X chart","Sample No.","X")
// R chart
xset("window",1)
z = [0.15,0.20,0.33,0.46,0.10,0.15,0.20,0.23,0.50,0.30]
plot(x,z)
xtitle("R chart" ,"Sample no.", "R")
```

Scilab code Exa 21.2 Construct the control charts

```
clc
clf()
n = 100 // total number of sub groups
s = 10 // number of samples
// number of defectives
d1 = 3
d2 = 2
d3 = 3
d4 = 5
d5 = 3
d6 = 3
d7 = 2
d8 = 4
d9 = 3
d10 = 2
d = d1+d2+d3+d4+d5+d6+d7+d8+d9+d10 // total number
    of defectives
p1 = d/(n*s) // average fraction of defectives
sigmap1 = sqrt(p1*(1-p1)/n)
ucl1 = p1 + 3*sigmap1
lcl1 = p1 - 3*sigmap1
// control chart for fraction defectives
x = linspace(0,10,10)
y = linspace(0,0.081,10)
plot(x,y)
xtitle("Control chart for fraction defectives" , "
    Samples" ,"Fraction defectives")
// percent defective (mean)
p1 = p1*100
sigmap2 = sqrt(p1*(100-p1)/n)
ucl2 = p1 + 3*sigmap2
lcl2 = p1 - 3*sigmap2
printf("\n Control limits \n Fraction defectives \n
    UCL = %0.3f\n LCL = %0.4f\n Percent defectives \n
     UCL = %0.1f \n LCL = %0.1f", ucl1,lcl1,ucl2,lcl2
    )
// control chart for percent defect
xset("window" ,1)
```

```
z = linspace(0,8.1,10)
plot(x,z)
xtitle("Control chart for percent defects" , "Sample no." , "percent defects")
//  'Answers vary due to round off error'
```

Scilab code Exa 21.4 Calculate poisson probabilities

```
clc
n = 1000 // number of units
s = 4 // random sample
d = 50 // defectives
z = d*s/n
pp0 = exp(-0.2)*1 // poisson probabilities for 0 defectives
pp1 = exp(-0.2)*(z) // poisson probabilities for 1 defectives
pp2 = exp(-0.2)*(z^2/factorial(2)) // poisson probabilities for 2 defectives
pp3 = exp(-0.2)*(z^3/factorial(3))// poisson probabilities for 3 defectives
printf("\n Proabilities for 0,1,2 and 3 defectives are : %0.3f ,%0.4f, %0.4f, %0.5f" , pp0,pp1,pp2,pp3)
```

Scilab code Exa 21.5 Calculate probabilities of defective items

```
clc
d = 50 // defectives
l = 1000 // lot of pieces
p = d/l // proability of an event happening
q = 1-p // proability of an event not happening
n = 4 // sample size
```

```
p0 = q^n //probabilities for 0 defectives
p1 = 4*(q)^3*p // probabilities for 1 defectives
p2 = 6*(q)^2*p^2 // probabilities for 2 defectives
p3 = 4*q*(p)^3 // probabilities for 3 defectives
printf("\n  Proabilities for 0,1,2 and 3 defectives
    are : %0.4f %0.4f %0.4f %0.6f" , p0,p1,p2,p3)
```

Scilab code Exa 21.6 Determine producers and consumers risk

```
clc
// producer's risk
n = 71 // sample size
AQL = 0.005
LTPD = 0.05
l_s = 500 // lot size
z1 = n*AQL // mean number of defects
pp1 = exp(-z1)+z1*exp(-z1) // poisson proability for
     1 or less defective
alpha = (1-pp1)*100 // producer's risk
alpha = ceil(alpha)
// consumer's risk
z2 = n*LTPD // mean number of defects
pp2 = exp(-z2)+z2*exp(-z2) // poisson proability for
     1 or less defective
bita = pp2*100 // consumer's risk
printf("\n Producers risk = %d percent\n Consumers
    risk = %0.2f percent", alpha,bita)
```

Scilab code Exa 21.7 Evaluate preliminary and revised control limits

