# Scilab Textbook Companion for Vector Mechanics for Engineers: Stastics And Dynamics
by F. P. Beer, E. R. Johnston, D. F. Mazurek, P. J. Cornwell And E. R. Eisenberg[1]

Created by
Akshatha Nayak
M.Tech
Electrical Engineering
Bits Pilani
College Teacher
None
Cross-Checked by
Lavitha

June 16, 2015

[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:** Vector Mechanics for Engineers: Stastics And Dynamics

**Author:** F. P. Beer, E. R. Johnston, D. F. Mazurek, P. J. Cornwell And E. R. Eisenberg

**Publisher:** McGraw-Hill, NY

**Edition:** 9

**Year:** 2007

**ISBN:** 9780073529400

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

# Statics of particle

**Scilab code Exa 2.1** Determine the resultant

```
clc;
//Page 22
//Getting resultant of two vectors

P=40; // N   Magnitude of vector P
Q=60 // N Magnitude of vector Q
// imagine triangle for triangle law of vectors
B=180-25;// degree , Angle between vector P and vector Q

//R- Resultant vector
B=B*%pi/180;// conversion into radian
//R^2=P^2+Q^2-2*P*Q*cos(B);     Cosine Law
R=sqrt(P^2+Q^2-2*P*Q*cos(B));// N

printf("Maginitude of Resultant is R= %.2f N\n",R);


//A- Angle between Resultant and P vector , Unknown

// sin(A)/Q == sin(B)/R  sine law
```

```
A=asin(Q*sin(B)/R);// radian


A=A*180/%pi;//// Conversion into degree

alpha=A+20;// degree
printf("Angle of Resultant  vector R with x axis is
   %.2f Degrees\n",alpha);
```

**Scilab code Exa 2.2** Tension in ropw

```
clc;
//Page 29
R=25; // kN   Magnitude of Resultant vector
alpha=45;//degree
// T1 and T2 are tensions in rope 1 and rope 2
   respectively
A=30;// degree , Angle between vector T1 and
   resultant
B=alpha;// degree , Angle between vector T2 and
   resultant
C=180-(A+B);// degree , Angle between vector T1 and
   T2


// conversion of angles into radian
A=A*%pi/180;
B=B*%pi/180;
C=C*%pi/180;


// sin(A)/T2 == sin(B)/T1 == sin(C)/R ..............
    sine law

```

```
T1=(R*sin(B))/sin(C);//kN
T2=(R*sin(A))/sin(C);//kN


printf("Tension in rope 1 is T1=%.2f kN and in rope 2 is T2=%.2f kN \n",T1,T2);


// Minimum value of T2 occcurs when T1 and T2 are perpendicular to each other i.e C=90 degree
C=90;//degree
A=30;// degree
B=180-(A+C);//degrees
alpha=B;//degrees
B=B*%pi/180;// radian
T2=R*sin(B);// kN
T1=R*cos(B);//kN
printf("Minimum tension in rope 2 is T2=%.2f kN \n",T2);
printf("corrosponding T1=%.2f kN \n ",T1);
printf("alpha=%.2f degrees",alpha);
```

**Scilab code Exa 2.3** Resultant of forces

```
clc;
//page 31
F1=150;// N
F2=80;// N
F3=110;//N
F4=100// in N

F1x=129//in N
F2x=-27.4
F3x=0
F4x=96.6
```

```
F1y=75
F2y=75.2
F3y=-110
F4y=-25.9

Rx=F1x+F2x+F3x+F4x;//N Horizontal component of R-resultant
Ry=F1y+F2y+F3y+F4y;//N Vertical component of R-resultant

//R=Rx i +Ry j

printf("R= %.2f i + %.2f j \n", Rx,Ry);

alpha=atan(Ry/Rx);//Radian, Angle made by resultant with +ve x axis
alpha=alpha*180/%pi;//Conversion into degrees

R=sqrt(Rx^2+Ry^2);// N , Magnitude of resultant
printf("alpha= %.2f degrees and R= %.2f N",alpha,R);
```

**Scilab code Exa 2.4** Tension of Tab and Tac

```
clc;
//page 38
W=3500; // lb   weight of automobile
alpha=2;//degree
// TAB and TAC are tensions in cable AB and cable AC respectively
A=90+30;// degree , Angle between vector T1 and resultant
B=alpha;// degree , Angle between vector T2 and resultant
C=180-(A+B);// degree , Angle between vector T1 and T2
```

```


// conversion of angles into radian
A=A*%pi/180;
B=B*%pi/180;
C=C*%pi/180;


// sin(A)/TAB == sin(B)/TAC == sin(C)/W
   .............. sine law


TAB=(W*sin(A))/sin(C);//N
TAC=(W*sin(B))/sin(C);//N

printf("Tension in cable AB is TAB=%.2f lb and in
   Cable AC is TAC=%.2f lb \n",TAB,TAC);
```

**Scilab code Exa 2.5** Force

```
clc
//page 39
mass=30;// kg
W=mass*9.81;// N, Weight of package
alpha=15;//degree
alpha=alpha*%pi/180;// Conversion into radian
F=W*sin(alpha);//N
printf("F= %.2f N",F);
```

**Scilab code Exa 2.6** Drag force

```
clc;
//page 39
```

```
alpha=atan(7/4);//rad
beta=atan(1.5/4);//rad
T_AB=200;//N tension in cable AB
T_AE=-300;//N, tension in cable AE
// R= T_AB+T_AC+T_AE+F_D=0          ...Equillibrium
   Condition...........1


T_ABx=-T_AB*sin(alpha);// Xcomponent of T_AB
T_ABy=T_AB*cos(alpha);//Y component of T_AB

// T_ACx=T_AC*sin(beta); Xcomponent of T_AC
// T_ACy=T_AC*cos(beta); Y component of T_AC

// Sum Fx =0 gives -T_AB*sin(alpha) N + T_AC*sin(
   beta) +F_D = 0..........2
//Sum Fy=0 gives T_AB*cos(alpha) N +T_AC*cos(beta) +
   T_AE  = 0................3

T_AC=(-T_AB*cos(alpha)-T_AE)/cos(beta);//N, From 3

F_D=T_AB*sin(alpha)-T_AC*sin(beta);//N, From 2

printf("Value of drag force is F_D=%.2f N and
   tension in cable AC is T_AC= %.2f N",F_D,T_AC);
```

**Scilab code Exa 2.7** Resultant force on AB and Ac

```
//page 50
clc;
dx=-40;//m
dy=80;//m
dz=30;//m
f=2500;//N, Mafnitude of force F
d=sqrt(dx^2+dy^2+dz^2);//m, total distance of vector
```

```
AB
//F=f*lambda, lambda - unit vector= AB/d. So we can calculate each component by multiplying this unit vector
Fx=f*dx/d;//N , X component of F
Fy=f*dy/d;//N , Y component of F
Fz=f*dz/d;//N , Z component of F

printf("Component of F along X axis is %.2f N\n",Fx);
printf("Component of F along Y axis is %.2f N\n",Fy);
printf("Component of F along Z axis is %.2f N\n",Fz);
printf("We may write F as \n F = %.2f i + %.2f j + %.2f k\n",Fx,Fy,Fz);

thetax=acos(Fx/f);// radian , angle with +ve x axis
thetay=acos(Fy/f);// radian , angle with +ve y axis
thetaz=acos(Fz/f);// radian , angle with +ve z axis

//Conversion of angles into degree
thetax=thetax*180/%pi;//degree
thetay=thetay*180/%pi;//degree
thetaz=thetaz*180/%pi;//degree

printf("Angle made by F with +ve X axis %.2f degree\n",thetax);

printf("Angle made by F with +ve Y axis %.2f degree\n",thetay);
printf("Angle made by F with +ve Z axis %.2f degree\n",thetaz);
printf("\n\n")
F=800 // N , given force

theta=145 // Degrees , angle with posiyive X axis

```

```
theta=theta*%pi/180;// Conversion into radian



Fx=F*sin(theta);//N, Horizontal component

Fy=F*cos(theta);// N, Vertical Component


printf("\n\n")
printf("Horizontal component of F is %.2f N\n",Fx);

printf("Vertial component of F is %.2f N\n",Fy);

printf("We may write F as \n F = %.2f i + %.2f j",Fx
    ,Fy);

F=300 // N , given force

AB=sqrt(8^2+6^2);// m Length of AB

cos_alpha=8/AB;

sin_alpha=-6/AB;

Fx=F*cos_alpha;//N, Horizontal component

Fy=F*sin_alpha;// N, Vertical Component
```

