# Chapter 3
# Centers of Mass

## 3.1 First Moment

Figure 3.1 shows a set of $n$ points $P_i$, $\{S\} = \{P_1, P_2, \ldots, P_n\} = \{P_i\}_{i=1,2,\ldots,n}$. The position vector of a point $P_i$ relative to an arbitrarily selected reference point $O$ is $\mathbf{r}_{P_i}$, where $\mathbf{r}_{P_i} = \mathbf{r}_i$. A scalar $s_i$ can be associated with $P_i$ as for example the mass $m_i$ of a particle situated at $P_i$. The *first moment* of a point $P_i$ with respect to a point $O$ is the vector $\mathbf{M}_i = s_i \, \mathbf{r}_{P_i}$. The scalar $s_i$ is called the *strength* of $P_i$. The strengths of the points $P_i$ are $s_i$, $i = 1, 2, \ldots, n$ and are $n$ scalars, having the same dimension, and associated with one of the points of $\{S\}$.

The *centroid* of the set $\{S\}$ is the point $C$ with respect to which the sum of the first moments of the points of $\{S\}$ is equal to zero. The position vector of $C$ relative to a point $O$ is $\mathbf{r}_C$. The position vector of $P_i$ relative to $C$ is $\mathbf{r}_i - \mathbf{r}_C$. The sum of the first moments of the points $P_i$ with respect to $C$ is $\sum_{i=1}^{n} s_i(\mathbf{r}_i - \mathbf{r}_C)$. If $C$ is the centroid of $\{S\}$ then

$$\sum_{i=1}^{n} s_i(\mathbf{r}_i - \mathbf{r}_C) = \sum_{i=1}^{n} s_i \mathbf{r}_i - \mathbf{r}_C \sum_{i=1}^{n} s_i = 0.$$

The position vector $\mathbf{r}_C$ of the centroid $C$ is given by

$$\mathbf{r}_C = \frac{\sum_{i=1}^{n} s_i \, \mathbf{r}_i}{\sum_{i=1}^{n} s_i}.$$

If $\sum_{i=1}^{n} s_i = 0$ the centroid is not defined. The centroid $C$ of a set of points of given strength does not depend on the choice of the reference point $O$.

D. B. Marghitu et al., *Statics with MATLAB®*,
DOI: 10.1007/978-1-4471-5110-4_3, © Springer-Verlag London 2013

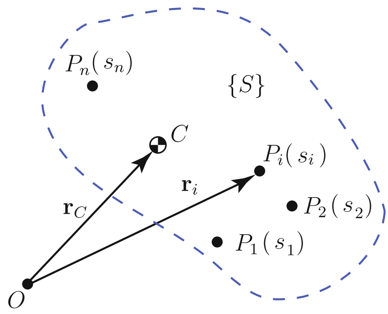



**Fig. 3.1** Set of points and centroid of a set of points

## 3.2 Center of Mass of a Set of Particles

The *center of mass* of a set of particles $\{S\} = \{P_1, P_2, \ldots, P_n\} = \{P_i\}_{i=1,2,\ldots,n}$ is the centroid of the set of points with $s_i = m_i,\ i = 1, 2, \ldots, n$, where $m_i$ is the mass of the particle $P_i$. The position vector of the center of mass, $C$, of the system with $n$ particles is

$$\mathbf{r}_C = \frac{\sum_{i=1}^{n} m_i\,\mathbf{r}_i}{\sum_{i=1}^{n} m_i} = \frac{\sum_{i=1}^{n} m_i\,\mathbf{r}_i}{M}, \tag{3.1}$$

where $M$ is the total mass of the system.

## 3.3 Center of Mass of a Body

The position vector of the center of mass $C$, Fig. 3.2, of a body of mass $m$ and volume $V$ relative to a point $O$ is

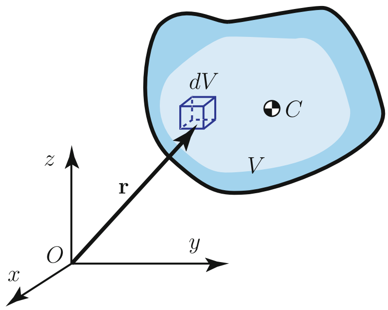



**Fig. 3.2** Center of mass of a volume $V$

$$\mathbf{r}_C = \frac{\iiint\limits_V \mathbf{r}\,\mathrm{d}m}{\iiint\limits_V \mathrm{d}m}. \tag{3.2}$$

The mass of a differential element of volume d$V$ is d$m = \rho\,\mathrm{d}V$ where $\rho$ is the density of a body (mass per unit volume). The orthogonal cartesian coordinates of $C$ are

$$x_C = \frac{\iiint\limits_V x\,\rho \mathrm{d}V}{\iiint\limits_V \rho \mathrm{d}V},\ y_C = \frac{\iiint\limits_V y\,\rho \mathrm{d}V}{\iiint\limits_V \rho \mathrm{d}V},\ z_C = \frac{\iiint\limits_V z\,\rho \mathrm{d}V}{\iiint\limits_V \rho \mathrm{d}V}. \tag{3.3}$$

The center of mass of a body is the point at which the total moment of the body's mass about that point is zero. If the mass density $\rho$ of the body is the same at all points of the body, the body is uniform and the coordinates of the center of the mass $C$ are

$$x_C = \frac{\iiint\limits_V x\,\mathrm{d}V}{\iiint\limits_V \mathrm{d}V},\ y_C = \frac{\iiint\limits_V y\,\mathrm{d}V}{\iiint\limits_V \mathrm{d}V},\ z_C = \frac{\iiint\limits_V z\,\mathrm{d}V}{\iiint\limits_V \mathrm{d}V}. \tag{3.4}$$

For a uniform curve $\rho = \rho_l = m/L$ is the mass per unit of length and

$$x_C = \frac{\int\limits_L x\,\mathrm{d}l}{\int\limits_L \mathrm{d}l},\ y_C = \frac{\int\limits_L y\,\mathrm{d}l}{\int\limits_L \mathrm{d}l},\ z_C = \frac{\int\limits_L z\,\mathrm{d}l}{\int\limits_L \mathrm{d}l}, \tag{3.5}$$

where $L$ is the length of the curve. For a uniform surface $\rho = \rho_s = m/A$ is the mass per unit of area and

$$x_C = \frac{\iint\limits_A x\,\mathrm{d}A}{\iint\limits_A \mathrm{d}A},\ y_C = \frac{\iint\limits_A y\,\mathrm{d}A}{\iint\limits_A \mathrm{d}A},\ z_C = \frac{\iint\limits_A z\,\mathrm{d}A}{\iint\limits_A \mathrm{d}A}, \tag{3.6}$$

where $A$ is the area of the surface.

The *method of decomposition* is used to locate the center of mass of a composite body:

1. divide the body into a number of simpler body shapes, which may be particles, curves, surfaces, or solids; Holes are considered as pieces with negative size, mass, or volume.
2. locate the coordinates $x_{C_i},\ y_{C_i},\ z_{C_i}$ of the center of mass of each part of the body;
3. determine the center of mass using the equations

$$x_C = \frac{\sum_{i=1}^{n} \int_\tau x\, d\tau}{\sum_{i=1}^{n} \int_\tau \mathrm{d}\tau},\ y_C = \frac{\sum_{i=1}^{n} \int_\tau y\, d\tau}{\sum_{i=1}^{n} \int_\tau \mathrm{d}\tau},\ z_C = \frac{\sum_{i=1}^{n} \int_\tau z\, d\tau}{\sum_{i=1}^{n} \int_\tau \mathrm{d}\tau}, \tag{3.7}$$

where $\tau$ is a curve, area, or volume. Equation (3.7) can be simplify as

$$x_C = \frac{\sum_{i=1}^{n} x_{C_i}\, \tau_i}{\sum_{i=1}^{n} \tau_i},\ y_C = \frac{\sum_{i=1}^{n} y_{C_i}\, \tau_i}{\sum_{i=1}^{n} \tau_i},\ z_C = \frac{\sum_{i=1}^{n} z_{C_i}\, \tau_i}{\sum_{i=1}^{n} \tau_i}, \tag{3.8}$$

where $\tau_i$ is the length, area, or volume of the $i^{\text{th}}$ object.

## 3.4 First Moment of an Area

A planar surface of area $A$ is shown in Fig. 3.3. The first moment of the area $A$ about the $x$-axis is

$$M_x = \iint_A y\ \mathrm{d}A. \tag{3.9}$$

The first moment about the $y$-axis is

$$M_y = \iint_A x\ \mathrm{d}A. \tag{3.10}$$

The first moment of area gives information of the shape, size, and orientation of the area. The coordinates $x_C$ and $y_C$ of the center of mass of the area $A$ are calculated with

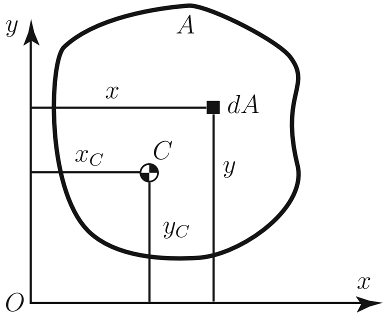



**Fig. 3.3** Planar surface of area $A$

$$x_C = \frac{\displaystyle\iint_A x \, \mathrm{d}A}{A} = \frac{M_y}{A}, \tag{3.11}$$

$$y_C = \frac{\displaystyle\iint_A y \, \mathrm{d}A}{A} = \frac{M_x}{A}. \tag{3.12}$$

The location of the center of mass of an area is independent of the reference axes employed. If the axes $xy$ have their origin at the centroid, $O \equiv C$, then these axes are called *centroidal axes*. The first moments about the centroidal axes are zero. The center of mass of an area with one axis of symmetry is located along the axis of symmetry. The axis of symmetry is a centroidal axis and the first moment of area must be zero about the axis of symmetry. If a body has two orthogonal axes of symmetry the centroid is at the intersection of these axes. For surfaces as circles, rectangles, triangles, the center of mass can be determined by inspection.

## 3.5 Center of Gravity

The *center of gravity* is a point which locates the resultant weight of a system of particles or body. The sum of moments due to individual particle weight about any point is the same as the moment due to the resultant weight located at the center of gravity. The sum of moments due to the individual particles weights about center of gravity is equal to zero. Similarly, the center of mass is a point which locates the resultant mass of a system of particles or body. The center of gravity of a body is the point at which the total moment of the force of gravity is zero. The coordinates of the center of gravity are

$$x_C = \frac{\iiint\limits_V x\rho\, g\, \mathrm{d}V}{\iiint\limits_V \rho\, g\, \mathrm{d}V},\ y_C = \frac{\iiint\limits_V y\rho\, g\, \mathrm{d}V}{\iiint\limits_V \rho\, g\, \mathrm{d}V},\ z_C = \frac{\iiint\limits_V z\rho\, g\, \mathrm{d}V}{\iiint\limits_V \rho\, g\, \mathrm{d}V}. \tag{3.13}$$

If the acceleration of gravity $g$ is constant throughout the body, then the location of the center of gravity is the same as that of the center of mass. The acceleration of gravity is $g = 9.81$ m/s$^2$ or $g = 32.2$ ft/s$^2$.