```
clc
td1= 20 // total number of days
n1 = 200 // sample size
```

```
// number of defectives
d1 = 10
d2 = 15
d3 = 10
d4 = 12
d5 = 11
d6 = 9
d7 = 22
d8 = 4
d9 = 12
d10 = 24
d11 = 21
d12 = 15
d13 = 8
d14 = 14
d15 = 4
d16 = 10
d17 = 11
d18 = 11
d19 = 26
d20 = 13
d = d1+d2+d3+d4+d5+d6+d7+d8+d9+d10+d11+d12+d13+d14+d15+d16+d17+d18+d19+d20 // total number of defectives
p1 = d/(n1*td1) // average fraction of defectives
sigmap1 = sqrt(p1*(1-p1)/n1)
ucl1 = p1 + 3*sigmap1
lcl1 = p1 - 3*sigmap1
// revised control limits
td2 = 18 // total number of days
D = d - (d10+d19) // number of defects
p2 = D/(n1*td2)
sigmap2 = sqrt(p2*(1-p2)/n1)
ucl2 = p2 + 3*sigmap2
lcl2 = p2 - 3*sigmap2
printf("\n Preliminary control limits \n UCL = %0.3f \n LCL = %0.3f \n Revised control limits \n UCL = %0.3f \n LCL = %0.3f" , ucl1,lcl1,ucl2,lcl2)
```

Scilab code Exa 21.8 Find control limits for c chart

```
clc
n1 = 15 // total number of sub groups
// number of defectives
d1 = 77
d2 = 64
d3 = 75
d4 = 93
d5 = 45
d6 = 61
d7 = 49
d8 = 65
d9 = 45
d10 = 77
d11 = 59
d12 = 54
d13 = 84
d14 = 40
d15 = 92
d = d1+d2+d3+d4+d5+d6+d7+d8+d9+d10+d11+d12+d13+d14+
    d15 // total number of defectives
c1 = d/n1
ucl1 = c1 + 3*sqrt(c1)
lcl1 = c1 - 3*sqrt(c1)
// revised control limits
n2 = 12 // total number of sub groups
D = d - (d4+d14+d15) // number of defects
c2 = D/n2
ucl2 = c2 + 3*sqrt(c2)
lcl2 = c2 - 3*sqrt(c2)
printf("\n Preliminary control limits \n UCL = %0.2f
     \n LCL = %0.2f \n Revised control limits \n UCL
    = %0.3f \n LCL = %0.3f" , ucl1,lcl1,ucl2,lcl2)
```

Scilab code Exa 21.9 find control limits for charts

```
clc
n = 20 // number of samples
A = 1.342
A1 = 1.596
A2 = 0.577
d2 = 2.326
d3 = 0.864
D1 = 0
D2 = 4.918
D3 = 0
D4 = 2.115
// number of defectives
x1 = 3290
x2 = 3180
x3 = 3350
x4 = 3470
x5 = 3080
x6 = 3240
x7 = 3260
x8 = 3310
x9 = 3640
x10 = 4110
x11 = 3220
x12 = 3590
x13 = 4270
x14 = 4040
x15 = 3580
x16 = 3500
x17 = 3570
x18 = 3560
x19 = 2740
x20 = 3200
```

```
r1 = 560
r2 = 410
r3 = 200
r4 = 300
r5 = 90
r6 = 650
r7 = 890
r8 = 410
r9 = 1120
r10 = 520
r11 = 580
r12 = 670
r13 = 480
r14 = 250
r15 = 170
r16 = 670
r17 = 440
r18 = 660
r19 = 560
r20 = 590
x = x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+
    x15+x16+x17+x18+x19+x20
r = r1+r2+r3+r4+r5+r6+r7+r8+r9+r10+r11+r12+r13+r14+
    r15+r16+r17+r18+r19+r20
Xavg = x/n
Ravg = r/n
// for X chart
ucl1 = Xavg + A2*Ravg
lcl1 = Xavg - A2*Ravg
// for R chart
ucl2 = D4*Ravg
lcl2 = D3*Ravg
// Revised control limits
n1 = 15
n2 = 19
X = (x - (x5+x10+x13+x14+x19))/n1
R = (r - (r9))/n2
// for X chart
```