**Scilab code Exa 2.8** resultant of AB and AC

```
clc;
//page 51
```

```
T_AB=4200;//N , Tension in cable AB
T_AC=6000;//N , Tension in cable AC
// Vector AB=-(5m)i+(3m)j+(4m)k
//Vector Ac= -(5m)i+(3m)j+(5m)k
ABx=-5;//m
ABy=3;//m
ABz=4;//m
ACx=-5;//m
ACy=3;//m
ACz=-5;//m

AB=sqrt((-5)^2+3^2+4^2);//m, Magnitude of vector AB
AC=sqrt((-5)^2+3^2+5^2);//m, Magnitude of vector AC
//vT_AB=T_AB*lambdaAB, lambdaAB - unit vector= vAB/
    AB. So we can calculate each component by
    multiplying this unit vector
T_ABx=T_AB*ABx/AB;//N , X component of T_AB
T_ABy=T_AB*ABy/AB;//N , Y component of T_AB
T_ABz=T_AB*ABz/AB;//N , Z component of T_AB

printf("Component of T_AB along X axis is %.2f N\n",
    T_ABx);
printf("Component of T_AB along Y axis is %.2f N\n",
    T_ABy);
printf("Component of T_AB along Z axis is %.2f N\n",
    T_ABz);
printf("We may write T_AB as \n T_AB = %.2f i + %.2f
     j + %.2f k\n",T_ABx,T_ABy,T_ABz);


//vT_AC=T_AC*lambdaAC, lambdaAC - unit vector= vAC/
    AC. So we can calculate each component by
    multiplying this unit vector
T_ACx=T_AC*ACx/AC;//N , X component of T_AC
T_ACy=T_AC*ACy/AC;//N , Y component of T_AC
T_ACz=T_AC*ACz/AC;//N , Z component of T_AC

printf("Component of T_AC along X axis is %.2f N\n",
```

```
T_ACx);
printf("Component of T_AC along Y axis is %.2f N\n",T_ACy);
printf("Component of T_AC along Z axis is %.2f N\n",T_ACz);
printf("We may write T_AC as \n T_AC = %.2f i + %.2f j + %.2f k\n",T_ACx,T_ACy,T_ACz);

Rx=T_ABx+T_ACx;//N ,X component of R
Ry=T_ABy+T_ACy;//N ,Y component of R
Rz=T_ABz+T_ACz;//N ,Z component of R

printf("Component of R along X axis is %.2f N\n",Rx);
printf("Component of R along Y axis is %.2f N\n",Ry);
printf("Component of R along Z axis is %.2f N\n",Rz);
printf("We may write R as \n R = %.2f i + %.2f j + %.2f k\n",Rx,Ry,Rz);

R=sqrt(Rx^2+Ry^2+Rz^2);//N, Magnitude of resultant

thetax=acos(Rx/R);// radian , angle with +ve x axis
thetay=acos(Ry/R);// radian , angle with +ve y axis
thetaz=acos(Rz/R);// radian , angle with +ve z axis

//Conversion of angles into degree
thetax=thetax*180/%pi;//degree
thetay=thetay*180/%pi;//degree
thetaz=thetaz*180/%pi;//degree

printf("Angle made by R with +ve X axis %.2f degree\n",thetax);

printf("Angle made by R with +ve Y axis %.2f degree\n",thetay);
printf("Angle made by F with +ve Z axis %.2f degree\
```

```
n",thetaz);
```

# Chapter 3

# Rigid bodies equivalent systems of forces

**Scilab code Exa 3.1** Vertical force

```
clc;
// Given data
//page 85
F=100; // lb , Vertical force applied to end of
    lever
theta=60;// degree , angle made by lever with +ve X
    axis
l=24; //  , length of lever

// a ) Momemt about O
d=l*cosd(theta);// mm ,perpendicular distance from
    o to the line of action

Mo=F*d;// N.m, Magnitude of moment about O
printf("Magnitude of moment about O of the 500 N is
    %d lb.in and it is in clockwise direction as
    force tends to rotate lever clockwise\n",Mo);

// b) Horizontal force
```

```
d=l*sind(theta);//in, perpendicular distance from  o
    to the line of action

F=Mo/d;// N, Horizontal Force at A required to
   produce same Moment about O
printf("Magnitude of Horizontal Force at A required
   to produce same Moment about O is %f lb \n",F);

// c)Smallest force

// F is smaller when d is maximum in expression Mo=F
   *d, so we choose force perpendicular to OA
Mo=1200//in lb
d=24// in ,perpendicular distance from  o to the
   line of action
F=Mo/d;// N, Smallest Force at A required to produce
    same Moment about O
printf("Magnitude of smallest Force at A required to
    produce same Moment about O is %f lb \n",F);

//d) 1200 N vertical force
Mo=1200;// lb-in,
F=240//in lb
d=Mo/F;// m, perpendicular distance from  o to the
   line of action of force
OB=d/cosd(theta);//m, distance of point B from O

printf("Verical force of 1200 N must act at %f in
   far from the shaft to create same moment about O\
   n",OB);
```

**Scilab code Exa 3.2** Moment of force

```
clc;
```

```
//Page 86
// Given data
F=800; // N , Force applied on bracket
theta=60;// degree, angle made by lever with +ve X
    axis
theta=theta*%pi/180;// Conversion of angle into
    radian
r_AB=[-0.2, 0.16];//m vector drawn from B to A
    resolved in rectangular component
F=[F*cos(theta), F*sin(theta)]//N , vector F
    resolved in rectangular component
k=1;// Unit vector along Z axis

// M_B=r_AB * F relation 3.7 from section 3.5
M_B=det([r_AB; F])*k;// N.m
printf("The moment of force 800 N about B is %.2f N.
    m . -ve sign shows its acting clockwise\n",M_B);
```

**Scilab code Exa 3.3** Moment of force

```
clc;
//page 86
// Given data
P=30; // lb, Force applied to shift lever
alpha=20;// degree, angle made by force P with -ve X
    axis
Q=P*sind(alpha)//in degree

d=3//in ft
M_o=Q*d//N.m , here negative signs are taken as each
    component creates moment clockwise
printf("The moment of force P about B is %.2f lb-ft
    . -ve sign \n shows its acting clockwise\n",M_o);
```

**Scilab code Exa 3.4** magnitude of force and lambda

```
clc;
//page 87
// Given data
// M_A=r_CA * F relation 3.7 from section 3.5
f=200; // N , Magnitude of Force directed along CD
r_CA=[0.3,0, 0.08];//m, vector AC reprecsented in rectangular component
//lambda=CD/norm(CD)-m, Unit vector along CD
//F=f*lambda;//m, Force
CD=[-0.3, 0.24, -0.32];//Vector CD resolved into rectangular component
// norm(CD); m, magnitude of vector CD

lambda=CD/norm(CD);//m, Unit vector along CD
F=f*lambda;//m, Force
// M_A=r_CA * F relation 3.7 from section 3.5
//i=1; j=1; k=1; Unit vectors along X, Y and Z direction respectively

// Componenets of moment M_A along X,Y and Z direction respectively
M_Ax=det([r_CA(2),r_CA(3); F(2), F(3)]);//N.m
M_Ay=-det([r_CA(1),r_CA(3) ; F(1),F(3)]);//N.m
M_Az=det([r_CA(1),r_CA(2) ;F(1), F(2)]);// N.m

printf("Answer can be written as M_B = %.2f N.m i + %.2f N.m j + %.2f N.m k \n",M_Ax,M_Ay,M_Az);
```

**Scilab code Exa 3.6** Couple M equivalent to two couple

```
clc;
//Given data
// Moment arms
Fx=-30;//in lb
Fy=20;//in lb
Fz=20;//in lb

//couple Forces
x=18;//iN
y=12;//iN
z=9;//iN

Mx=Fx*x;//N.m, Component of Moment along X axis
My=Fy*y;//N.m, Component of Moment along Y axis
Mz=Fz*z;//N.m, Component of Moment along Z axis
//This three moments represent component of single
    couple M
printf("Couple M equivalent to two couple can be
    written as \n M = %.2f lb-in i + %.2f lb-in j + %
    .2f lb-in k \n",Mx,My,Mz);
```