## 3.6 Theorems of Guldinus-Pappus

*Theorem 1* Consider a planar generating curve and an axis of revolution in the plane of this curve Fig. 3.4. The axis of revolution does not intersect the curve. It can only touch the generating curve. The surface of revolution $A$ developed by rotating the generating curve about the axis of revolution equals the product of the length of the generating $L$ curve times the circumference of the circle formed by the centroid of the generating curve $y_C$ in the process of generating a surface of revolution

$$A = 2\,\pi y_C L. \tag{3.14}$$

*Proof* A length element $dl$ of the generating curve is considered as shown in Fig. 3.4. For a revolution of the generating curve about axis of revolution, $x$-axis, the length element $dl$ describes the area

$$\mathrm{d}A = 2\,\pi y \mathrm{d}l.$$

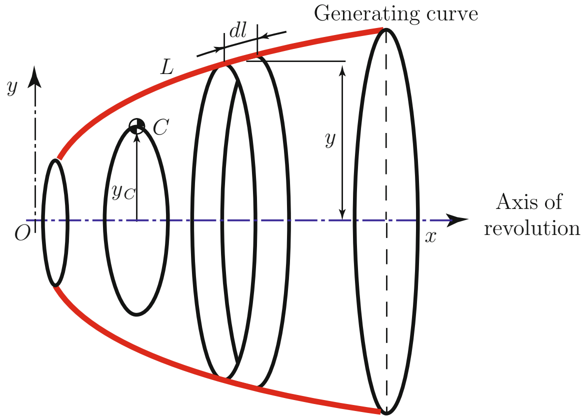



**Fig. 3.4** Surface of revolution

For the total surface of revolution developed the area is

$$A = 2\,\pi \int y\,\mathrm{d}l = 2\,\pi y_C L,$$

where $L$ is the length of the curve and $y_C$ is the centroidal coordinate of the curve. The circumferential length of the circle formed by having the centroid of the curve rotate about the $x$-axis is $2\pi y_C$. The surface of revolution $A$ is equal to $2\pi$ times the first moment of the generating curve about the axis of revolution. For a composite generating curve the following formula is used

$$A = 2\,\pi \left( \sum_i L_i y_{C_i} \right), \tag{3.15}$$

where $y_{C_i}$ is the centroidal coordinate for the $i^{\text{th}}$ line segment $L_i$. The generating curve is composed of simple curves, $L_i$ and the axis of revolution is the $x$-axis.

*Theorem 2* A generating planar surface $A$ and an axis of revolution located in the same plane as the surface is considered in Fig. 3.5. The volume of revolution $V$ developed by rotating the generating planar surface about the axis of revolution equals the product of the area of the surface times the circumference of the circle formed by the centroid of the surface $y_C$ in the process of generating the body of revolution

$$V = 2\,\pi y_C A. \tag{3.16}$$

The axis of revolution does not intersect the generating surface. It can only touch the generating plane surface as a tangent at the boundary.

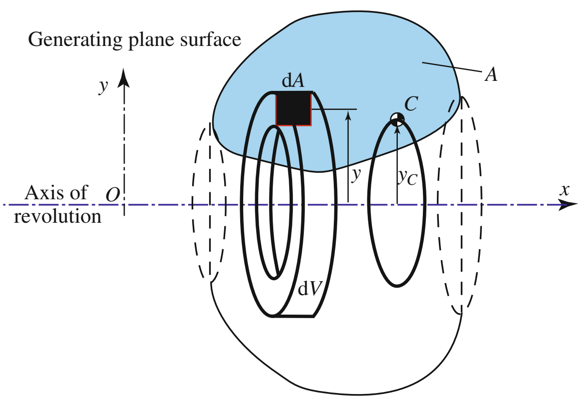



**Fig. 3.5** Volume of revolution

*Proof* The volume generated by rotating an element $dA$ of the plane surface, $A$ is shown in Fig. 3.5, about the $x$-axis is

$$\mathrm{d}V = 2\,\pi y\,\mathrm{d}A.$$

The volume of the body of revolution formed from $A$ is

$$V = 2\,\pi \int_A y\,\mathrm{d}A = 2\,\pi y_C\,A.$$

Thus, the volume $V$ equals the area of the generating surface $A$ times the circumferential length of the circle of radius $y_C$. The volume $V$ equals $2\,\pi$ times the first moment of the generating area $A$ about the axis of revolution.

The areas and center of mass for some practical configurations are shown in Fig. 3.6.

## 3.7 Examples

*Example 3.1* Find the length and the position of the center of mass for the homogeneous curve given by the Cartesian equation $y = b\sqrt{x^a}$ m, where $a = 3$, $b = 2$ and $0 \le x \le 1$ m.

*Solution* The differential element of the curve, $\mathrm{d}l = \sqrt{1 + (\mathrm{d}y/\mathrm{d}x)^2}$, is given in MATLAB by:

```
syms x real
a = 3;
b = 2;
y  = b*sqrt(x^a);
dy = diff(y,x);
dl =sqrt(1+dy^2);
% dl = (1+ (dy/dx)^2)^0.5
%  0< x < 1
```

The MATLAB statement `int (f, x, a, b)` is the definite integral of `f` with respect to its symbolic variable `x` from `a` to `b`. The length of the homogeneous curve is:

```
L = eval(int(dl,0,1));
```

and the coordinates of the center of mass $C$ are:

```
My=eval(int(x*dl,0,1));
xC=My/L;
```

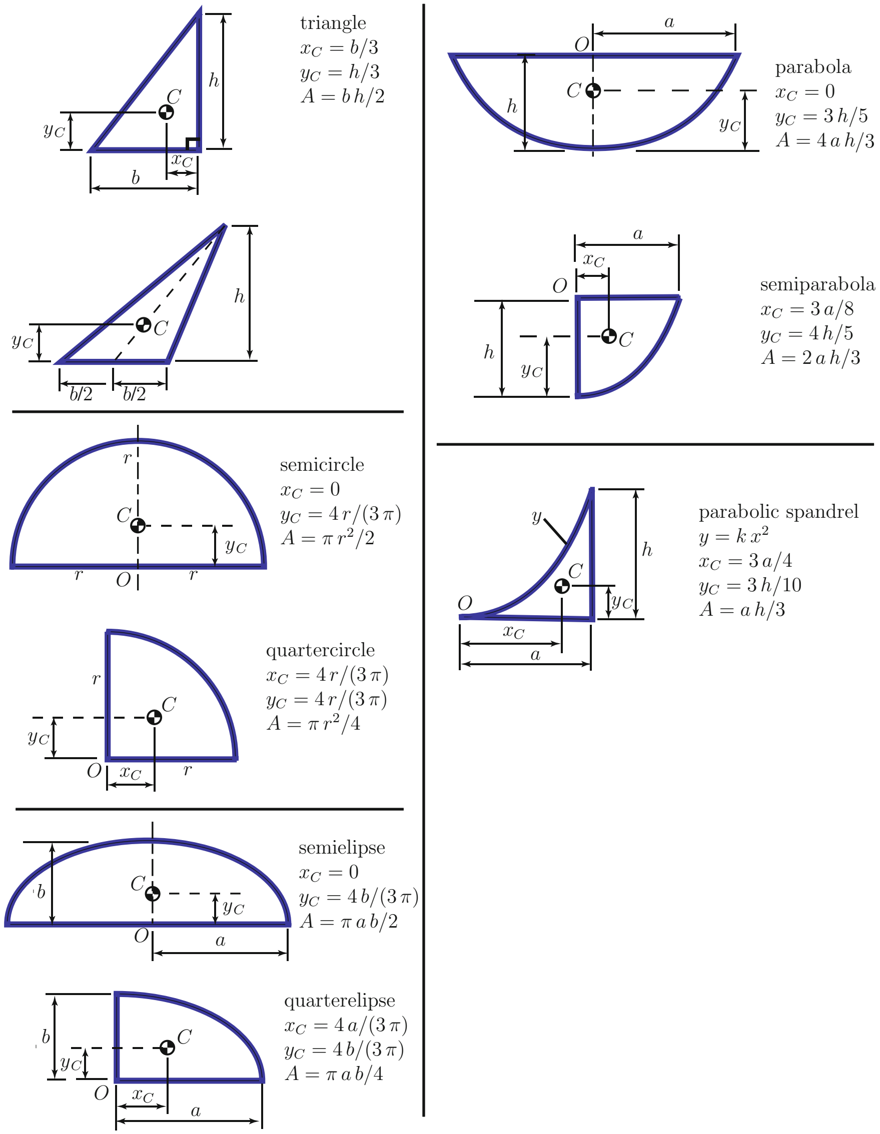



**Fig. 3.6** Coordinates of center of mass, $x_C$ and $y_C$, and area $A$

```
Mx=eval(int(y*dl,0,1));
yC=Mx/L;
```

The numerical values for the length and the centroid are:

```
L = 2.268 (m)
xC = 0.575 (m)
```

```
yC = 0.952 (m)
```

The MATLAB statements for the graphical representation are:

```
% plot the curve and CM
xf=1;
xn = 0:xf/100:xf;
yn = b*sqrt(xn.^a);
axis ([0 1 0 1])
plot(xn,yn,'-b','LineWidth',2)
hold on
plot(xC,yC,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
text(xC,yC,'     C','FontSize',18)
title('y=f(x)=2 x^{3/2}')
```

and the results are depicted in Fig. 3.7.

*Example 3.2* A homogeneous circle is given by the Cartesian equation $x^2 + y^2 = r^2$, where $r = 1$ m. (a) Find the length of the homogeneous circle. (b) Find the length and the position of the center of mass for the homogeneous semi-circle, $-1 \leq x \leq 1$ and $0 \leq y \leq 1$. (c) Find the length and the position of the center of mass for the homogeneous quarter-circle, $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

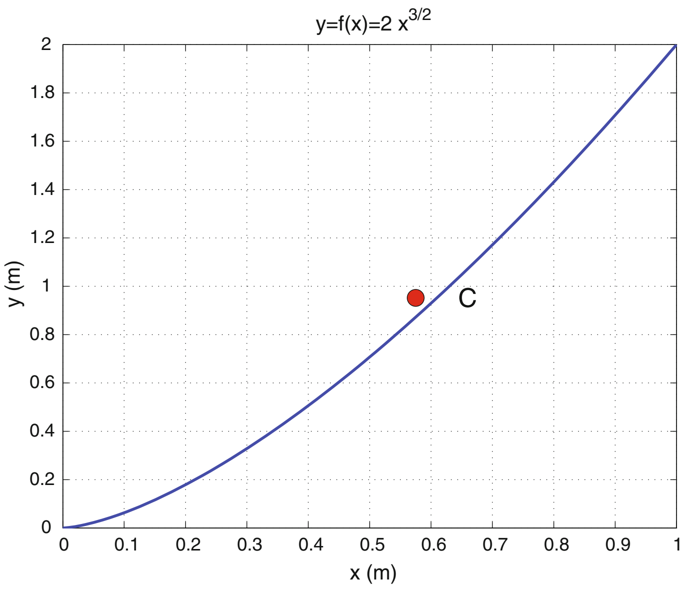



**Fig. 3.7** Example 3.1

*Solution* (a) The parametric equations for the circle are:

```
syms r t real
x = r*cos(t);
y = r*sin(t);
% 0 < t < 2*pi
% r > 0
```

The differential arc length is calculated in MATLAB with:

```
dx = diff(x,t);
dy = diff(y,t);
% dl = ((dx/dt)^2+(dy/dt)^2)^0.5 dt
dl = (dx^2+dy^2)^0.5;
dl = simplify(dl);
```

and the result is:

```
dl = abs(r) dt
```

The length of the circle is given by:

```
L = int(dl,t,0,2*pi);
```

and the MATLAB result is:

```
L = 2*pi*abs(r)
```

(b) For the semi-circle the length and the center of mass are:

```
Ls  = int(dl,t,0,pi);
Mys = int(x*dl,t,0,pi);
xCs = simplify(Mys/Ls);
Mxs = int(y*dl,t,0,pi);
yCs = simplify(Mxs/Ls);
```