```
ucl3 = X + A2*R
lcl3 = X - A2*R
// for R chart
ucl4 = D4*R
lcl4 = D3*R
printf("\n Preliminary control limits \n  For X
   charts \n UCL = %0.2f \n LCL = %0.2f\n For R
   charts \n UCl = %0.3f \n LCL = %0.3f \n Revised
   control limits \n For X chart \n UCL = %0.3f \n
   LCL = %0.3f\n For R charts \n UCl = %0.3f \n LCL
   = %0.3f" , ucl1,lcl1,ucl2,lcl2,ucl3,ucl3,ucl4,
   lcl4)
//  'Answers vary due to round off error'
```

Scilab code Exa 21.10 Determine producers and consumers risk

```
clc
clf()
n = 50 // sample size
rn = 2 // rejection number
AQL = 0.02
LTPD = 0.08
// Producer's risk
z1 = n*AQL // mean number of defectives
pp1 = exp(-z1)+z1*exp(-z1) // poisson proability for
    1 or less defective
alpha = (1-pp1)*100 // producer's risk
// consumer's risk
z2 = n*LTPD // mean number of defectives
bita = (exp(-z2)+z2*exp(-z2))*100 // consumer's risk
d1 = 1 // incoming defective in percent
z3 = n*d1/100 // average number of defective
ppa1 = exp(-z3)+z3*exp(-z3) // proability of
   acceptance
ppa1 = ppa1*100
```

```
ppr1 = 100-ppa1 // proability of rejection
AOQ1 = ppr1*0 + ppa1*d1/100
d2 = 2 // incoming defective in percent
z4 = n*d2/100 // average number of defective
ppa2 = exp(-z4)+z4*exp(-z4) // proability of
    acceptance
ppa2 = ppa2*100
ppr2 = 100-ppa2 // proability of rejection
AOQ2 = ppr2*0 + ppa2*d2/100
d3 = 4 // incoming defective in percent
z5 = n*d3/100 // average number of defective
ppa3 = exp(-z5)+z5*exp(-z5) // proability of
    acceptance
ppa3 = ppa3*100
ppr3 = 100-ppa3 // proability of rejection
AOQ3 = ppr3*0 + ppa3*d3/100
d4 = 6 // incoming defective in percent
z6 = n*d4/100 // average number of defective
ppa4 = exp(-z6)+z6*exp(-z6) // proability of
    acceptance
ppa4 = ppa4*100
ppr4 = 100-ppa4 // proability of rejection
AOQ4 = ppr4*0 + ppa4*d4/100
d5 = 8 // incoming defective in percent
z7 = n*d5/100 // average number of defective
ppa5 = exp(-z7)+z7*exp(-z7) // proability of
    acceptance
ppa5 = ppa5*100
ppr5 = 100-ppa5 // proability of rejection
AOQ5 = ppr5*0 + ppa5*d5/100
printf("\n Producers risk = %0.2f percent\n
    Consumers risk = %0.3f percent", alpha,bita)
x = [1,2,4,6,8]
y = [0.91,1.4716,1.624,1.194,0.733]
plot(x,y)
xtitle("AOQ curve","Percent dectives" , "AOQ of lot"
    )
```

Chapter 22

Kinematics of machine tools

Scilab code Exa 22.1 Find range of cutting velocity

```
clc
d1 = 10 // min. dia of cutter in mm
d2 = 60 // max. dia of cutter in mm
v = 30e3 // operating speed in m/min
n1 = v / (%pi * d2) // n_min in rpm
n2 = v / (%pi * d1) // n_max in rpm
phi = (n2 / n1)^(1/5)
spindle_speeds = zeros()
for i=0:5
    spindle_speeds(i+1) = phi^i * n1
end
cutter_dia = v ./ (%pi * spindle_speeds)
clf()
y = [0; v]
plot([0; cutter_dia(1)], y, [0; cutter_dia(2)], y,
   [0; cutter_dia(3)], y, [0; cutter_dia(4)], y, [0;
    cutter_dia(5)], y, [0; cutter_dia(6)], y)
xtitle("","cutter diameter mm","cutting velocity, m/
   min")
// from graph
vmax1 = 36 // m/min
```

```
vmin1 = 24.5 // m/min
r1 = vmax1 - vmin1 // Range of cutting speed for 12 mm diameter in m/min
vmax2 = 36.5 // m/min.
vmin2 = 26 // m/min.
r2 = vmax2 - vmin2 // Range of cutting speed for 36 mm diameter in m/min
printf("\n Range of cutting speed for 12 mm diameter = %0.1f m/min.\n Range of cutting speed for 36 mm diameter = %0.1f m/min." , r1 , r2)
```

Scilab code Exa 22.2 Determine speed ratios and teeth