**Scilab code Exa 3.7** Distance from the shaft

```
clc;
//page 113
Mo=24;//N.m *k, Couple of moment
f=-400;//N, Magnitude of force
OB=300;//mm, Distance of force from point O
theta=60;// degree, angle made by lever with +ve X
    axis
x=cosd(theta)
BC=Mo/(-f*x);//m
BC=BC*1000;//mm, Conversion into millimeter
disp(BC)
OC=OB+BC;//mm, Distance from the shaft to the point
```

```
of application of this equivalenet force

printf("Distance from the shaft to the point of
   application of this equivalenet single force is
   %f mm",OC)
```

# Chapter 4

# Equilibrium of rigid bodies

**Scilab code Exa 4.1** Angle and degee

```
clc;
//page 166
//Determination of B
//At equillibrium +sum(M_A)=0
//B*1.5m-(9.81kN)(2 m)-(23.5 kN)(6 m)=0, B assumed to be in +ve X direction
B=(9.81*2+23.5*6)/1.5//kN
printf("B=%.2f kN \n +ve sign shows reaction is directed as assumed ",B);
//Determination of Ax
//Sum Fx=0
//Ax+B=0
Ax=-B;//kN
printf("Ax=%.2f kN\n",Ax);
//Determination of Ay
//Sum Fy=0
//Ay-9.81 kN-23.5kN=0
Ay=9.81+23.5;//kN
printf("Ay=%.2f kN\n",Ay);
A=[Ax,Ay];//kN Adding component
A=norm(A);//Magnitude of force A
```

```
theta=atan(Ay/Ax);//radians
theta=theta*180/%pi;//degrees , conversion into
    degrees
printf("Reaction at A is A=%.2f kN making angle %.2f
     degrees \n with + ve x axis ",A,theta);
//Slight variation in the answer because of roundoff
     error
```

**Scilab code Exa 4.2** Angle and degee

```
clc;
//Page 148

//At equillibrium equations are +-> sum Fx=0, +sum(
    M_A)=0, +sum(M_B)=0
//Sum Fx=0 gives
Bx=0;//kN
printf("Bx=%.0f kN \n",Bx);
//+sum(M_A)=0 gives -(70kN)(0.9m)+By(2.7m)-(27kN)
    (3.3m)-(27kN)(3.9m)=0, B assumed to be in +ve Y
    direction
By=(70*0.9+27*3.3+27*3.9)/2.7//kN
printf("By=%.2f kN  +ve sign shows reaction is
    directed as assumed \n",By);

//+sum(M_B)=0 gives -A(2.7m)+(70kN)(1.8m)-(27kN)(0.6
    m)-(27kN)(1.2m)=0, A assumed to be in +ve Y
    direction
A=(70*1.8-27*0.6-27*1.2)/2.7//kN
printf("A=%.2f kN  +ve sign shows reaction is
    directed as assumed \n",A);
//Answer displayed in KN
```

**Scilab code Exa 4.3** Reaction and direction

```
clc;
//page 168
//Take x axis parallel to track and Y axis
    perpendicular to track
W=25;//kN
// Resolving weight
Wx=W*cos(25*%pi/180);//kN
Wy=-W*sin(25*%pi/180);//kN
//At equillibrium equations are +−> sum Fx=0, +sum(
    M_A)=0, +sum(M_B)=0

//+sum(M_A)=0 gives −(10.5kN)(625 mm)−(22.65 kN)(150
    mm)+ R2(1250 mm)=0, R2 assumed to be in +ve Y
    direction
R2=(10.5*625+22.65*150)/1250;//kN
printf("R2=%.0f kN +ve sign shows reaction is
    directed as assumed \n",R2);

//+sum(M_B)=0 gives (10.5kN)(625 mm)−(22.65 kN)(150
    mm)+ R1(1250 mm)=0, R1 assumed to be in +ve Y
    direction
R1=(10.5*625-22.65*150)/1250;//kN
printf("R1=%.1f kN +ve sign shows reaction is
    directed as assumed \n",R1);

//Sum Fx=0 gives, 22.65 N−T=0
T=22.65;//kN
printf("T=%.2f kN +ve sign shows reaction is
    directed as assumed \n",T);
```

**Scilab code Exa 4.4** Reaction and direction

```
clc;
```

```
////page 168
Ax=4.5//in m
Ay=6//in m
DF=sqrt((Ax^2)+(Ay^2))
F=150//in KN
Ex=-(Ax/DF)*F
printf("Ex=%.2f kN \n",Ex);
Ey=((Ay/DF)*F)+(4*20)
printf("Ey=%.2f kN \n",Ey);

M_E=-((20*7.2)+(20*5.4)+(20*3.6)+(20*1.8)-((Ay/DF)*F*Ax))
printf("M_E=%.0f kN  +ve sign shows reaction is directed as assumed \n",M_E);
```

**Scilab code Exa 4.5** Angle and degee

```
clc;
//page 169

//At equillibrium  +sum(Mo)=0,
//s=r*theta;
//F=k*s=k*r*theta;
k=45;//N/mm
r=75;//mm
W=1800;//N
l=200;//mm


// trial and error
printf("Probable answers by trial and error method are \n");
for i=0:0.1:%pi/2 // from 0 to 90 degrees

difference=(sin(i)-k*r^2*(i)/(W*l));
```

```
if difference<0.01 then   // Approximation
   theta=i;
   theta=theta*180/%pi;//Degrees , conversion into
      degrees
printf("Theta=%.2f degrees\n",theta);
end
end
```

**Scilab code Exa 4.6** Tension and angle

```
clc;
//page 185

m=10;//kg mass of joist
g=9.81;//m/s^2 gravitational acceleration
W=m*g;//N
AB=4;//m
// Three force body
BF=AB*cos(45*%pi/180);//m
AF=BF;//m

AE=1/2*AF;//m
EF=AE;//m
CD=AE;//m
BD=CD/tan((45+25)*%pi/180);//m
DF=BF-BD;//m
CE=DF;//m
alpha=atan(CE/AE);//radians
alpha=alpha*180/%pi;//degrees

//From geometry

G=90-alpha;//degrees
B=alpha-(90-(45+25));//degrees
C=180-(G+B);//Degrees
```

```

//Force triangle
//T/sin(G)=R/sin(C)=W/sin(B)..... sine law

T=W/sin(B*%pi/180)*sin(G*%pi/180);//N
R=W/sin(B*%pi/180)*sin(C*%pi/180);//N
printf("Tension in cable T= %.1f N\n Reaction At A
    is \n R= %.1f N with angle alpha= %.1f degrees
    with +ve X axis",T,R,alpha);
```

**Scilab code Exa 4.7** Reaction

```
clc;
//page 194
m1=80;//kg mass of man
m2=20;//kg, mass of ladder
m=m1+m2;//kg
g=9.81;//m/s^2 gravitational acceleration
W=-m*g;//N, j

C=-0.6*W/3;//N
Bz=-0.6*C/1.2;//N
By=-0.9*W/1.2;//N

printf(" Reaction At B is B= (%.0f) N j +(%.1f N)k\n
    ",By,Bz);
printf(" Reaction At C is C= (%.2f) N k\n",C);
Ay=-W-By;//N
Az=-C-Bz;//N


printf(" Reaction At A is A= (%.0f) N j +(%.1f N)k \
    n",Ay,Az);
```

**Scilab code Exa 4.8** Reaction and direction

```
clc;
W=-1200;//N,j   Weight
BD=[-2.4,1.2,-2.4];//m, Vector BD
EC=[-1.8,0.9,0.6];//m, Vector EC
//T_BD=norm(T_BD)*BD/norm(BD);// m, vector of
    tension in BD
//T_EC=norm(T_EC)*EC/norm(EC);// m, vector of
    tension in EC
// Applying equillibrium conditions we get
// Sum_F=0, and Sum(M_A)=0 and setting co-efficient
    equal to zero
A=[0.8,0.771;1.6,-0.514];//MAtrix of co-efficient
b=[-1440;0];//matrix b
x=linsolve(A,b);// solution matrix
T_BD=x(1);// N,Tension in BD
T_EC=x(2);//N, Tension in EC
printf("T_BD= (%.0f N) and T_EC= (%.0f N) \n",x(1),x
    (2));

Ax=2/3*T_BD+6/7*T_EC;//N, x component of reaction at
     A
Ay=-(1/3*T_BD+3/7*T_EC+W);//N, Y component of
    rection at A
Az=2/3*T_BD-2/7*T_EC;//N, z component of reaction at
     A

printf("Reaction at A is A=(%.0f N)i +(%.0f N)j +(%
    .1f N)k \n",Ax,Ay,Az);
//Answe in Newton instead of lbs
//1lbs=4.44N
```