The results are:

```
Ls = pi*abs(r)
xCs = 0
yCs = (2*r)/pi
```

(c) For the quarter-circle the length and the center of mass are:

```
Lq  = int(dl,t,0,pi/2);
Myq = int(x*dl,t,0,pi/2);
xCq = simplify(Myq/Lq);
Mxq = int(y*dl,t,0,pi/2);
yCq = simplify(Mxq/Lq);
```

and the MATLAB results are:

```
Lq = (pi*abs(r))/2
xCq = (2*r)/pi
yCq = (2*r)/pi
```

The MATLAB statements for the semi-circle and the quarter-circle graphical representation are:

```
rn=1;
% plot the semi-circle and CM
figure(1)
xCsn = subs(xCs,r,1);
yCsn = subs(yCs,r,1);
tn = 0:pi/18:pi;
xn = rn*cos(tn);
yn = rn*sin(tn);
axis manual
axis equal
hold on
grid on
sa = 1;
axis ([-sa sa -sa sa])
plot(xn,yn,'-b','LineWidth',2)
text(0,0,' O','fontsize',14)
line([-sa,sa],[0,0],'Color','k')
line([0,0],[0,sa],'Color','k')
plot(xCsn,yCsn,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
% plot the quarter-circle and CM
figure(2)
xCqn = subs(xCq,r,1);
yCqn = subs(yCq,r,1);
tn = 0:pi/18:pi/2;
xn = rn*cos(tn);
yn = rn*sin(tn);
axis manual
axis equal
hold on
grid on
sa = 1;
axis ([-sa sa -sa sa])
plot(xn,yn,'-b','LineWidth',2)
text(0,0,' O','fontsize',14)
line([0,sa],[0,0],'Color','k')
line([0,0],[0,sa],'Color','k')
plot(xCqn,yCqn,'o','MarkerSize',12,...
```

```
                   'MarkerEdgeColor','k',...
                   'MarkerFaceColor','r')
text(xCqn,yCqn,'  C','FontSize',18)
```

The graphics are depicted in Fig. 3.8.

*Example 3.3* A homogeneous quarter-astroid (one cusp) is given by the Cartesian equation $x^{2/3} + y^{2/3} = a^{2/3}$, where $a = 1$ m and $0 \leq x \leq 1$ . Find the length and the position of the center of mass for the homogeneous given curve.

*Solution* The parametric equations for the astroid are:

```
syms t real
a = 1; % (m)
x = a*cos(t)^3;
y = a*sin(t)^3;
% 0 < t < pi/2 - quarter-astroid
```

The differential arc length, $\mathrm{d}l = \sqrt{(\mathrm{d}x/\mathrm{d}t)^2 + (\mathrm{d}y/\mathrm{d}t)^2)}\,\mathrm{d}t$, is calculated in MATLAB with:

```
dx = diff(x,t);
dy = diff(y,t);
% dl=((dx/dt)^2+(dy/dt)^2)^0.5 dt
dl = (dx^2+dy^2)^0.5;
dl = simplify(dl);
```

and the result is:

```
dl = (3*(sin(2*t)^2)^(1/2))/2
```

The length of the quarter-astroid is given by:

```
L = int(dl,t,0,pi/2);
L = double(L);
```

and the MATLAB result is:

```
L = 6*a/4
L = 1.500 (m)
```

For the quarter-astroid the length and the center of mass are:

```
My = int(x*dl,t,0,pi/2);
xC = My/L;
xC = double(xC);
Mx = int(y*dl,t,0,pi/2);
yC = Mx/L;
yC = double(yC);
```

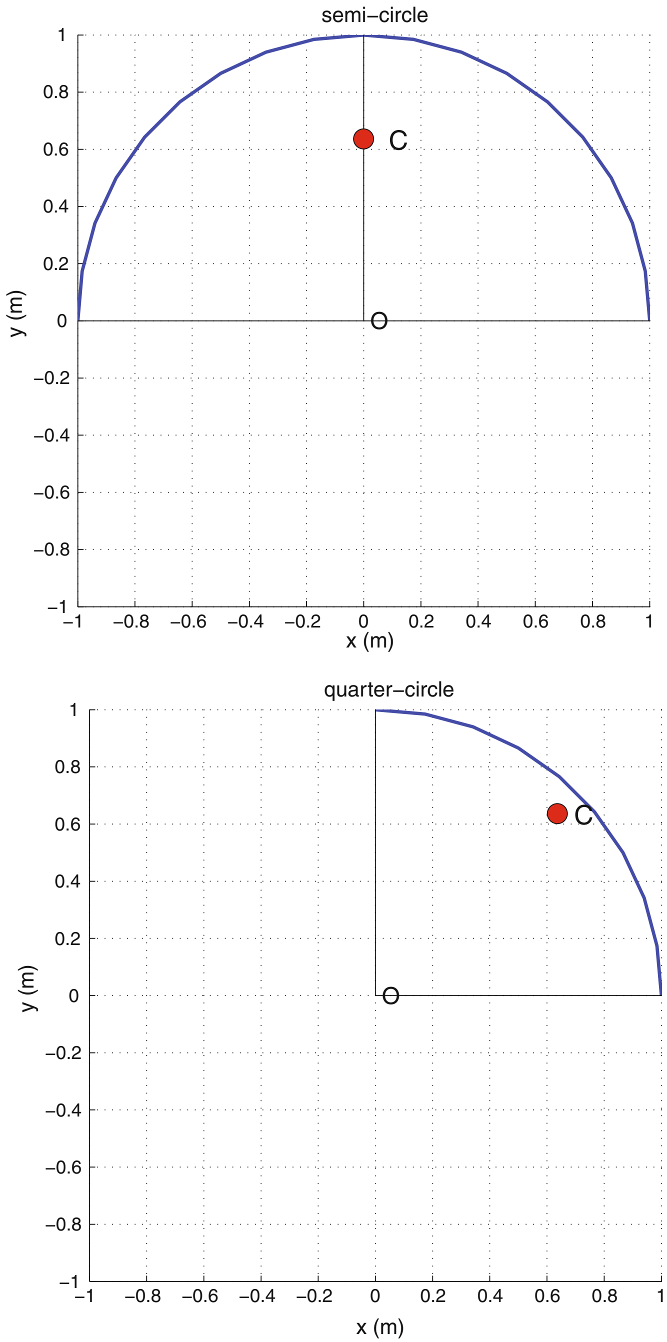



**Fig. 3.8** Example 3.2

and the MATLAB results are:

```
xC = 0.400 (m)
yC = 0.400 (m)
```

The MATLAB statements for the semi-circle and the quarter-circle graphical representation are:

```
tn = 0:pi/18:pi/2;
xn = a*cos(tn).^3;
yn = a*sin(tn).^3;
sa = 1;
axis ([0 sa 0 sa])
plot(xn,yn,'-b','LineWidth',2)
text(0,0,' O','fontsize',14)
line([0,sa],[0,0],'Color','k')
line([0,0],[0,sa],'Color','k')
plot(xC,yC,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
```

The graphics are depicted in Fig. 3.9.

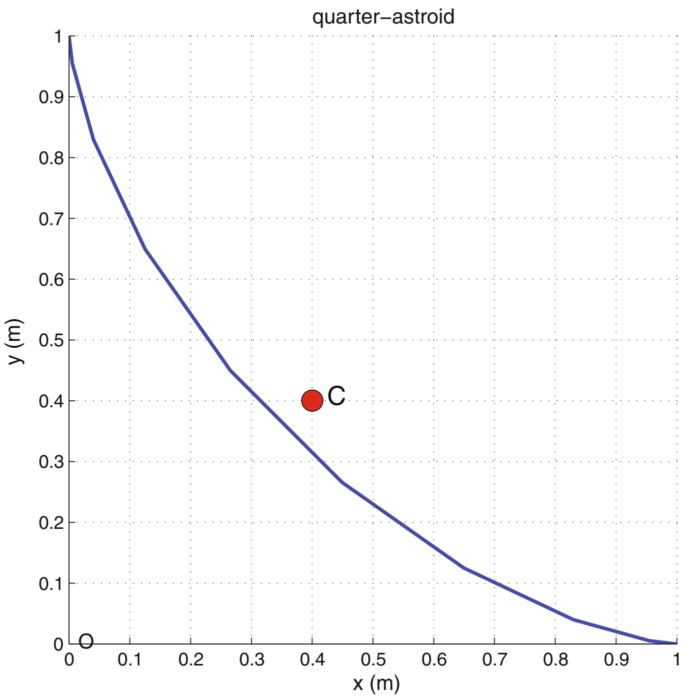



**Fig. 3.9** Example 3.3

*Example 3.4* A homogeneous circular helix is given by the Cartesian equation

$$x = a\cos t;\;\; y = a\sin t;\;\; \text{and } z = h\,t,$$

where $a = 1$ m is the radius of the helix and $2\,\pi\,h$ is the pitch of the helix, $h = 1$ m. Find the length and the position of the center of mass for the spatial homogeneous helix.

*Solution* The differential arc length for the spatial curve is

$$\mathrm{d}l = \sqrt{(\mathrm{d}x/\mathrm{d}t)^2 + (\mathrm{d}y/\mathrm{d}t)^2 + (\mathrm{d}z/\mathrm{d}t)^2}\,\mathrm{d}t$$

and is calculated in MATLAB with:

```
syms a h t real
x = a*cos(t);
y = a*sin(t);
z = h*t;
dx = diff(x,t);
dy = diff(y,t);
dz = diff(z,t);
%dl=((dx/dt)^2+(dy/dt)^2+(dz/dt)^2)^0.5 dt
dl = (dx^2+dy^2+dz^2)^0.5;
```

The MATLAB result for the differential arc length is:

```
dl = (a^2 + h^2)^(1/2) dt
```

The length of the helix is calculated with:

```
tf = 6*pi;
L = int(dl,t,0,tf);
```

and the coordinates of the center of mass are:

```
xC = int(x*dl,t,0,tf)/L;
yC = int(y*dl,t,0,tf)/L;
zC = int(z*dl,t,0,tf)/L;
```

The numerical results are:

```
L = 26.657 (m)
xC = 0.000 (m)
yC = 0.000 (m)
zC = 9.425 (m)
```

The MATLAB statements for the helix graphical representation are:

```
tn = 0:pi/50:tf;
plot3(sin(tn),cos(tn),tn)
```

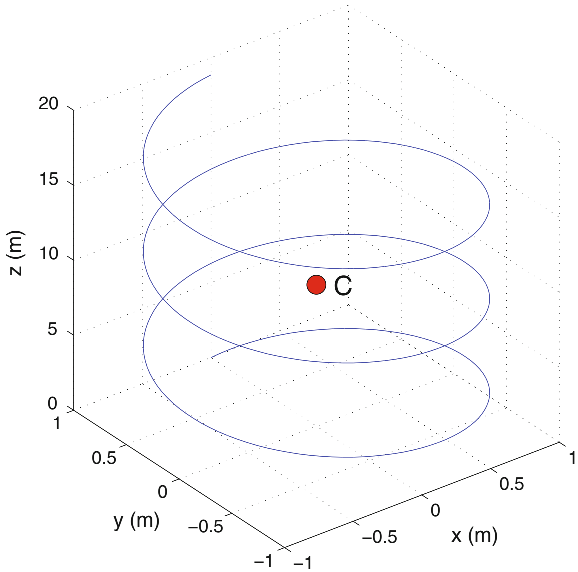



**Fig. 3.10** Example 3.4

```
plot3(xC,yC,zC,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
```

and the graphics are depicted in Fig. 3.10.