```
clc
m = 2.5 // module in mm
phi = 1.2 // common ratio
n = 150 // speed in rev/min. of driving shaft
n1 = 70 // speed in rev/min. of driven shaft
n2 = (phi)^1*n1 // speed in rev/min. of driven shaft
n3 = (phi)^2*n1 // speed in rev/min. of driven shaft
n4 = (phi)^3*n1 // speed in rev/min. of driven shaft
T1=poly(0,'T1')
t1=n1/n*T1
T1=roots(t1+T1-80)
t1=horner(t1,T1)
T2=poly(0,'T2')
t2=n2/n*T2
T2=roots(t2+T2-80)
t2=horner(t2,T2)
T3=poly(0,'T3')
t3=n3/n*T3
T3=roots(t3+T3-80)
t3=horner(t3,T3)
T4=poly(0,'T4')
t4=n4/n*T4
```

```
T4=roots(t4+T4-80)
t4=horner(t4,T4)
t1 = floor(t1) // number of teeth on driving shaft
T1 = ceil(T1) // number of teeth on driven shaft
t2 = ceil(t2) // number of teeth on driving shaft
T2 = floor(T2) // number of teeth on driven shaft
t3 = floor(t3) // number of teeth on driving shaft
T3 = ceil(T3) // number of teeth on driven shaft
t4 = ceil(t4) // number of teeth on driving shaft
T4 = floor(T4) // number of teeth on driven shaft
 // running speeds
n1 = n*t1/T1
n2 = n*t2/T2
n3 = n*t3/T3
n4 = n*t4/T4
printf("\n Number of teeth on driver and driven are
    :- \n t1 = %d ,T1 = %d\n t2 = %d ,T2 = %d \n t3 =
     %d ,T3 = %d \n t4 = %d ,T4 = %d ",t1,T1,t2,T2,t3
    ,T3,t4,T4)
printf("\n The actual running speed of driven shaft
    will be : \n n1 = %0.2f rev/min\n n2 = %0.2f rev/
    min \n n3 = %0.2f rev/min \n n4 = %0.2f rev/min",
    n1,n2,n3,n4)
// Answer of n3 is given wrong in book
// Answer vary due to round off error
```

Scilab code Exa 22.3 Calculate speed and number of teeths

```
clc
z = 6 // number of steps
n1 = 180 // rev/min
n2 = 100 // rev/min
Rn = n1/n2
phi = (Rn)^(1/(z-1)) // common ratio
n3 = phi*n2 // rev/min
```

```
n4 = (phi)^2*n2 // rev/min
n5 = (phi)^3*n2 // rev/min
n6 = (phi)^4*n2 // rev/min
n7 = 225 // speed of input shaft in rev/min
Ta=poly(0,'Ta')
tb=n7/n5*Ta
Ta=roots(tb+Ta-52)
tb=horner(tb,Ta)
tb = ceil(tb)
Tc=poly(0,'Tc')
td=n7/n6*Tc
Tc=roots(td+Tc-52)
td=horner(td,Tc)
Tc = ceil(Tc)
Te=poly(0,'Te')
tf=n7/n1*Te
Te=roots(tf+Te-52)
tf=horner(tf,Te)
tf = ceil(tf)
Th=poly(0,'Th')
tj=n2/n5*Th
Th=roots(tj+Th-46)
Th = ceil(Th)
tj=horner(tj,Th)
tj = floor(tj)
Ti=poly(0,'Ti')
tg=n5/n5*Ti
Ti=roots(tg+Ti-46)
tg=horner(tg,Ti)
printf("\n Ta = %d Tb = %d \n Tc = %d Td = %d \n Te
    = %d tf = %d \n Th = %d Tj = %d \n Ti = %d Tg =
    %d" , Ta,tb,Tc,td,Te,tf,tj,Th,Ti,tg)
// 'Answers vary due to round off error'
```

Scilab code Exa 22.4 Calculate common ratio