**Scilab code Exa 4.9** Tension in vector form

```
clc;
//page 198
//Free body diagram
m=30//in kg
g=9.81//in m/s2
w=-m*g//in J
DC=[-480 240 -160]//in mm
X=norm(DC)
T=DC/X
disp("Tension in the vector form=")
disp(T)
//Equilibrium equations
//From equation 2, setting unit vector=0
Ax=49//in N
Ay=73.5//in N
A=[Ax Ay]
y=norm(A)
disp("Tension in the vector form in N=")
disp(y)
```

**Scilab code Exa 4.10** coordinates

```
clc;
//page 197
AC=[12 12 0]
w=[0;-450;0]
x1=AC*w
disp(x1)
x=[0 0 x1]
lambda=[2/3 2/3 -1/3]*[0;0;-x1]
```

```
y=x*lambda
disp(y)

//Location of G
//EG and Tmin are having same direction , so their
    component should be in proportion
x=-1.8/Tmin(3)*Tmin(1)+1.8;//m, X co-ordinate of G
y=-1.8/Tmin(3)*Tmin(2)+3.6;//m, Y co-ordinate of G
printf("Co-ordinates of G are x=%.0f m and y= %.1f m
    ",x,y);
```

# Chapter 5

# Distrubuted forces centroids and centers of gravity

**Scilab code Exa 5.1** centroid

```
clc;
//page 228
n=4; // no of component
A=[120*80,120*60/2,%pi*60*60/2,-%pi*40*40];//mm^2,
    Areas of Rectangle, triangle, Semicircle, and
    Circle respectively
x=[60,40,60,60];//mm, x components of centroids of
    Rectangle, triangle, Semicircle, and Circle
    respectively
y=[40,-20,105.46,80];//mm, y components of centroids
     of Rectangle, triangle, Semicircle, and Circle
    respectively

sumA=0;
sumxA=0;
sumyA=0;

for(i=1:n)
    sumA=sumA+A(i);
```

```
    sumxA=sumxA+x(i)*A(i);
    sumyA=sumyA+y(i)*A(i);

end

// First Moment of area
Qx=sumyA;// About X axis
Qy=sumxA;//About Yaxis
printf("First moments of the area are Qx= %.0f mm^3
   and Qy=%.0f mm^3 \n",Qx,Qy);

//Location of centroid
X=sumxA/sumA;// X co-ordinate
Y=sumyA/sumA;// Y co=ordinate
printf("Co-ordinates of centroid are X= %.1f mm and
   Y= %.1f mm \n",X,Y);
```

**Scilab code Exa 5.2** Coordinates of centroid

```
clc;
//page 229
n=3; // no of segment
L=[600,650,250];//mm, Lengths of segment AB , BC and
    CA respectively
x=[300,300,0];//mm, x components of centroids of
   segment AB , BC and CA respectively
y=[0,125,125];//mm, y components of centroids of
   segment AB , BC and CA respectively

sumL=0;
sumxL=0;
sumyL=0;

for(i=1:n)
    sumL=sumL+L(i);
```

```
    sumxL=sumxL+x(i)*L(i);
    sumyL=sumyL+y(i)*L(i);

end



//Location of centre of gravity
X=sumxL/sumL;// X co-ordinate
Y=sumyL/sumL;// Y co=ordinate
printf("Co-ordinates of centroid are X= %.0f mm and
    Y= %.0f mm \n",X,Y);
//There is variation because of roundoff
```

**Scilab code Exa 5.7** Mass of steel

```
clc;
//page 242
p=7850;//kg/m^3, density of steel rim
n=2; // no of component
A=[(20+60+20)*(30+20),-60*30];//mm^2,Cross section
    Areas of rectangle I and II

y=[375,365];//mm, y components of centroids of
    Rectangles I and II respectively


sumV=0;

for(i=1:n)
    C(i)=2*%pi*y(i);//mm, Distance travelled by C
    V(i)=A(i)*C(i);//mm^3, Volume of 1 component
    sumV=sumV+V(i);// mm^3, Total volume of rim

end
```

```
sumV=sumV*10^(-9);//Conversion into m^3
g=9.81;//m/s^2, acceleration due to gravity
m=p*sumV;//kg, mass
W=m*g;//N, Weight
printf("mass of steel is m= %.0f kg and Wight is W= %.0f N\n",m,W);
```

**Scilab code Exa 5.9** equivalent concentrated mass

```
clc;
//page 250
n=2; // no of triangle
A=[4.5,13.5];//kN, loads
x=[2,4];//mm, distances of centroid from point A

sumA=0;
sumxA=0;
for(i=1:n)
    sumA=sumA+A(i);
    sumxA=sumxA+x(i)*A(i);

end


//Location of centroid
X=sumxA/sumA;// X co-ordinate
W=sumA;//kN, Concentrated load
printf("The equivalent concentrated mass is W= %.0f kN and its line of action is located at a distance X= %.1f m to the right of A \n",W,X);

// Reactions
// Applying sum(F_x)=0
Bx=0;//N
//Applying sum(M_A)=0
```

```
By=W*X/6;//kN, Reaction at B in Y direction
//Applying sum(M_B)=0
A=W*(6-X)/6;//kN, Reaction at B in Y direction

printf("The rection at A=%.1f kN, At Bx=%.1f kN and
    By=%.1f kN \n",A,Bx,By);
```

**Scilab code Exa 5.10** Reaction and direction

```
clc;
//page 251
t=0.3;//m thickness of dam
g=9.81;// m/s^2, acceleration due to gravity
p1=2400;//kg/m^3, density of concrete
p2=1000;//kg/m^3, density of water
W1=0.5*2.7*6.6*t*p1*g/1000;//kN, Weight of concrete
    component 1
W2=1.5*6.6*t*p1*g/1000;//kN, Weight of concrete
    component 2
W3=1/3*3*5.4*t*p1*g/1000;//kN, Weight of concrete
    component 3
W4=2/3*3*5.4*t*p2*g/1000;//kN, Weight of water
P=0.5*2.7*6.6*t*p1*g/1000;//kN, pressure force
    exerted by water

// Applying sum(F_x)=0
H=42.9;//kN, Horizontal reation at A

//Applying sum(Fy)=0
V=W1+W2+W3+W4;//kN, Vertical Reaction at A

printf("The horizontal reaction is H=%.1f kN ,
    Vertical rection at A V=%.1f kN, \n",H,V);
//Applying sum(M_A)=0
M=W1*1.8+W2*3.45+W3*5.1+W4*6-P*1.8;//kN.m, Moment at
```

```
    A


// We can replace force couple system by single
    force acting at distance right to A
d=M/V;// m Distance of resultant force from A

printf("The moment about A is M=%.1f kN.m
    anticlockwise and \n if we replace it by force
    couple system resultant,s distance from A is d=
    %0.2f m  \n",M,d);
//Difference is because of round off
```

**Scilab code Exa 5.11** Coordinates of centroid

```
clc;
//page 263
n=3; // no of component
r=60;//mm, radius
l=100;//mm length of cylinder
V=[0.5*4/3*%pi*(r)^3,%pi*r*r*l,-%pi/3*r*r*l];//mm^3,
     Volumes of Hemisphere, cylinder and cone
    respectively
x=[-3/8*r,l/2,3/4*l];//mm, x components of centroids
     of Hemisphere, cylinder and cone respectively

sumV=0;
sumxV=0;

for(i=1:n)
    sumV=sumV+V(i);
    sumxV=sumxV+x(i)*V(i);

end

```

```


//Location of centre of gravity
X=sumxV/sumV;// X co-ordinate

printf("Co-ordinates of centroid are X= %.0f mm \n",X);
```