*Example 3.5* A homogeneous spatial curve is given by the Cartesian equation

$$x = ae^{kt} \cos t \text{ (m)}; \; y = ae^{kt} \sin t \text{ (m)}; \text{ and } z = ae^{kt} \text{ (m)},$$

where $a = 2$ and $k = 1$. Find the length and the position of the center of mass for the spatial homogeneous curve for $t \in [0, \; 3]$.

*Solution* The differential arc length for the spatial curve is

$$\mathrm{d}l = \sqrt{(\mathrm{d}x/\mathrm{d}t)^2 + (\mathrm{d}y/\mathrm{d}t)^2 + (\mathrm{d}z/\mathrm{d}t)^2}\,\mathrm{d}t$$

and is calculated in MATLAB with:

```
syms t real
a=2; k=1;
x = a*exp(k*t)*cos(t);
y = a*exp(k*t)*sin(t);
```

```
z = a*exp(k*t);
dx = diff(x,t);
dy = diff(y,t);
dz = diff(z,t);
dl = (dx^2+dy^2+dz^2)^0.5;
```

The MATLAB result for the differential arc length is:

```
dl = 2*3^(1/2)*exp(t) dt
```

The length of the helix is calculated with:

```
tf = 3;
L = int(dl,0,tf);
```

and the numerical value is `L = 66.114 (m)`. The coordinates of the center of mass are:

```
xC = int(x*dl,t,0,tf)/L;
yC = int(y*dl,t,0,tf)/L;
zC = int(z*dl,t,0,tf)/L;
```

The numerical values for $C$ are:

```
xC = -15.590 (m)
yC = 10.778 (m)
zC = 21.086 (m)
```

The MATLAB statements for the curve graphical representation are:

```
tn = 0:pi/100:tf;
xn = a*exp(k*tn).*cos(tn);
yn = a*exp(k*tn).*sin(tn);
zn = a*exp(k*tn);
ht = plot3(xn,yn,zn);
hold on
plot3(xC,yC,zC,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
text(xC,yC,zC,'   C','FontSize',18)
grid on
axis square
```

and the graphics are depicted in Fig. 3.11.

*Example 3.6* Find the coordinates of the mass center for a homogeneous planar plate located under the line of equation $y = bx/a$ from $x = 0$ to $x = a$. For the numerical application select $a = b = 1$ m.

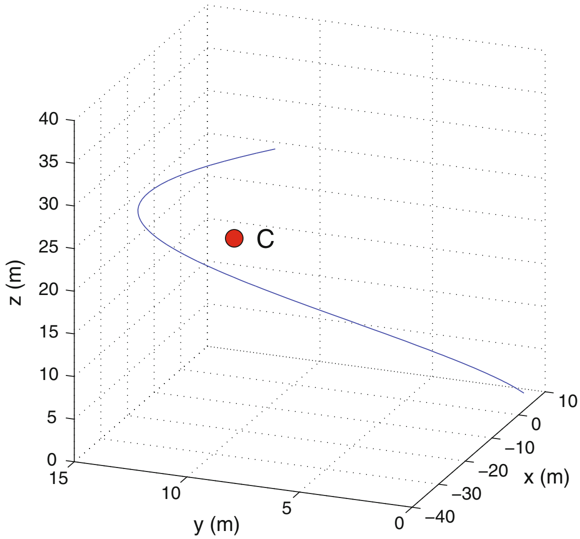



**Fig. 3.11** Example 3.5

*Solution* The differential element of area is $\mathrm{d}A = \mathrm{d}x\,\mathrm{d}y$ and the area of the figure is

$$A = \int_A \mathrm{d}x\,\mathrm{d}y = \int_0^a \int_0^{bx/a} \mathrm{d}x\,\mathrm{d}y = \int_0^a \mathrm{d}x \int_0^{bx/a} \mathrm{d}y$$
$$= \int_0^a \mathrm{d}x\ \{y\}_0^{bx/a} = \int_0^a (bx/a)\mathrm{d}x = \left\{b\,x^2/(2\,a)\right\}_0^a = b\,a/2.$$

The MATLAB program for the area is given by:

```
syms x y a b real
f = b*x/a;
xf = a;
Ay = int(1,y,0,f);
Area = int(Ay,x,0,xf);
```

The first moment of the area $A$ about the $y$ axis is

$$M_y = \int_A x\,\mathrm{d}A = \int_0^a \int_0^{bx/a} x\,\mathrm{d}x\,\mathrm{d}y = \int_0^a x\,\mathrm{d}x \int_0^{bx/a} \mathrm{d}y$$
$$= \int_0^a x\,\mathrm{d}x\ \{y\}_0^{bx/a} = \int_0^a x\,(bx/a)\mathrm{d}x = \int_0^a (bx^2/a)\mathrm{d}x = b\,a^2/3.$$

The $x$ coordinate of the mass center is $x_C = M_y/A = 2a/3 = 0.667$ m. The MATLAB program for $x_C$ is :

```
% first moment of area about y-axis
% My = int(x dx dy) where
% 0<x<xf and 0<y<f
% Qyy = int(dy) ; 0<y<f
Qyy = int(1,y,0,f);
% My = int(x Qyy dx) where 0<x<xf
My = int(x*Qyy,x,0,xf);
% centroid xC = My/Area
xC = My/Area;
```

The $y$ coordinate of the mass center is $y_C = M_x/A$, where the first moment of the area $A$ about the $x$ axis is

$$M_x = \int_A y\,\mathrm{d}A = \int_0^a \int_0^{bx/a} y\,\mathrm{d}x\,\mathrm{d}y = \int_0^a \mathrm{d}x \int_0^{bx/a} y\,\mathrm{d}y$$
$$= \int_0^a \mathrm{d}x \left\{\frac{y^2}{2}\right\}_0^{bx/a} = \int_0^a \frac{b^2 x^2}{2a^2}\mathrm{d}x = \frac{b^2}{2a^2}\int_0^a x^2\,\mathrm{d}x = \frac{b^2 a}{6}.$$

The coordinate $y_C$ is

$$y_C = \frac{M_x}{A} = \frac{b}{3} = 0.333 \text{ m}.$$

The MATLAB program for $y_C$ is :

```
Qxy = int(y,y,0,f);
% Mx = int(x Qxy dx) where 0<x<xf
Mx = int(Qxy,x,0,xf);
yC = Mx/Area;
```

The MATLAB statements for the graphical representation are:

```
ls = {a,b};
ln = {1,1};
xfn = subs(xf,ls,ln);
xCn = subs(xC,ls,ln);
yCn = subs(yC,ls,ln);
sa = 1.5;
axis([0 sa 0 sa])
hold on, grid on
xx = 0:.1:xfn;
fx = subs(f,{a,b,x},{1,1,xx});
plot(xx,fx,'--','LineWidth',2)
```

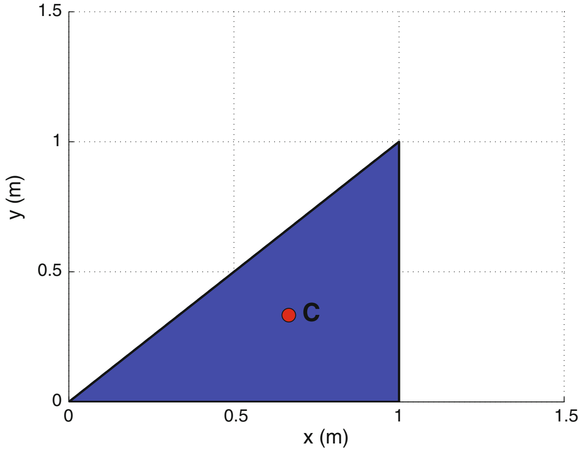



**Fig. 3.12** Example 3.6

```
hold on
area(xx,fx,'FaceColor','b',...
           'EdgeColor','k',...
           'LineWidth',2)
hold on
plot(xCn,yCn,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
xlabel('x'), ylabel('y')
text(xCn,yCn,'C','fontsize',14,'fontweight','b')
```

and the graphic is shown in Fig. 3.12.

*Example 3.7* Find the coordinates of the mass center for a homogeneous planar plate located under the curve of equation $y = A\ \sin(kx)$ from $x = 0$ to $x = 3\,\pi/(4\,k)$. For the numerical application use $A = 1.5$ m and $k = 0.75$ m$^{-1}$.

*Solution* The differential element of area is $\mathrm{d}A = \mathrm{d}x\,\mathrm{d}y$ and the area of the figure is

$$Area = \int_A \mathrm{d}x\,\mathrm{d}y = \int_0^{3\,\pi/(4\,k)} \int_0^{A\sin(k\,x)} \mathrm{d}x\,\mathrm{d}y = \int_0^{3\,\pi/(4\,k)} \mathrm{d}x \int_0^{A\ \sin(k\,x)} \mathrm{d}y$$
$$= \int_0^{3\,\pi/(4\,k)} \mathrm{d}x\,\{y\}_0^{\sin\,x} = \int_0^{3\,\pi/(4\,k)} A\sin(k\,x)\mathrm{d}x.$$

The MATLAB program for the area is given by:

```
syms x y A k real
% f(x) = y(x) = A*sin(k*x);
f = A*sin(k*x);
% 0 < x < xf
xf = (3*pi/4)/k;
% Area = int(dx dy) where
% 0<x<xf and 0<y<f
% Ay = int(dy) where 0<y<f
Ay = int(1,y,0,f);
% Ay = A*sin(k*x)
% Area = int(Ay dx) where 0<x<xf
Area = int(Ay,x,0,xf);
```

and the result is:

```
Area =
      1/2
  A (2     + 2)
  ------------
      2 k
```

The first moment of the area about the $y-$axis is

$$M_y = \int_A x\,\mathrm{d}A = \int_0^{3\pi/(4k)} \int_0^{A\,\sin(k\,x)} x\,\mathrm{d}x\,\mathrm{d}y = \int_0^{3\pi/(4k)} x\,\mathrm{d}x \int_0^{A\,\sin(k\,x)} \mathrm{d}y$$
$$= \int_0^{3\pi/(4k)} x\,\mathrm{d}x\,\{y\}_0^{A\,\sin(k\,x)} = \int_0^{3\pi/(4k)} A\,x\,\sin(k\,x)\mathrm{d}x,$$

With MATLAB the first moment of the area about the $y-$axis is:

```
Qyy = int(1,y,0,f);
My = int(x*Qyy,x,0,xf);
```

and the symbolic result is:

```
My =
   1/2
  2     A (3 pi + 4)
  -----------------
           2
        8 k
```

The $x$ coordinate of the mass center is $x_C = M_y/Area$:

```
xC =
   1/2
  2    (3 pi + 4)
  ---------------
          1/2
  4 k (2    + 2)

xC = 1.854 (m)
```

The first moment of the area $A$ about the $x-$axis is

$$M_x = \int_A y\,\mathrm{d}A = \int_0^{3\pi/(4k)} \int_0^{A\,\sin(kx)} y\,\mathrm{d}x\,\mathrm{d}y = \int_0^{3\pi/(4k)} \mathrm{d}x \int_0^{A\,\sin(kx)} y\,\mathrm{d}y$$
$$= \int_0^{3\pi/(4k)} \mathrm{d}x \left\{\frac{y^2}{2}\right\}_0^{A\,\sin(kx)} = \int_0^{3\pi/(4k)} \frac{A^2\,\sin^2(kx)}{2}\mathrm{d}x.$$

The first moment of the area about the $x-$axis in MATLAB is calculated with:

```
Qxy = int(y,y,0,f);
Mx = int(Qxy,x,0,xf);
```

and the symbolic result is:

```
Mx =
   2
  A  (3 pi + 2)
  -------------
      16 k
```

The $y$ coordinate of the mass center is $y_C = M_x/Area$:

```
yC =
  A (3 pi + 2)
  ------------
      1/2
  8 (2    + 2)

yC = 0.627 (m)
```

The MATLAB statements for the graphical representation are:

```
A = 1.5;  % m
k = 0.75; % m^(-1)
sa = 4;
axis([0 sa  0 sa])
hold on, grid on
```

```
xx = 0:.1:xfn;
fx = A*sin(k*xx);
plot(xx,fx,'--','LineWidth',2)
hold on
area(xx,fx,'FaceColor','b',...
           'EdgeColor','k',...
           'LineWidth',2)
hold on
plot(xCn,yCn,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
xlabel('x (m)'), ylabel('y (m)')
text(xCn,yCn,'   C',...
'fontsize',14,'fontweight','b')
```

and the graphic is shown in Fig. 3.13.