```
clc
v = 21 // cutting speed in rev/min.
z = 6
dmin = 5 // daimeter in mm
dmax = 20 // daimeter in mm
nmax = 1000*v/(%pi*dmin) // spindle speed in rev/min
    .
nmin = 1000*v/(%pi*dmax) // spindle speed in rev/min
    .
phi = (nmax/nmin)^(1/(z-1)) // common ratio
n1 = nmin // rev/min.
n2 = phi*n1 // rev/min.
n3 = (phi)^2*n1 // rev/min.
n4 = (phi)^3*n1 // rev/min.
n5 = (phi)^4*n1 // rev/min.
n6 = (phi)^5*n1 // rev/min.
printf("\n Common ratio = %0.2f \n Spindle speeds
    are : %0.2f , %0.1f , %0.2f , %0.2f ,%0.2f and %0
    .1f rev/min.",phi,n1,n2,n3,n4,n5,n6)
// 'Answers vary due to round off error'
```

Scilab code Exa 22.5 Calculate gear ratio teeth and speed

```
// from fig. 22.18A
clc
// Three gear ratios between input and intermediate
    shaft
nmax = 1400 // maximum speed in rev/min.
i1 = 1/1
i2 = 1/1.26
i3 = 1/(1.26)^2
// The two ratios between intermediate and output
    shaft
i4 = 1/1
i5 = 1/(1.26)^3
```

```
// number of teeth for input and intermediate shaft
t1 = 27/27
t2 = 24/30
t3 = 21/33
//  number of teeth for output and intermediate shaft
t4 = 34/34
t5 = 20/48
// output speeds in rev./min
n1 = t3*t5*nmax
n2 = t2*t5*nmax
n3 = t1*t5*nmax
n4 = t3*t4*nmax
n5 = t2*t4*nmax
n6 = t1*t4*nmax
printf("\n Three gear ratios between input and intermediate shaft i1 = %d i2 = %0.2f i3 = %0.3f \n  The two ratios between intermediate and output shaft i4 = %d i5 = %0.3f \n number of teeth for each pair between input and intermediate shaft t1 = 27/27 ,t2 = 24/30 , t3 = 21/33 \n number of teeth for each pair between output and intermediate shaft = t4 = 34/34 ,t5 = 20/48 \n Output speeds \n n1 = %d  rev/min , n2 = %d rev/min , n3 = %d rev/min \n n4 = %d rev/min, n5 = %d rev/min , n6 = %d rev/min" ,i1 , i2 , i3 , i4 , i5 , n1 , n2, n3,n4,n5,n6)
// Answer vary due to round off error
```

Chapter 23

Production planning and control

Scilab code Exa 23.1 Calculate forecast

```
clc
d1 = 90 // demand for first quarter
d2 = 100 // demand for second quarter
d3 = 80 // demand for third quarter
sa = (d1+d2+d3)/3 // simple average
printf("\n Forecast = %d" , sa)
```

Scilab code Exa 23.2 Calculate forecat by SMA method

```
clc
d1 = 300 // demand for july
d2 = 350 // demand for august
d3 = 400 // demand for september
d4 = 500 // demand for october
d5 = 600 // demand for november
d6 = 700 // demand for december
```

```
// assuming n = 3 , where n is number of time period
forecast = (d6+d5+d4)/3 // forecast
printf("\n Forecast = %d" , forecast)
```

Scilab code Exa 23.3 Calculate forecat by WMA method

```
clc
d1 = 500 // demand for october
d2 = 600 // demand for november
d3 = 700 // demand for december
w1 = 0.25 // relative weight with december
w2 = 0.25 // relative weight with november
w3 = 0.5 // relative weight with october
f = w1*d1 + w2*d2 + w3*d3 // forecast
printf("\n Forecast by weighted moving average = %d"
    , f)
```

Scilab code Exa 23.4 Calculate forecast for january

```
clc
alpha = 0.7 // smoothing coefficient
d1 = 250 // demand for november
d2 = 300 // demand for december
f1 = 200 // forecast for november
f2 = alpha*d1 + (1-alpha)*f1 // forecast for
    december
f3 = alpha*d2 + (1-alpha)*f2 // forecast for january
f3 = ceil(f3)
printf("\n Forecast for january = %d units" , f3)
```

Scilab code Exa 23.5 Calculate total cost