---

**Scilab code Exa 5.12** components of centroids

```
clc;
clear all
//page 264
l=4.5; // in in
b=2;//in
h=.5;//in
a_I=l*b*h
a_II=((1/4)*%pi*b^2*h)
a_III=-%pi*(h^2)*h
a_IV=-%pi*(h^2)*h
V=[a_I a_II a_III a_IV]
//disp(V)

x=[.25,1.3488,.25,.25];//in, x components of centroids of part I,II , III and IV respectively
y=[-1,-0.8488,-1,-1];//in, y components of centroids of part I,II , III and IV respectively
z=[2.25,0.25,3.5,1.5];//in, z components of centroids of part I,II , III and IV respectively


for(i=1:4)
    temp=0
    sum_xV=0
    sum_xV=V(i)*x(i)
```

```
   y(i)=[sum_xV]
end
x=sum(y)
printf("The sum of x*V=%f  in^4 \n",x)

for(i=1:4)
    temp=0
    sum_zV=0
    sum_zV=V(i)*z(i)
   y(i)=[sum_zV]
end
z=sum(y)
printf("The sum of z*V=%f  in^4 \n",z)

for(i=1:4)
    temp=0
    sum_yV=0
    sum_yV=V(i)*y(i)
   y(i)=[sum_yV]
   disp(y(i))
end
s=sum(y)
printf("The sum of y*V=%f  in^4 \n",s)
```

# Chapter 6

# Analysis of structures

**Scilab code Exa 6.1** force

```
clc;
//page 294
//Entire truss
//Applying sum(M_C)=0
E=(10*12+5*6)/3;//kN

//Applying sum Fx=0
Cx=0

//Applying sumFy=0

Cy=10+5-E;//kN

//At joint A
//By proportion 10kN/4=F_AB/3=F_AD/5
F_AB=10/4*3;//kN, force in member AB
F_DA=10/4*5;//kN, force in member AD

//At joint D
F_DB=F_DA;//kN, force in member DB
F_DE=2*3/5*F_DA;//kN, force in member DE
```

```

//At joint B
//applying sumFy=0
F_BE=5/4*(-5-4/5*F_DB);//kN, force in member BE
//Applying sumFx=0

F_BC=F_AB+3/5*F_DB-3/5*F_BE;//kN, force in member BC

//At joint E
//Applying sumFx=0
F_EC=-5/3*(F_DE-3/5*F_BE);//kN, Force in member EC

printf("The forces in member of truss are \n F_AB= %.1f kN T \n F_AD= %.1f kN C, \n F_DB= %.1f kN T, \n F_DE= %.0f kN C \n F_BE= %.2f kN  \n F_BC= %.2f kN \n F_EC= %.2f kN ",F_AB,F_DA,F_DB,F_DE,F_BE,F_BC,F_EC);
//Variation in answe because of round off
```

**Scilab code Exa 6.2** force

```
clc;
//page 306
//Entire truss
v1=140;//kn, verical force 1
v2=140;//kN, Vertical force 2
h=80;//kN , Horizontal force
//Applying sum(M_B)=0
J=(v1*4+v2*12+h*5)/16;//kN

//Applying sum Fx=0
Bx=-h;//kN, negative sign shows it is along negative x axis

//Applying sumFy=0
```

```

By=v1+v2-J;//kN

//Force in member EF
//Applying sumFy=0
F_EF=By-v2;//kN, Force in member EF
printf("Force in member EF is %.0f kN \n Negative sign shows member is in compression \n",F_EF);

//Force in member GI
F_GI=(-J*4-Bx*5)/5;//kN Force in member GI
printf("Force in member GI is %.0f kN \n Negative sign shows member is in compression \n",F_GI);
//Answer difference is because of rounding off variables
```

**Scilab code Exa 6.3** Calculation of force

```
clc;
//page 307
//Entire truss
vB=1;//kN, verical force at B
vD=1;//kN, verical force at D
vF=1;//kN, verical force at F
vH=1;//kN, verical force at H
vJ=1;//kN, verical force at J
vC=5;//kN, verical force at C
vE=5;//kN, verical force at E
vG=5;//kN, verical force at G
h=8;//m, height
v=5;//m, horizontal distance between successive node

A=12.50;//kN, reaction at A
L=7.50;//kN, reaction at L

```

```
alpha=atan(h/3/v);// rad, angle made by inclined members with X axis
//alpha=alpha/%pi*180;// Conversion of angle into degrees



//Force in member GI
//Applying sum(M_H)=0
F_GI=(L*2*v-vJ*v)/(2*v*tan(alpha));//kN Force in member GI
printf("Force in member GI is %.2f kN \n ",F_GI);

//Force in member FH
//Applying sum(M_G)=0
F_FH=(L*3*v-vH*v-vJ*2*v)/(-h*cos(alpha));//kN, Force in member FH
printf("Force in member FH is %.2f kN \n Negative sign shows member is in compression \n",F_FH);


//Force in member GH
be=atan(v/(2*v*tan(alpha)));//rad, as tan(be)=GI/HI
//Applying sum(M_L)=0
F_GH=(-vH*v-vJ*2*v)/(3*v*cos(be));//kN, Force in member FH
printf("Force in member GH is %.3f kN \n Negative sign shows member is in compression \n",F_GH);
```

**Scilab code Exa 6.4** components of force

```
clc;
//page 319
//Entire truss
//Applying sum(Fy)=0
```

```
Ay=480;//N, Y component of reaction at A
//Applying sum(M_A)=0
B=480*100/160;//N,  reaction at B
//Applying sum(Fx)=0
Ax=-300;//N, X component of reaction at A

alpha=atan(80/150);//radian

//Free body member BCD

//Applying sum(M_C)=0
F_DE=(-480*100-B*60)/(sin(alpha)*250);//N,  Force in
    link DE
printf("Force in link DE is F_DE=%.0f N\n Negative
   sign shows force is compressive\n",F_DE);
//Applying sum(Fx)=0
Cx=F_DE*cos(alpha)-B;//N, X component of force
   exerted at C
//Applying sum(Fy)=0
Cy=F_DE*sin(alpha)+Ay;//N, Y component of force
   exerted at C
printf("Components of force exerted at C is Cx=%.0f
   N and Cy=%.0f N \n",Cx,Cy);
```

**Scilab code Exa 6.5** components of force

```
clc;
//page 320
P=18;//kN, Force applied at D
AF=3.6;//m, Length AF
EF=2;//m, Length EF
ED=2;//m, Length ED
DC=2;//m, Length DC
//Entire frame
//Applying sum(M_F)=0
```

```
Ay=-P*(EF+ED)/AF;//kN, Y component of reaction at A

//Applying sum(Fx)=0
Ax=-P;//kN, X component of reaction at A
//Applying sum(Fy)=0
F=-Ay;//kN,  reaction at B


printf("Components of force exerted at A is Ax=%.0f
   kN and Ay=%.0f kN \n",Ax,Ay);
printf("Force exerted at F is F=%.0f kN \n",F);
//Free body member BE
//Applying sum(Fx)=0
//B=E, and as it is 2 force member
By=0;
Ey=0;

//Member ABC
//Applying sum(Fy)=0
Cy=-Ay;//kN, Y component of force exerted at C
//Applying sum(M_C)=0
B=(Ay*AF-Ax*(DC+ED+EF))/(ED+DC);//kN,  Force in link
    DE
printf("Force exerted at B is B=%.0f kN  \n",B);
//Applying sum(Fx)=0
Cx=-Ax-B;//kN, X component of force exerted at C

printf("Components of force exerted at C is Cx=%.0f
   kN and Cy=%.0f kN \n",Cx,Cy);

printf("Negative signs shows forces are in negative
   direction\n")
```

**Scilab code Exa 6.6** Force

```
clc;
P=3;//kN, Horizontal Force applied at A
AB=1;//m, perpendicular distance between A and B
BD=1;//m, perpendicular distance between D and B
CD=1;//m, perpendicular distance between C and D
FC=1;//m, perpendicular distance between C and F
EF=2.4;//m, perpendicular distance between E and F
//Entire frame
//Applying sum(M_E)=0
Fy=P*(AB+BD+CD+FC)/EF;//kN, Y component of reaction at F


//Applying sum(Fy)=0
Ey=-Fy;//kN, Y component of  reaction at E

//Free body member ACE
//Applying sum(Fy)=0, and sum(M_E)=0 we get 2 equation
A=[-AB/sqrt(AB^2+EF^2),CD/sqrt(CD^2+EF^2);-EF/sqrt(AB^2+EF^2)*(AB+BD+CD+FC),-EF/sqrt(CD^2+EF^2)];// Matrix of coefficients
B=[Ey;-P*(AB+BD+CD+FC)];// Matrix B
X=linsolve(A,B);//kN Solution matrix
F_AB=X(1);//kN, Forec inmember AB
F_CD=X(2);//kN, Forec inmember CD
Ex=-P-EF/sqrt(AB^2+EF^2)*F_AB-EF/sqrt(CD^2+EF^2)*F_CD;//kN, X component of force exerted at E
//Free body : Entire frame
//Applying sum(F_X)=0
Fx=-P-Ex;//kN, X component of force exetered at F
printf("Components of force exerted at F is Fx=%.1f kN and Fy=%.0f kN \n",Fx,Fy);
printf("Force in member AB is F_AB=%.1f kN \n",F_AB);
printf("Force in member CD is F_CD=%.1f kN \n",F_CD);
printf("Components of force exerted at E is Ex=%.1f
```

```
kN and Ey=%.1f kN \n",Ex,Ey);

printf("Negative signs shows forces are in negative direction\n")
```