*Example 3.8* Find the coordinates of the centroid of the region bounded by the curves $y_1(x) = x/4$ and $y_2(x) = \sqrt{2\,(x-3)}$, $x_1 \leq x \leq x_2$, as shown in Fig. 3.14. All coordinates are in meters.

*Solution* The two curves will have two intersection points calculated in MATLAB with:

```
syms x y real
y1 = x/4;
y2 = sqrt(2*(x-3));
```

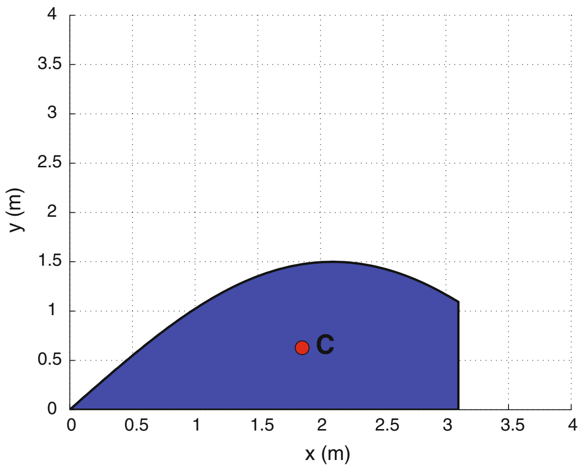



**Fig. 3.13** Example 3.7

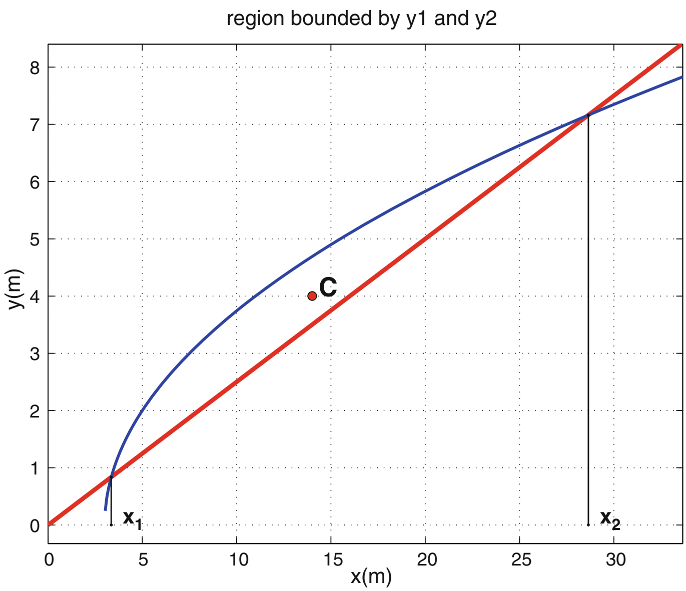



**Fig. 3.14** Example 3.8

```
sol = eval(solve(y2-y1));
if sol(2) > sol(1)
x1 = sol(1); x2 = sol(2);
else
x1 = sol(2); x2 = sol(1);
end
y11 = subs(y1,x,x1);
y12 = subs(y1,x,x2);
```

The $x$ values of the intersection points are:

```
x1 = 3.351 (m)
x2 = 28.649 (m)
```

The graphic shown in Fig. 3.14 is plotted with:

```
axis equal
g1=ezplot(y1,[0,x2+5])
set(g1, 'Color', 'r','LineWidth',3)
hold on
g2=ezplot(y2,[0,x2+5])
set(g2, 'Color', 'b','LineWidth',2)
hold on
line([x1 x1],[0 y11],...
    'Color','k','LineWidth',1,...
```

```
    'Marker','.','LineStyle','-')
hold on
line([x2 x2],[0 y12],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')
hold on
grid on
title('region bounded by y1 and y2')
xlabel('x(m)'), ylabel('y(m)')
```

The area of the region is calculated with:

$$A = \int_A \mathrm{d}x\,\mathrm{d}y = \int_{x_1}^{x_2}\int_{y_1}^{y_2} \mathrm{d}x\,\mathrm{d}y = \int_{x_1}^{x_2} \mathrm{d}x \int_{x/4}^{\sqrt{2\,(x-3)}} \mathrm{d}y$$
$$= \int_{x_1}^{x_2} \left(\sqrt{2\,(x-3)} - x/4\right)\,\mathrm{d}x.$$

The command in MATLAB for calculating the area is:

```
A = double(int(int(1,y1,y2),x1,x2));
```

and the numerical result is:

```
A = 21.082 (m^2)
```

The first moment of the area about the $y-$axis is

$$M_y = \int_A x\,\mathrm{d}A = \int_{x_1}^{x_2}\int_{y_1}^{y_2} x\,\mathrm{d}x\,\mathrm{d}y = \int_{x_1}^{x_2} x\,\mathrm{d}x \int_{y_1}^{y_2} \mathrm{d}y$$
$$= \int_{x_1}^{x_2} x\,\mathrm{d}x\,\{y\}_{y_1}^{y_2} = \int_{x_1}^{x_2} x\,\left(\sqrt{2\,(x-3)} - x/4\right)\,\mathrm{d}x.$$

With MATLAB the first moment of the area about the $y-$axis and the $x$ coordinate of the mass center $x_C = M_y/A$ are:

```
Qyy = int(1, y, y1, y2);
My = int(x*Qyy, x, x1, x2);
xC = eval(My/A);
```

and the result is:

```
xC = 14.000 (m)
```

The first moment of the area $A$ about the $x-$axis is

$$M_x = \int_A y\,\mathrm{d}A = \int_{x_1}^{x_2}\int_{y_1}^{y_2} y\,\mathrm{d}x\,\mathrm{d}y = \int_{x_1}^{x_2}\mathrm{d}x\int_{y_1}^{y_2} y\,\mathrm{d}y$$
$$= \int_{x_1}^{x_2}\mathrm{d}x\left\{\frac{y^2}{2}\right\}_{y_1}^{y_2} = \int_{x_1}^{x_2}\frac{1}{2}\left[2\,(x-3)-\frac{x^2}{16}\right]\mathrm{d}x.$$

The first moment $M_x$ and $y_C$ are calculated in MATLAB with:

```
Qxy = int(y, y, y1, y2);
Mx = int(Qxy, x, x1, x2);
yC = eval(Mx/A);
```

and the result is:

```
yC = 4.000 (m)
```

*Example 3.9* Find the position of the center of mass the region defined by *OABDEF* as shown in Fig. 3.15, where $EF = DB = a = 4$ m and $AB = DE = b = 2$ m. The material is homogeneous.

*Solution*

The region is bounded by the lines of equations $y_1(x) = 2\,b$ for $0 \le x \le a$, $y_2(x) = b$ for $a < x \le 2\,a$ and the $x-$axis. The area of the region is given by

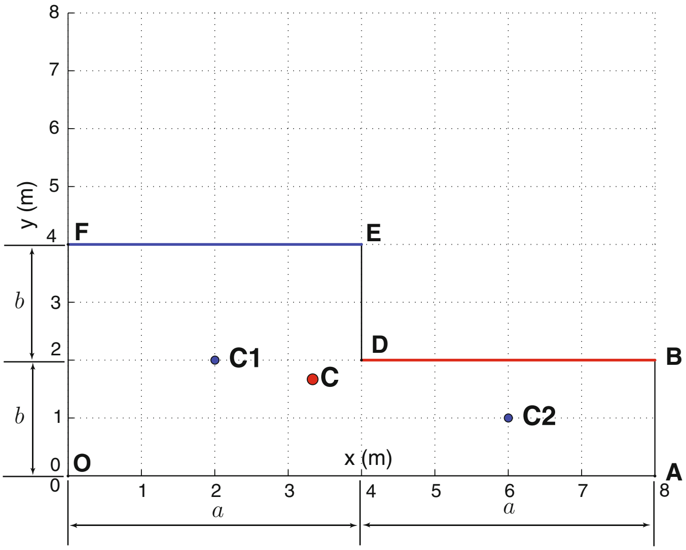



**Fig. 3.15** Example 3.9

$$A = A_1 + A_2 = \int_0^a y_1 \, \mathrm{d}x + \int_a^{2a} y_2 \, \mathrm{d}x = \int_0^a (2\,b) \, \mathrm{d}x + \int_a^{2a} b \, \mathrm{d}x$$
$$= 2\,b\,a + a\,b = 3\,a\,b = 24 \text{ m}^2.$$

The first moment of the area about the $y-$axis for the composite region is

$$M_y = \int_A x \, \mathrm{d}A = \int_0^a x\,y_1 \, \mathrm{d}x + \int_a^{2\,a} x\,y_2 \, \mathrm{d}x = \int_0^a 2\,bx \, \mathrm{d}x + \int_a^{2\,a} bx \, \mathrm{d}x.$$

With MATLAB the first moment of the area about the $y-$axis and the $x$ coordinate of the mass center $x_C = M_y/A$ are:

```
syms x a b  real
sl = {a,b};
sn = {4,2};
y1 = 2*b;
y2 = b;
A1 = int(y1,x,0,a);
A2 = int(y2,x,a,2*a);
A = A1+A2;
Mx1 = int(x*y1,x,0,a);
Mx2 = int(x*y2,x,a,2*a);
xC1 = Mx1/A1;
xC2 = Mx2/A2;
xC = (Mx1+Mx2)/A;
```

The results are:

```
My =
          2
   5 a b
   ------
     2

xC =
   5 a
   ---
    6

xC = 3.333 (m)
```

The first moment of the area about the $x-$axis is calculated with the general formula

$$M_x = 0.5 \int_{x_1}^{x_2} y^2(x)\,\mathrm{d}x,$$

and for $A = A_1 + A_2$ it results

$$M_x = M_{x_1} + M_{x_2} = 0.5 \int_0^a {y_1}^2\,\mathrm{d}x + 0.5 \int_a^{2a} {y_2}^2\,\mathrm{d}x$$
$$= 0.5 \int_0^a (2\,b)^2\,\mathrm{d}x + 0.5 \int_a^{2a} b^2\,\mathrm{d}x.$$

The first moment $M_x$ and $y_C$ are calculated in MATLAB with:

```
My1 = 0.5*int(y1^2,x,0,a);
My2 = 0.5*int(y2^2,x,a,2*a);
My = My1 + My2;
yC1 = My1/A1;
yC2 = My2/A2;
yC = (My1+My2)/A;
```

and the result are:

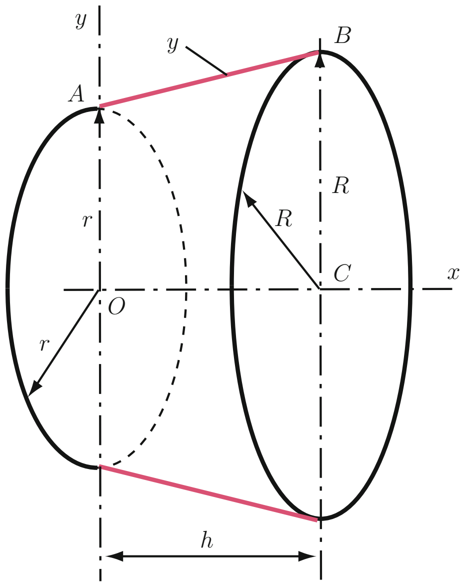
```
Mx =
      2
  5 a  b
  ------
    2

yC =
  5 b
  ---
   6

yC = 1.667 (m)
```

*Example 3.10* Find the volume of the frustum of a cone, shown in Fig. 3.16, where $h = 2$ m is the height, $R = 2$ m is the radius of large base, and $r = 1$ m radius of small base. The material is homogeneous.