```
clc
s = 600 // set up cost per lot in Rs
c = 6 // unit cost of item in Rs
a = 100000 // annual demand for item
i = 25 //  annual carrying charges of average
    inventory
i = 25/100
k = c*i // carrying cost factor in unit/year
n = sqrt(2*s*a/k) // most economic lot size
tc = a*c + s*a/n + k*n/2 // total cost in Rs
printf("\n Total cost = Rs %0.2f" , tc)
//  'Answers vary due to round off error'
```

Scilab code Exa 23.6 Calculate economical order quantity

```
clc
a = 8000 // annual requirement of parts
c = 60 // unit cost of part in Rs
r = 150 // ordering cost per lot in Rs
i = 30 // annual carrying charges of average
    inventory
i = 30/100
k = i*c // carrying cost per unit per year
n = sqrt(2*r*a/k) // most economical order quantity
printf("\n Most economical ordering quantity = %d
    units" , n)
```

Scilab code Exa 23.7 Calculate economic lot size

```
clc
a = 12000 // annual requirement
```

```
c = 5 // unit cost of part
s = 60 // set up cost per lot
p = 18750 // production rate per year
i = 20 // inventory carrying cost
i = 20/100
k = i*c // carrying cost per unit per year
n = sqrt(2*s/(1/a-1/p)*k) // Most economic lot size
printf("\n Most economic lot size = %d parts" , n)
```

Scilab code Exa 23.8 Calculate inventory control terms

```
clc
a = 15625 // annual requirement of parts
c = 12 // unit cost of part in Rs
r = 60 // ordering cost per lot in Rs
k = 1.2 // inventory carrying cost per unit
n = sqrt(2*r*a/k) // economical order quantity
oc = r*a/n // ordering cost in Rs
cc = k*n/2 // carrying cost in Rs
tc = oc + cc // total inventory cost in Rs
printf("\n Economical order quantity = %d units\n
   order cost = Rs %d\n carrying cost = Rs %d\n
   Total inventory cost = Rs %d" , n , oc , cc , tc)
```

Scilab code Exa 23.9 Calculate discount offered

```
clc
// case a
a = 50 // annual requirement of parts in tonnes
c = 500 // unit cost of part in Rs
r = 100 // ordering cost per order in Rs
i = 20 // inventory carrying cost
i = i/100
```

```
d = 2 // discount of purchase cost in percent
k = i*c // inventory carrying cost per unit
n1 = sqrt(2*r*a/k) // economical order quantity
oc1 = r*a/n1 // ordering cost in Rs
cc1 = k*n1/2 // carrying cost in Rs
tc1 = oc1 + cc1 // total inventory cost in Rs
// case b
n2 = 25 // order per lot
oc2 = r*a/n2 // ordering cost in Rs
cc2 = k*n2/2 // carrying cost in Rs
tc2 = oc2 + cc2 // total inventory cost in Rs
i = tc2-tc1 // increase in cost in Rs
d_o = d*c*a/100 // discount offered
printf("\n Increase in inventory cost = Rs %d\n Discount offered = Rs%d",i,d_o)
disp("offer is worth accepting")
```

Scilab code Exa 23.10 Calculate EOQ and reorder point

```
clc
a = 1000000 // annual requirement of parts
r = 32 // ordering cost per lot in Rs
k = 4 // inventory carrying cost per unit
d1 = 250 // number of working days
d2 = 2 // days for safety stock
d3 = 4 // lead time in days
eoq = sqrt(2*r*a/k) // economical order quantity
oc = r*a/eoq // ordering cost in Rs
cc = k*eoq/2 // carrying cost in Rs
tc = oc + cc // total inventory cost in Rs
ss = a*d2/d1 // safety stock
ro_p = ss+eoq*d3 // reorder point
printf("\n Economic order qunantity = %d components\n Re-order point = %d components" , eoq ,ro_p)
```

Chapter 26

Plant layout

Scilab code Exa 26.1 Calculate number of machine required

```
clc
N = 100000 // annual output of parts
s = 2 // expected scrap
t = 105 // estimated time per part in s
ita = 80 // production efficiency of machine
a = 2300 // number of working hours
output = (3600*ita)/(t*100) // parts required per
    hour
pr = N*(100+s)/(a*100) // output from one machine
    per hour
mr = pr/output // machines required
printf("\n Number of machines required = %0.2f" , mr
    )
disp("If machine is to be used exclusively for part
    considered two machines required")
```