# Chapter 7

# Forces in beams and cable

**Scilab code Exa 7.1** free body diagram we

```
clc;
//Page 335
P=2400;//N, Vertical Force applied at D
AB=2.7;//m, perpendicular distance between A and B
BE=2.7;//m, perpendicular distance between E and B
BK=1.5;//m, perpendicular distance between B and K
AJ=1.2;//m, perpendicular distance between A and J
EF=4.8;//m, perpendicular distance between E and F
BD=3.6;//m, perpendicular distance between D and B
//For entire truss
//By free body diagram we get the force at A, B , c
A=1800;//N
B=1200;//N
C=3600;//N
alpha=atan(EF/(AB+BE));//rad
//a. Internal forces at j
//Applying sum(M_J)=0
M=A*AJ;//N.m,Couple on member ACF at J
//Applying sum(Fx)=0
F=A*cos(alpha);//N, Axial force at J
//Applying sum(Fy)=0
```

```
V=A*sin(alpha);//N, shearing force at J
printf("Thus, Internal forces at J are equivalent to \n Couple M = %.0f N.m \n Axial force F= %.0f N \n Shearing force V= %.0f N\n",M,F,V);

//a. Internal forces at K
//Applying sum(M_K)=0
M=B*BK;//N.m,Couple on frame
//Applying sum(Fx)=0
F=0;//N, Axial force at J
//Applying sum(Fy)=0
V=-B;//N, shearing force at J
printf("Thus, Internal forces at K are equivalent to \n Couple M = %.0f N.m \n Axial force F= %.0f N \n Shearing force V= %.0f N\n",M,F,V);
```

**Scilab code Exa 7.2** free body diagram

```
clc;
//page 344
//Drawing of shear and bending moment diagram
printf("Given problem is for drawing diagram, this diagram is drawn by step by step manner.\n ");
F_A=-20;//kN, force applied at A
F_C=-40;//kN, force applied at C
AB=2.5;//m, perpendicular distance between A and B
BC=3;//m, perpendicular distance between C and B
CD=2;//m, perpendicular distance between C and D
//By free body of entire beam
//By sum(m_D)=0
R_B=-(CD*F_C+(AB+BC+CD)*F_A)/(BC+CD);//kN, Reaction atB
//By sum(m_A)=0
R_D=-(BC*F_C-(AB)*F_A)/(BC+CD);//kN, Reaction atB
//For section 1
```

```
//Applying sum(Fy)=0
V1=F_A;//kN
//Applying sum(M1)=0
M1=V1*0;//kN.m

//For section 2
//Applying sum(Fy)=0
V2=F_A;//kN
//Applying sum(M1)=0
M2=F_A*AB;//kN.m

//For section 3
//Applying sum(Fy)=0
V3=R_B+F_A;//kN
//Applying sum(M1)=0
M3=F_A*AB;//kN.m

//For section 4
//Applying sum(Fy)=0
V4=R_B+F_A;//kN
//Applying sum(M1)=0
M4=F_A*(AB+BC)+R_B*BC //kN.m

//For section 5
//Applying sum(Fy)=0
V5=R_B+F_A+F_C;//kN
//Applying sum(M1)=0
M5=F_A*(AB+BC)+R_B*BC//kN.m

//For section 6
//Applying sum(Fy)=0
V6=R_B+F_A+F_C;//kN
//Applying sum(M1)=0
M6=V6*0//kN.m
X=[0,2.5,2.5,5.5,5.5,7.5]

V=[V1,V2,V3,V4,V5,V6];//Shear matrix
M=[M1,M2,M3,M4,M5,M6];//Bending moment matrix
```

```
xtitle( 'Shear and bending moment diagram', 'X axis', 'Y axis') ;
plot(X,V);//Shear diagram
plot(X,M,'r');//Bending moment diagram
```

**Scilab code Exa 7.3** free body diagram

```
clc;
//Drawing of shear and bending moment diagram
//Values taken in N and m instead of lb and in
printf("Given problem is for drawing diagram, this diagram is drawn by step by step manner.\n ");
F_AC=40;//lb/in, distributed load applied at A to C
F_E=400;//lb, force applied at E
AC=12;//in, perpendicular distance between A and B
CD=6;//in, perpendicular distance between C and D
DE=04;//in, perpendicular distance between E and D
EB=10;//in, perpendicular distance between E and B
AB=32;//in, length of beam AB
F=F_AC*AC;//N, Force due to districuted load at AC/2
//By free body of entire beam
//By sum(m_A)=0
By=(F*(AC/2)+F_E*(AC+CD+DE))/AB;//N,Y componet of Reaction at B
//By sum(m_B)=0
//disp(By)
A=(F*(AB-AC/2)+F_E*EB)/AB;//N, Reaction at A
//by sum(Fx)=0
//disp(A)
Bx=0;//N, xcomponent of rection at B
```

```
//Diagrams

//For section A to C

//Applying sum(Fy)=0

i=0;

for x=0:2:12

    i=i+1;

    X(i)=x;

V(i)=A-F*x;//N

//Applying sum(M1)=0

M(i)=A*x-F/2*x^2;//N.m

end



//For section Cto D

//Applying sum(Fy)=0

for x=12:2:18
    i=i+1;

    X(i)=x;

V(i)=A-F;//N

//Applying sum(M1)=0

M(i)=A*x-F*(x-0.15);//N.m
```

```
end
//For section D to B
for x=18:2:32
i=i+1;
    X(i)=x;
    //Applying sum(Fy)=0
    V(i)=A-F-F_E;//N
//Applying sum(M1)=0
M(i)=A*x-F*(x-0.15)+F_E*DE-F_E*(x-0.045);//N.m
end
xtitle( 'Shear and bending moment diagram', 'X axis', 'Y axis') ;
plot(X,V,'r');//Shear diagram
plot(X,M,'-');//Bending moment diagram
```

**Scilab code Exa 7.4** free body diagram

```
clc;
```

```
//Drawing of shear and bending moment diagram
printf("Given problem is for drawing diagram, this
    diagram is drawn by step by step manner.\n ");
F_B=500;//N, force applied at B
F_C=500;//N, force applied at C.
F_DE=2400;//N/m, distributed load applied at D to E
AB=0.4;//m, perpendicular distance between A and B
BC=0.4;//m, perpendicular distance between C and B
CD=0.4;//m, perpendicular distance between C and D
DE=0.3;//m, perpendicular distance between E and D
F_E=F_DE*DE;//N, force exerted at DE/2 from E

//By free body of entire beam
//By sum(m_D)=0
A=(CD*F_C+(BC+CD)*F_B-F_E*DE/2)/(AB+BC+CD);//N,
    Reaction at A
//By sum(Fy)=0
Dy=F_C+F_B+F_E-A;//N,Y component of  Reaction at D
//By sum(Fx)=0
Dx=0;//N,Y component of  Reaction at D
//For section 1
//Applying sum(Fy)=0
V1=A;//N, shear force from A to B

//For section 2
//Applying sum(Fy)=0
V2=A-F_B;//N, shear force from B to C

//For section 3
//Applying sum(Fy)=0
V3=A-F_B-F_C;//N, shear force  from C to D

//For section 4
//Applying sum(Fy)=0
V4=A-F_B-F_C+Dy;//N, shear force At D

//For section 5
//Applying sum(Fy)=0
```

```
V5=0;//N, shear force at A
//Area under bending curve is change in bending
   moment of that 2 points
MA=0;//N.m
MB=MA+V1*AB;//N.m
MC=MB+V2*BC;//N.m
MD=MC+V3*CD;//N.m
ME=MD+1/2*V4*AB;//N.m


X=[0,0.4,0.4,0.8,0.8,1.2,1.2,1.5];
V=[V1,V1,V2,V2,V3,V3,V4,V5];//Shear matrix,

plot(X,V);//Shear diagram
X=[0,AB,AB+BC,AB+BC+CD,AB+BC+CD+DE];
M=[MA,MB,MC,MD,ME];//Bending moment matrix
plot(X,M,'r');//Bending moment diagram
```