*Solution* The formula for calculating the volume is:

$$V = \pi \int_a^b f^2(x)\mathrm{d}x$$

**Fig. 3.16** Example 3.10




where $y = f(x)$ is the generating equation of the planar curve. For the frustum of a cone the generating equation is

$$y = f(x) = f = \frac{(R-r)\,x}{h} + r,$$

and the MATLAB program is;

```
 syms R r h x  real
f = (R-r)*x/h+r;
V = pi*int(f^2,x,0,h);
ls = {R,r,h};
ln = {2,1,2};
fn = subs(f,ls,ln);
Vn = subs(V,ls,ln);
```

The results are:

```
V =
         2          2
   pi h (R  + R r + r )
   --------------------
            3
 V = 14.661 (m^3)
```

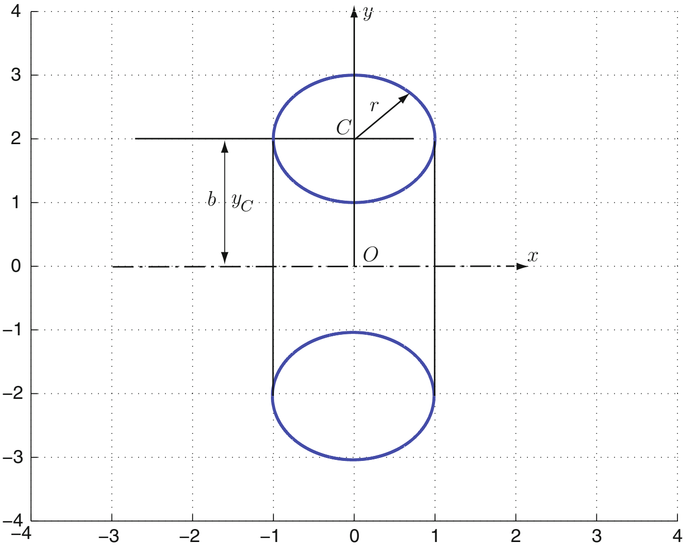



**Fig. 3.17** Example 3.11

*Example 3.11* Find the volume and surface area of the complete torus of circular cross section of radius $r = 1$ m as shown in Fig. 3.17 where $b = 2$ m.

*Solution* The torus is generated by revolving the circular area of radius $r$ through 360° about the $x-$axis. With the first theorem of Guldinus-Pappus, the surface of revolution is $S = 2\,\pi\, y_C\, L$, where $L = 2\,\pi\, r$ is the length of generating circle and $y_C = b$ is the centroid of generating circle

$$S = 2\,\pi\, b\, 2\,\pi\, r = 4\,\pi^2\, b\, r = 78.957 \text{ m}^2.$$

The second theorem of Guldinus-Pappus gives the volume of revolution $V = 2\,\pi\, y_C A$, where $A = \pi\, r^2$ is the area of generating circular surface and $y_C = b$ is the centroid of generating circular surface

$$V = 2\,\pi\, b\,\pi\, r^2 = 2\,\pi^2\, b\, r^2 = 39.478 \text{ m}^3.$$

The equation of the generating circle is

$$x^2 + (y - b)^2 - r^2 = 0.$$

The volume of the torus can also be calculated with the formula

$$V = \pi \int_a^b f^2(x)\mathrm{d}x = \pi \int_{-r}^{r} (f_2^2 - f_1^2)\mathrm{d}x,$$

where

$$f_1 = b - \sqrt{r^2 - x^2} \text{ and } f_2 = b + \sqrt{r^2 - x^2},$$

and the MATLAB program is:

```
syms b r x  real
f1 = b-sqrt(r^2-x^2);
f2 = b+sqrt(r^2-x^2);
V = pi*int(f2^2-f1^2,x,-r,r);
```

## 3.8 Problems

3.1 Locate the centroid of the uniform wire bent in the shape shown in Fig. 3.18. For the numerical application use $r = 1$ m, $a = 2$ m, and $b = 1.75$ m.

3.2 Find the location of the centroid $C$ of the uniform area shown in Fig. 3.19 where $a = 0.4$ m, $b = 0.8$ m, and $c = 0.6$ m.

3.3 Find the location of the centroid of the area shown in Fig. 3.20. For the numerical application use $r = 0.1$ m and $h = 0.2$ m.

3.4 Determine the location of the centroid of the uniform area shown in Fig. 3.21. For the numerical application use $a = 0.2$ m, $b = 0.25$ m, and $c = 0.27$ m.

3.5 Find the location of the centroid of the uniform area shown in Fig. 3.22. For the numerical application use $a = 0.6$ m, $b = 0.4$ m, and $c = 0.3$ m.

3.6 Locate the centroid of the volume shown in Fig. 3.23, where $r = 0.5$ m, and $h = 1.2$ m. The material is homogeneous.

3.7 Locate the centroid of the volume shown in Fig. 3.24, where $r = 0.3$ m, and $h = 0.9$ m. The material is homogeneous.

3.8 Locate the centroid of the homogeneous volume shown in Fig. 3.25, where $R = 0.6$ m, $r = 0.4$ m, $a = 0.5$ m, and $b = 0.6$ m. The material is homogeneous.

3.9 Locate the centroid of the volume shown in Fig. 3.26, where $R = 0.7$ m, $r = 0.4$ m, $p = 0.5$ m, $a = 0.4$ m, and $b = 0.5$ m. The material is homogeneous.

3.10 Find the centroid of the volume depicted in Fig. 3.27, where $r = 25$ mm, $a = 200$ m, $b = 100$ mm, and $t = 15$ mm. The material is homogeneous.

3.11 Find the centroid of the volume shown in Fig. 3.28, where $a = 200$ mm, $b = 150$ mm, and $t = 30$ mm. The material is homogeneous.

3.12 Find the centroid of the volume shown in Fig. 3.29, where $a = 400$ mm, $b = 200$ mm, and $c = 100$ mm. The material is homogeneous.

3.13 Find the centroid of the volume shown in Fig. 3.30, where $a = 100$ mm, $b = 125$ mm, $c = 150$ mm, and $t = 25$ mm. The material is homogeneous.

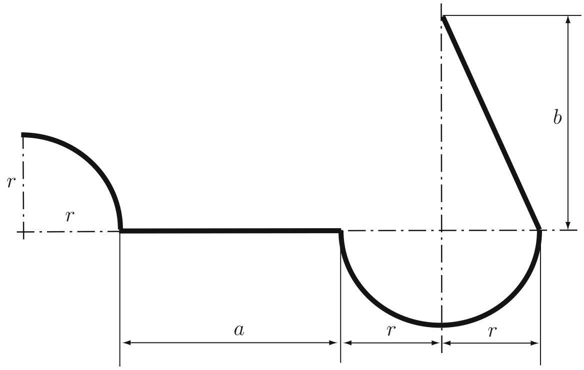



**Fig. 3.18** Problem 3.1

**Fig. 3.19** Problem 3.2

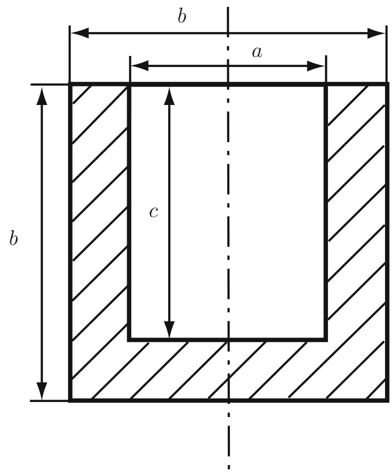
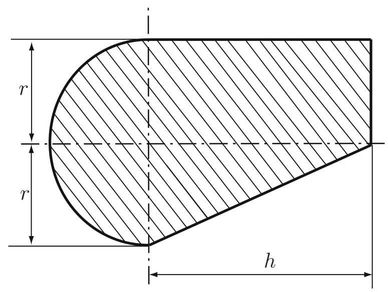



**Fig. 3.20** Problem 3.3



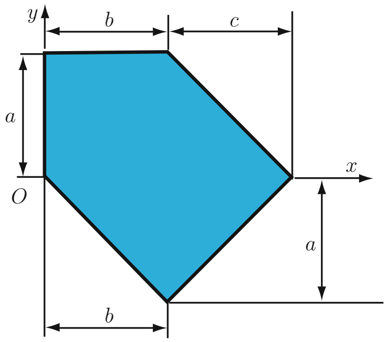



**Fig. 3.21** Problem 3.4

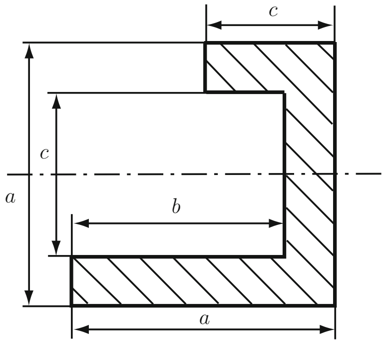



**Fig. 3.22** Problem 3.5

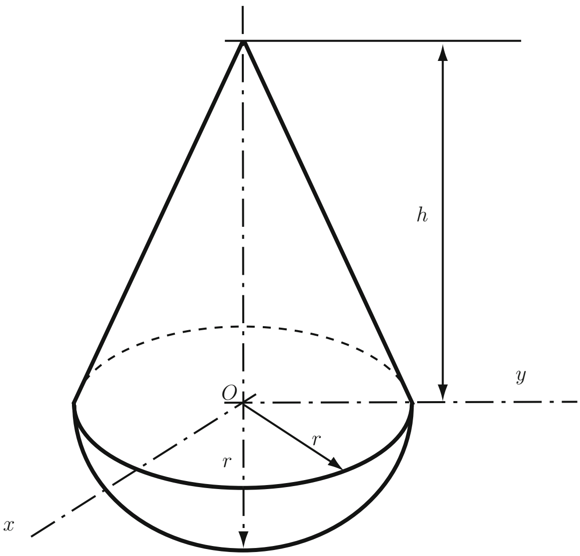



**Fig. 3.23** Problem 3.6

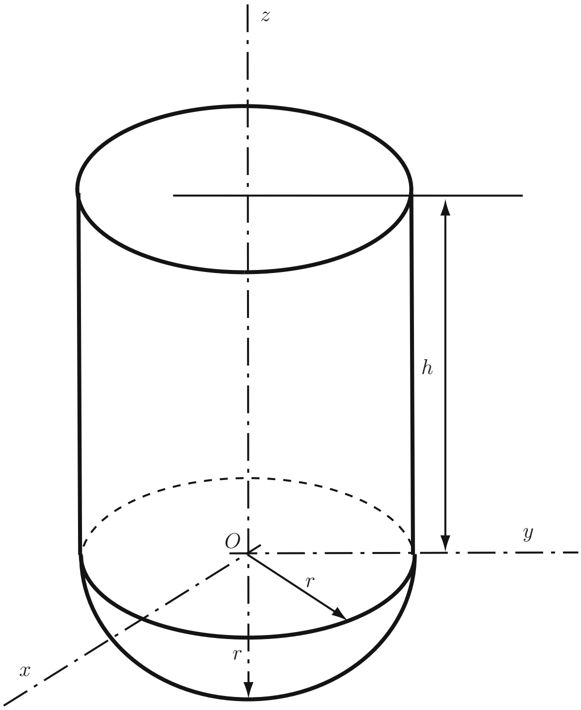



**Fig. 3.24** Problem 3.7

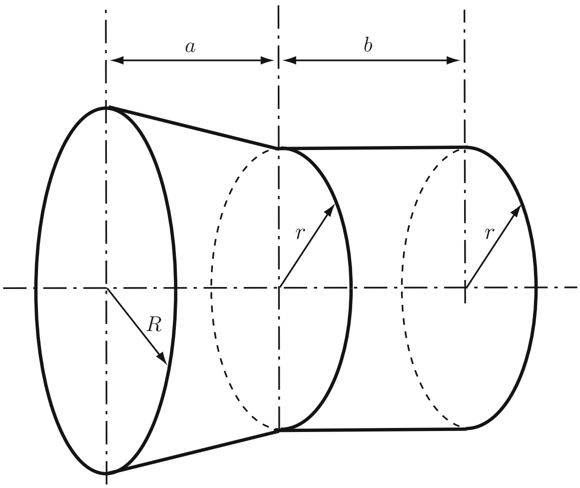



**Fig. 3.25** Problem 3.8

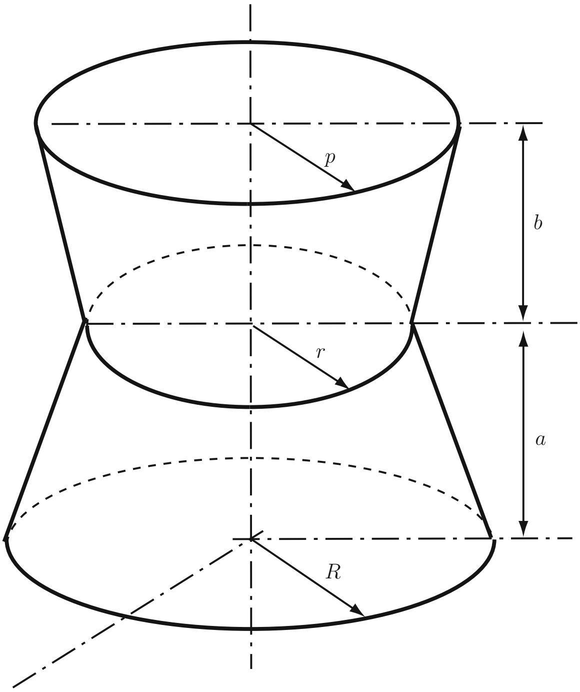



**Fig. 3.26** Problem 3.9

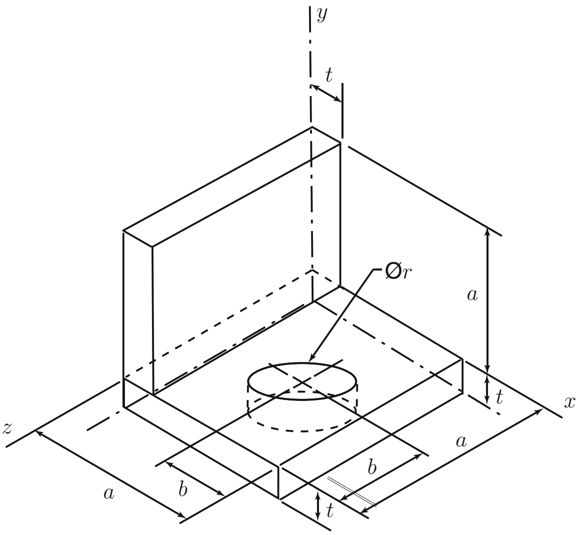



**Fig. 3.27** Problem 3.10

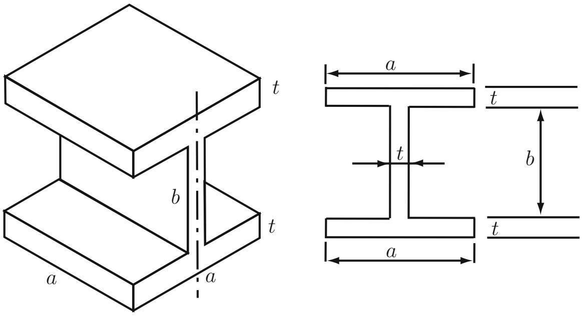



**Fig. 3.28** Problem 3.11

**Fig. 3.29** Problem 3.12

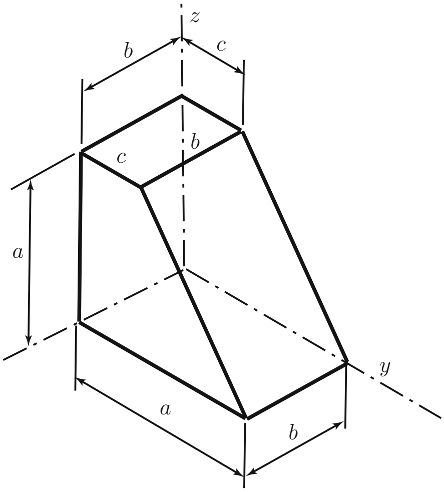
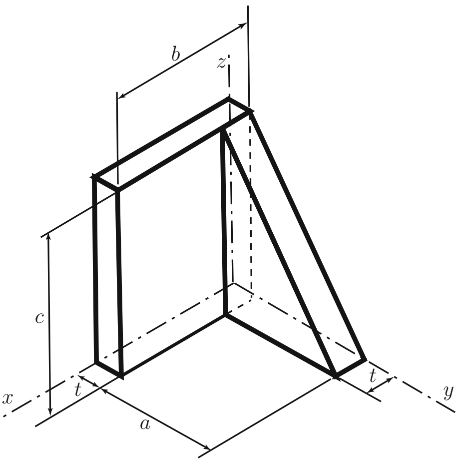



**Fig. 3.30** Problem 3.13



3.14 Find the coordinates of the centroid of the region is bounded by the curves $y = x$ and $y = \sqrt{x}$ where $0 \leq x \leq 1$. All coordinates may be treated as dimensionless.

3.15 Determine the coordinates of the centroid of the region is bounded by the curves $y = x^2$ and $y = \sqrt{x}$ where $0 \leq x \leq 1$. All coordinates may be treated as dimensionless.

## 3.9 Programs

### *3.9.1 Program 3.1*

```
% example 3.1
% center of mass of a curve
clear all; clc; close all;
syms x real

a = 3;
b = 2;
y  = b*sqrt(x^a);
dy = diff(y,x);
dl =sqrt(1+dy^2);
% ds = (1+ (dy/dx)^2)^0.5
%  0< x < 1
L = eval(int(dl,0,1));
My=eval(int(x*dl,0,1));
xC=My/L;
Mx=eval(int(y*dl,0,1));
yC=Mx/L;
fprintf('L = %4.3f (m) \n', L)
fprintf('xC = %4.3f (m) \n', xC)
fprintf('yC = %4.3f (m) \n', yC)

% plot the curve and CM
xf=1;
xn = 0:xf/100:xf;
yn = b*sqrt(xn.^a);
axis ([0 1 0 1])
plot(xn,yn,'-b','LineWidth',2)
hold on
plot(xC,yC,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
```

```
text(xC,yC,'       C','FontSize',18)
title('y=f(x)=2 x^{3/2}')
grid on
xlabel('x(m)')
ylabel('y(m)')

% end of program
```

### 3.9.2 Program 3.2

```
% example 3.2
% center of mass
% circle (Cartesian equation)
% x^2+y^2=1
% semi-circle
% quarter-circle
clear all; clc; close all;
syms r t real

% parametric equation
x = r*cos(t);
y = r*sin(t);
% 0 < t < 2*pi
% r > 0
dx = diff(x,t);
dy = diff(y,t);
% arc length
% dl = ((dx/dt)^2+(dy/dt)^2)^0.5 dt

dl = (dx^2+dy^2)^0.5;
dl = simplify(dl);
fprintf('dl = %s dt \n', char(dl));
L = int(dl,t,0,2*pi);
fprintf('L = %s \n', char(L))
fprintf('\n');

% semi-circle
Ls  = int(dl,t,0,pi);
Mys = int(x*dl,t,0,pi);
xCs = simplify(Mys/Ls);
Mxs = int(y*dl,t,0,pi);
yCs = simplify(Mxs/Ls);
fprintf('Ls = %s \n', char(Ls))
```

```
fprintf('xCs = %s \n', char(xCs))
fprintf('yCs = %s \n', char(yCs))
fprintf('\n');

% quarter-circle
Lq  = int(dl,t,0,pi/2);
Myq = int(x*dl,t,0,pi/2);
xCq = simplify(Myq/Lq);
Mxq = int(y*dl,t,0,pi/2);
yCq = simplify(Mxq/Lq);
fprintf('Lq = %s \n', char(Lq))
fprintf('xCq = %s \n', char(xCq))
fprintf('yCq = %s \n', char(yCq))

rn=1;
% plot the semi-circle and CM
figure(1)
xCsn = subs(xCs,r,1);
yCsn = subs(yCs,r,1);

tn = 0:pi/18:pi;
xn = rn*cos(tn);
yn = rn*sin(tn);
axis manual
axis equal
hold on
grid on
sa = 1;
axis ([-sa sa -sa sa])
plot(xn,yn,'-b','LineWidth',2)
text(0,0,' O','fontsize',14)
line([-sa,sa],[0,0],'Color','k')
line([0,0],[0,sa],'Color','k')
plot(xCsn,yCsn,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
text(xCsn,yCsn,'   C','FontSize',18)
title('semi-circle')
xlabel('x(m)')
ylabel('y(m)')

% plot the quarter-circle and CM

figure(2)
xCqn = subs(xCq,r,1);
```

```
yCqn = subs(yCq,r,1);

tn = 0:pi/18:pi/2;
xn = rn*cos(tn);
yn = rn*sin(tn);
axis manual
axis equal
hold on
grid on
sa = 1;
axis ([-sa sa -sa sa])
plot(xn,yn,'-b','LineWidth',2)
text(0,0,' O','fontsize',14)
line([0,sa],[0,0],'Color','k')
line([0,0],[0,sa],'Color','k')
plot(xCqn,yCqn,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
text(xCqn,yCqn,'  C','FontSize',18)
title('quarter-circle')
xlabel('x(m)')
ylabel('y(m)')

% end of program
```