**Scilab code Exa 7.5** free body diagram

```
clc;
//Drawing of shear and bending moment diagram
printf("Given problem is for drawing diagram, this
   diagram is drawn by step by step manner.\n ");

w=20;//kN/m, distributed load applied at D to E
AB=6;//m, perpendicular distance between A and B
BC=3;//m, perpendicular distance between C and B

F_B=w*AB;//kN, force exerted at AB/2 from A

//By free body of entire beam
//By sum(m_C)=0
RA=(F_B*(AB/2+BC))/(AB+BC);//kN, Reaction at A

```

```
//By sum(m_A)=0
RC=(F_B*(AB/2)/(AB+BC));//kN, Reaction at C

//For section 1
//Applying sum(Fy)=0
VA=RA;//N, shear force just to right to A

//For section 2
//Applying sum(Fy)=0
VB=VA-F_B;//kN, shear force just left to B

//For section 3
//Applying sum(Fy)=0
VC=VB;//kN, shear force  from B to C


//Bending moment at each end is zero
// Maximum bending moment is at D where V=0
VD=0;//kN

x=-(VD-VA)/w;//m, location of maximum bending moment
printf("Maximum bending moment is at D x= %.0f m
   from A\n",x);
MA=0;//kN.m
MD=MA+1/2*VA*x;//kN.m, maximum bending moment is at
   D
MB=MD+1/2*VB*(AB-x);//N.m
MC=MB+VB*BC;//N.m

printf("Maximum bending moment is at MD= %.0fkN. m
   from A\n",MD);
X=[0,x,AB,AB+BC];//m,
V=[VA,VD,VB,VC];//kN,Shear matrix,

plot(X,V);//Shear diagram
X=[0,x,AB,AB+BC];//m
M=[MA,MD,MB,MC];//kN.m,Bending moment matrix
plot(X,M,'r');//Bending moment diagram
```

**Scilab code Exa 7.8** free body diagram

```
clc;
F_B=30;//kN, Vertical Force applied at B
F_C=60;//kN, Vertical Force applied at C
F_D=20;//kN, Vertical Force applied at D
AB=6;//m, perpendicular distance between A and B
BC=3;//m, perpendicular distance between C and B
CD=4.5;//m, perpendicular distance between c and D
DE=4.5;//m, perpendicular distance between D and E
AE=6;//m, vertical perpendicular distance between A and E
AC=1.5;//m, vertical perpendicular distance between A and C
//For entire cable
//Sum(M_E)=0, AB*Ax-Ay*(AB+BC+CD+DE)+F_B*(BC+CD+DE)+F_C*(CD+DE)+F_D*(DE)=0

//Free body ABC
//Sum(M_c)=0 gives -Ax*AC-Ay*(AB+BC)+F_B*BC=0
//we get 2 equations in Ax and Ay
A=[AB,-(AB+BC+CD+DE);-AC,-(AB+BC)];//Matrix of coeficients
B=[-(F_B*(BC+CD+DE)+F_C*(CD+DE)+F_D*(DE));-F_B*BC];
X=linsolve(A,-B);//kN, Solution matrix
Ax=X(1);//kN, X component of reaction at A
Ay=X(2);//kN, Y component of reaction at A


//a. Elevation of points B and D
//Free body AB
//sum(M_B)=0
yB=-Ay*AB/Ax;//m, below A
printf("Elevation of point B is %.2f m below A\n",yB
```

```
);
//free body ABCD
//sum(M_D)=0
yD=(Ay*(AB+BC+CD)-F_B*(BC+CD)-F_C*CD)/Ax;//m, above
    A
printf("Elevation of point D is %.2f m above A\n",yD
    );

//Maximum slope and maximum tension
theta=atan((AE-yD)/DE);//rad
Tmax=-Ax/cos(theta);//kN, maximum tension
theta=theta/%pi*180;//degree

printf("Maximum slope is theta= %.1f degree and
    maximum tension in the cable is Tmax= %.1f kN \n"
    ,theta,Tmax);
```

**Scilab code Exa 7.9** free body diagram

```
clc;
yB=0.5;//m, sag of the cable
m=0.75;//kg/m, mass per unit length
g=9.81;//m/s^2, acceleration due to gravity
AB=40;//m, distance AB
//a. Load P
w=m*g;//N/m , Load per unit length
xB=AB/2;//m, distance CB
W=w*xB;//N, applied at halfway of CB

//Summing moments about B
//sum(M_B)=0
To=W*xB/2/yB;//N
//from force triangle
TB=sqrt(To^2+W^2);//N, =P, as tension on each side
    is same
```

```
printf("Magnitude of load P= %.0f N \n",TB);
//slope of cable at B
theta=atan(W/To);//rad
theta=theta*180/%pi;//degree, conversion to degree
printf("Slope of cable at B is theta= %.1f degree\n"
    ,theta);
//length of cable
//applying eq. 7.10
sB=xB*(1+2/3*(yB/xB)^2);//m

printf("Total length of cable from A to B is Length=
    %.4f m\n",2*sB);
```

**Scilab code Exa 7.10** free body diagram

```
clc;
AB=150;//m, distance AB
s=30;//m, sag of cable
w=45;//N/m Uniform weigth per unit length of cable

//Equation of cable, by 7.16
//Coordinates of B

xB=AB/2;//m
C=[99,105,98.4,90];//trial values

for i=1:4
    if ((30/C(i)+1)-cosh(xB/C(i)))<0.0001 then c=C(i
       );
    break;
end
end
yB=s+c;//m

//Maximum and minimum values of tension
```

```
Tmin=w*c;//N, To
Tmax=w*yB;//N TB
printf("Minimum value of tension in cable is Tmin= %
    .0f N\n",Tmin);
printf("Maximum value of tension in cable is Tmax= %
    .0f N\n",Tmax);
//Length of cable

S_CB=sqrt(yB^2-c^2);//m, one halph length by 7.17
S_AB=2*S_CB;//m, full length of cable

printf("Fulllength of cable is s_AB= %.0f m\n",S_AB)
    ;
```

# Chapter 8

# Friction

**Scilab code Exa 8.1** value of friction force

```
clc;
//page 396

h=100;//lb, horizontal force
W=300;//lb, weight of block

us=0.2;// Coeffiecient of static friction
uk=0.20;//Co=efficient of kinetic friction

//Applying sumFx =0 , we get
F=h-3/5*W;//lb, Force along plane
F=-F

//Applying sumFy=0, we get
N=4/5*W//lb, Normal force to the plane


printf("Force F required to maintain the
    equillibrium is thus %.0f lb, up and to right\n",
    F);

```

```
// Maximum friction force
Fm=us*N;//lb,Maximum friction force
printf("\n Maximum friction force is %.2f lb is less than that of required to maintain equillibrium that is %.2f lb \n So, equillibrium will nat maintain and block wil move down\n",Fm,F);
// Actual value of friction force
Fk=(0.6*300)-(h)-(Fm);//lb, Actual value of friction force
printf("\nActual value of friction force is %.2f lb directed up and to the right\n",Fk);
```

**Scilab code Exa 8.2** Force P to prevent block

```
clc;
clear all
//page 397
F=800;//N Firce in verical direction
us=0.35;// Coeffiecient of static friction
uk=0.25;//Co=efficient of kinetic friction
theta=25;//degree, angle of inclination
theta=theta*%pi/180;//rad, Conversion into radian
// Force P start block moving up
// At static equillibrium Tan(Theta_s)=us
theta_s=atan(us);//rad
P=F*tan(theta+theta_s);//N,Force P to start block moving up
printf("Force P to start block moving up is %.0f N\n",P);


// Force P to keep block moving up
// At kinetic equillibrium Tan(Theta_k)=uk
theta_k=atan(uk);//rad
P=F*tan(theta+theta_k);//N,Force P to keep block
```

```
moving up
printf("Force P to keep block moving up is %.0f N\n",P);


// Force P to prevent  block from sliding down

theta_s=atan(us);//rad
P=F*tan(theta-theta_s);//N,Force P to prevent  block from sliding down
printf("Force P to prevent  block from sliding down is %.0f N\n",P);
```

**Scilab code Exa 8.3** Minimum distance

```
clc;
us=0.25;// Coeffiecient of static friction
//Applying equillibrium equation we get relation in x
printf("Apply equillibrium equations. It is theoritical part. \n");
x=12-(.75*2)+1.5//in, Distance at which the applied load can be supported
printf("Minimum distance at which the applied load can be supported is %.0f in\n",x);
```

**Scilab code Exa 8.4** force required

```
clc;
//page 411
F=400;//lb, force exerte
us=0.35;// Coeffiecient of static friction
phi=atand(us);//rad, angle of friction
```

```
//disp(phi)
theta=8;//degree, angle of inclination
theta=theta*%pi/180;//rad, Conversion into radian

//Using sine rule
//force p to raise block
//free body , block B
R1=F*sind(109.3)/(sind(43.4))
//free body wedge A
P=R1*sind(46.6)/(sind(70.7))
printf(" force required to raise block is P=%.0f lb\n",P);

//force to lower block
//free body , block B
R1=F*sind(70.7)/(sind(98.0))
//free body wedge A
P=R1*sind(30.6)/(sind(70.7))
printf(" force required to lower block is P=%.0f lb\n",P);
```

**Scilab code Exa 8.5** Couple required to loosen clamp

```
clc;
clear all
//page 412
pitch=2;//mm, pitch of screw
d=10;//mm, mean diameter of thread
r=d/2;//mm, radius
us=0.30;// Coeffiecient of static friction
M=40;//kN.m , Maximum couple

//Force exerted by clamp
L=2*pitch;//mm, as screw is double threaded
theta=atan(L/(2*%pi*r));//rad, angle of inclination
```

```
phi=atan(us);//rad, angle of friction
Q=M/r*1000;//N, Force applied to block representing screw
Q=Q/1000//kN, Conversion into kN
W=Q/tan(theta+phi);//kN, Magnitude of force exerted on the piece of wood
printf("Magnitude of force exerted on the piece of wood  is W= %.2f kN \n",W);
//Couple required to loosen clamp
Q=W*tan(phi-theta);//kN, Force required to loosen clamp
Couple=Q*r;//N.m, Couple required to loosen clamp
printf("Couple  required to loosen clamp is %.2f N.m \n",Couple);
```

**Scilab code Exa 8.6** force required

```
clc;
clear all
//Page 423
r=1//in in
us=0.20;// Coeffiecient of static friction between shaft and pully

//Vertical Force required to raise load
rf=r*us;//in, Perpendicular distance from the center Of pully to line of action
//summing moment about B
P=(2.20*500)/1.8//lb , downward Force required to raise load
printf("Force required to raise load is %f lb in downward direction\n",P);

//Vertcal Force required to hold load

```

```
//summing moment about C
P=(1.80*500)/1.8//lb , downward Force required to hold load
printf("Force required to hold load is %.0f lb in downward direction\n",P);

//Horizontal force P to start raising the load
OE=rf;//mm,
OD=sqrt((d1/2)^2+(d1/2)^2);//mm, pythagorus theorm
theta=asin(OE/OD);//rad,

// from force triangle
P=500*cotd(40.9);//lb, Horizontal force P to start raising the load
printf("Horizontal force P required to start raising the load is %.0f lb\n",P);
```

**Scilab code Exa 8.7** Tension

```
clc;
//page 431
T1=150;//N, Force on free end of hawser
T2=7500;//N, Force on other end of hawser


//a, coefficient of friction
bta=2*2*%pi;//rad, angle of contact, 2 turns
//By equation 8.13
us=log(T2/T1)/bta;// Co-efficient of static friction
printf("Coefficient of static friction between hawser and ballard is us= %0.3f \n",us);

//Number of wraps when tension in hawser=75 kN

bta=3*2*%pi//in rad
```

```
//One turn = 2* pi angle , bta corresponds to
ten=T1*exp(bta*us)
printf(" Tension  is %f N \n ",ten);
```

**Scilab code Exa 8.8** Torque

```
clc;
//page 432
//Given
T2=600;//lb , Tension from side 2
us=0.25;// Coeffiecient of static friction between pulley and belt
bta=(2*%pi)/3;//Co=efficient of kinetic friction between pulley and belt
r1=8//in in
//Pulley B

T1=T2/(exp(us*bta))//N, Tension from side 1
//disp(T1)

//Pulley A
//Aumming moment about A
MA=(T2*r1)-(T1*r1);//lb-ft , Couple MA applied to pulley which is equal and opposite to torque

printf("The largest torque which can be exerted by belt on pulley A is MA= %0.0f lb-in\n",MA);
```

# Chapter 9

# Distributed forces Moment of Inertia

**Scilab code Exa 9.4** Area of plate

```
clc;
//page 465
//Area of plate


A=9*.75;//in^2
y=1/2*13.84+1/2*.75;//in, y co-ordinate of centroid of the plate
//All values for flange are from table from book
sumA=A+8.85;//in^2 Total area
sumyA=y*A+0;//in^3
Y=sumyA/sumA;//in
//disp(Y)
//Moment of inertia
//For wide flanfe
Ix1=291+8.85*Y^2;//in^4
//for plate
Ix2=1/12*9*(3/4)^3+6.75*(7.295-3.156)^2;//in^4
//For composite area
```

```
Ix=Ix1+Ix2;//in^4

printf("Moment of inertia Ix= %.2e in^4 \n",Ix);

//Radius of gyration
kx=sqrt(Ix/sumA);//mm
printf("Radius of gyration is kx= %.1f in\n",kx);
```

**Scilab code Exa 9.5** Principle moment of inertia

```
clc;
//page 466
//Given
r=90;//mm, radius of half circle
b=240;//mm, width
h=120;//mm, height

//Moment of inertia of rectangle
Ixr=1/3*b*h^3;//mm^4

//Moment of inertia of half circle
a=4*r/(3*%pi);//mm

b=h-a;//mm, Distance b from centroid c to X axis

I_AA=1/8*%pi*r^4;//mm^4, Moment of inertia of half
    circle with respect to AA'
A=1/2*%pi*r^2;//mm^2, Area of half circle

Ix1=I_AA-A*a^2;//mm^4, Parallel axis theorem

Ixc=Ix1+A*b^2;//mm^4, Parallel axis theorem

//Moment of inertia of given area
Ix=Ixr-Ixc;//mm^4
```

```

printf("Moment of inertia of area about X axis is Ix
    = %2.2e mm^4\n",Ix);
```

**Scilab code Exa 9.7** Principle moment of inertia

```
clc;
//page 479
Ix=10.38;//in^4,Moment of inertia about x axis
Iy=6.97;//in^4,Moment of inertia about y axis

Ixy=-3.28+0-3.28
disp(Ixy)//in in^4

//Principal axes
tan_2_theta_m=-(2*Ixy)/(Ix-Iy)
two_theta_m=atand(tan_2_theta_m)
theta_m=two_theta_m/2
printf("Orientation of principle axes of section
    about O is Theta_m= %.1f degree \n",theta_m);

//Principle moment of inertia , eqn 9.27
Imax=(Ix+Iy)/2+sqrt(((Ix-Iy)/2)^2+Ixy^2);//mm^4
Imin=(Ix+Iy)/2-sqrt(((Ix-Iy)/2)^2+Ixy^2);//mm^4

printf("Principle moment of inertia of section about
     O are \n Imax= %.2e in^4 \n Imin= %.0e  in^4\n",
    Imax,Imin);
//answer difference is due to roundoff
```

# Chapter 10

# Method of virtual work

**Scilab code Exa 10.3** Force exerted by each cylinder

```
clc;
m=1000;//kg, mass of krate
theta=60;//degree
theta=theta*%pi/180;//radians, conversion into rad
a=0.70;//m
L=3.20;//m
g=9.81;//m/s^2
//From theory we get
W=m*g;//N, Weight
W=W/1000;//kN, conversion into kN
S=sqrt(a^2+L^2-2*a*L*cos(theta));//m
F_DH=W*S/L/tan(theta);//kN

printf("Force exerted by each cylinder is F_DH=%.2f
   kN",F_DH);
```

---

**Scilab code Exa 10.4** Angle

```
clc;
m=10;//kg, mass of rim
r=300;//mm, radius of disk
a=0.08;//m
b=0.3;//m
k=4;//kN/m
g=9.81;//m/s^2  gravity
//From theory we get

//sin(theta)=k*a^2/m/g/b*theta
dif=1;
for theta=0:0.001:1
    dif=sin(theta)-k*a^2/m/g/b*theta;
    if dif<=0.001 then printf("theta= %.3f rad or %.1f degrees\n",theta,theta/%pi*180);
    end
end
```