### 3.9.3 Program 3.3

```
% example 3.3
% quarter-astroid (one cusp)
% astroid=hypocycloid with 4 cusps
% x^(2/3)+y^(2/3)=a^(2/3)
clear all; clc; close all;
syms t real
a = 1; % (m)
x = a*cos(t)^3;
y = a*sin(t)^3;
% a > 0
% 0 < t < pi/2 - quarter-astroid
dx = diff(x,t);
dy = diff(y,t);
% dl=((dx/dt)^2+(dy/dt)^2)^0.5 dt
dl = (dx^2+dy^2)^0.5;
dl = simplify(dl);
```

```
L = int(dl,t,0,pi/2);
L = double(L);
% L = int(dl,0,pi/2);
My = int(x*dl,t,0,pi/2);
xC = My/L;
xC = double(xC);
Mx = int(y*dl,t,0,pi/2);
yC = Mx/L;
yC = double(yC);
fprintf('L = %4.3f (m)\n', L)
fprintf('xC = %4.3f (m)\n', xC)
fprintf('yC = %4.3f (m)\n', yC)
fprintf('\n');

% plot and CM
tn = 0:pi/18:pi/2;
xn = a*cos(tn).^3;
yn = a*sin(tn).^3;
axis manual
axis equal
hold on
grid on
sa = 1;
axis ([0 sa 0 sa])
plot(xn,yn,'-b','LineWidth',2)
text(0,0,' O','fontsize',14)
line([0,sa],[0,0],'Color','k')
line([0,0],[0,sa],'Color','k')
plot(xC,yC,'o','MarkerSize',12,...
               'MarkerEdgeColor','k',...
               'MarkerFaceColor','r')
text(xC,yC,'   C','FontSize',18)
title('quarter-astroid')
xlabel('x(m)')
ylabel('y(m)')

% end of program
```

### 3.9.4 Program 3.4

```
% example 3.4
% three-dimensional helix
clear all; clc; close all
```

```
syms a h t real
% circular helix
% radius a and pitch 2*pi*h
% cartesian coordinates
x = a*cos(t);
y = a*sin(t);
z = h*t;

dx = diff(x,t);
dy = diff(y,t);
dz = diff(z,t);
%dl=((dx/dt)^2+(dy/dt)^2+(dz/dt)^2)^0.5 dt
dl = (dx^2+dy^2+dz^2)^0.5;
dl = simplify(dl);
fprintf('dl = %s dt \n', char(dl));
fprintf('\n');
% dl =(a^2 + h^2)^(1/2)

% numerical data a=h=1 (m)
x = cos(t);
y = sin(t);
z = t;
dl=(diff(x)^2+diff(y)^2+diff(z)^2)^0.5;

tf = 6*pi;
L = int(dl,t,0,tf);
L = double(L);
xC = int(x*dl,t,0,tf)/L;
xC = double(xC);
yC = int(y*dl,t,0,tf)/L;
yC = double(yC);
zC = int(z*dl,t,0,tf)/L;
zC = double(zC);
fprintf('L = %4.3f (m)\n', L)
fprintf('xC = %4.3f (m)\n', xC)
fprintf('yC = %4.3f (m)\n', yC)
fprintf('zC = %4.3f (m)\n', zC)
fprintf('\n');

% plot helix
tn = 0:pi/50:tf;
plot3(sin(tn),cos(tn),tn)
hold on
plot3(xC,yC,zC,...
'o','MarkerSize',12,...
```

```
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
text(xC,yC,zC,'   C','FontSize',18)
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
grid on
axis square

% end of program
```

### *3.9.5 Program 3.5*

```
% example 3.5
clear all; clc; close all;
syms t real
a=2; k=1;
x = a*exp(k*t)*cos(t);
y = a*exp(k*t)*sin(t);
z = a*exp(k*t);

dx = diff(x,t);
dy = diff(y,t);
dz = diff(z,t);
%dl=((dx/dt)^2+(dy/dt)^2+(dz/dt)^2)^0.5 dt
dl = (dx^2+dy^2+dz^2)^0.5;
dl = simplify(dl);
fprintf('dl = %s dt\n',char(dl))

tf = 3;
L = int(dl,0,tf);
L = double(L);

xC = int(x*dl,t,0,tf)/L;
xC = double(xC);
yC = int(y*dl,t,0,tf)/L;
yC = double(yC);
zC = int(z*dl,t,0,tf)/L;
zC = double(zC);
fprintf('L = %4.3f (m)\n', L)
fprintf('xC = %4.3f (m)\n', xC)
fprintf('yC = %4.3f (m)\n', yC)
fprintf('zC = %4.3f (m)\n', zC)
```

```
fprintf('\n');

% plot the curve
tn = 0:pi/100:tf;
xn = a*exp(k*tn).*cos(tn);
yn = a*exp(k*tn).*sin(tn);
zn = a*exp(k*tn);
ht = plot3(xn,yn,zn);
hold on
plot3(xC,yC,zC,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
text(xC,yC,zC,'   C','FontSize',18)
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
grid on
axis square

% end of program
```

### 3.9.6 Program 3.6

```
% example 3.6
clear all; clc; close all
syms x y a b real

% f(x) = y(x) = b*x/a;
f = b*x/a;
% 0 < x < xf
xf = a;

% Area = int(dx dy) where
% 0<x<xf and 0<y<f
% Ay = int(dy) where 0<y<f
Ay = int(1,y,0,f);
% Area = int(Ay dx) where 0<x<xf
Area = int(Ay,x,0,xf);

% first moment of area about y-axis
% My = int(x dx dy) where
% 0<x<xf and 0<y<f
```

```
% Qyy = int(dy) ; 0<y<f
Qyy = int(1,y,0,f);
% My = int(x Qyy dx) where 0<x<xf
My = int(x*Qyy,x,0,xf);
% centroid xC = My/Area
xC = My/Area;

% first moment of area about x-axis
% Mx = int(y dx dy) where
% 0<x<xf and 0<y<f
% Qxy = int(y dy) ; 0<y<f
Qxy = int(y,y,0,f);
% Mx = int(x Qxy dx) where 0<x<xf
Mx = int(Qxy,x,0,xf);
yC = Mx/Area;

ls = {a,b};
ln = {1,1}; % (m)

xfn = subs(xf,ls,ln);
xCn = subs(xC,ls,ln);
yCn = subs(yC,ls,ln);

fprintf('xf = %s = %4.3f (m)\n',char(xf),double(xfn))
fprintf('Area = %s (m^2)\n',char(Area))
fprintf('My = %s (m^3)\n',char(My))
fprintf('xC = %s = %4.3f (m)\n',char(xC),double(xCn))
fprintf('Mx = %s (m^3)\n',char(Mx))
fprintf('yC = %s = %4.3f (m)\n',char(yC),double(yCn))

sa = 1.5;
axis([0 sa 0 sa])
hold on, grid on
xx = 0:.1:xfn;
fx = subs(f,{a,b,x},{1,1,xx});
plot(xx,fx,'--','LineWidth',2)
hold on
area(xx,fx,'FaceColor','b',...
           'EdgeColor','k',...
           'LineWidth',2)
hold on
plot(xCn,yCn,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
```

```
xlabel('x (m)'), ylabel('y (m)')
text(xCn,yCn,'   C','fontsize',14,'fontweight','b')

% end of program
```

### 3.9.7 Program 3.7

```
% example 3.7
clear all; clc; close all
syms x y A k real

% f(x) = y(x) = A*sin(k*x);
f = A*sin(k*x);
% 0 < x < xf
xf = (3*pi/4)/k;

% Area = int(dx dy) where
% 0<x<xf and 0<y<f
% Ay = int(dy) where 0<y<f
Ay = int(1,y,0,f);
% Ay = A*sin(k*x)
% Area = int(Ay dx) where 0<x<xf
Area = int(Ay,x,0,xf);

% first moment of area about y-axis
% My = int(x dx dy) where
% 0<x<xf and 0<y<f
% Qyy = int(dy) ; 0<y<f
Qyy = int(1,y,0,f);
% My = int(x Qyy dx) where 0<x<xf
My = int(x*Qyy,x,0,xf);
xC = My/Area;

% first moment of area about x-axis
% Mx = int(y dx dy) where
% 0<x<xf and 0<y<f
% Qxy = int(y dy) ; 0<y<f
Qxy = int(y,y,0,f);
% Mx = int(x Qxy dx) where 0<x<xf
Mx = int(Qxy,x,0,xf);
yC = Mx/Area;
```

```
% A = 1.5;  % m
% k = 0.75; % m^(-1)
xfn = subs(xf,{A,k},{1.5,0.75});
xCn = subs(xC,{A,k},{1.5,0.75});
yCn = subs(yC,{A,k},{1.5,0.75});
Arean = subs(Area,{A,k},{1.5,0.75});

fprintf('xf = %s = %4.3f (m)\n',char(xf),double(xfn))
fprintf('\n')
fprintf('Area = ')
pretty(Area)
fprintf('\n')
fprintf('My = ')
pretty(My)
fprintf('\n')
fprintf('xC = ')
pretty(xC)
fprintf('\n')
fprintf('xC = %4.3f (m)\n',double(xCn))
fprintf('\n')
fprintf('Mx = ')
pretty(Mx)
fprintf('\n')
fprintf('yC = ')
pretty(yC)
fprintf('\n')
fprintf('yC = %4.3f (m)\n',double(yCn))

A = 1.5;  % m
k = 0.75; % m^(-1)
sa = 4;
axis([0 sa  0 sa])
hold on, grid on
xx = 0:.1:xfn;
fx = A*sin(k*xx);
plot(xx,fx,'--','LineWidth',2)
hold on
area(xx,fx,'FaceColor','b',...
           'EdgeColor','k',...
           'LineWidth',2)
hold on
plot(xCn,yCn,...
'o','MarkerSize',12,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
```

```
xlabel('x (m)'), ylabel('y (m)')
text(xCn,yCn,'   C',...
'fontsize',14,'fontweight','b')

% end of program
```

### 3.9.8 Program 3.8

```
% example 3.8
clear all; clc; close all
syms x y real
% y1 = x/4
% y2 = sqrt(2*(x-3))

y1 = x/4;
y2 = sqrt(2*(x-3));

sol = eval(solve(y2-y1));
if sol(2) > sol(1)
x1 = sol(1); x2 = sol(2);
else
x1 = sol(2); x2 = sol(1);
end

y11 = subs(y1,x,x1);
y12 = subs(y1,x,x2);

axis equal
g1=ezplot(y1,[0,x2+5])
set(g1, 'Color', 'r','LineWidth',3)
hold on
g2=ezplot(y2,[0,x2+5])
set(g2, 'Color', 'b','LineWidth',2)
hold on
line([x1 x1],[0 y11],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')
hold on
line([x2 x2],[0 y12],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')
hold on
grid on
```

```
title('region bounded by y1 and y2')
xlabel('x(m)'), ylabel('y(m)')

% Area = eval(int(abs(y2-y1),x,x1,x2))
A = double(int(int(1,y1,y2),x1,x2));

Qyy = int(1, y, y1, y2);
My = int(x*Qyy, x, x1, x2);
xC = eval(My/A);

Qxy = int(y, y, y1, y2);
Mx = int(Qxy, x, x1, x2);
yC = eval(Mx/A);

plot(xC,yC,...
'o','MarkerSize',6,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
text(xC,yC,'   C',...
'fontsize',14,'fontweight','b')

text(x1,0,'  x_1',...
'fontsize',14,'fontweight','b')
text(x2,0,'  x_2',...
'fontsize',14,'fontweight','b')

fprintf('x1 = %4.3f (m)\n',double(x1))
fprintf('x2 = %4.3f (m)\n',double(x2))
fprintf('A = %4.3f (m^2)\n',double(A))
fprintf('xC = %4.3f (m)\n',double(xC))
fprintf('yC = %4.3f (m)\n',double(yC))

% end of program
```

### *3.9.9 Program 3.9*

```
% example 3.9
clear all; clc; close all
% f = f(x)
% A = int(f,x,a,b)
% xC = int(x*f,x,a,b)/A
% yC = 0.5*int(f^2,x,a,b)/A

syms x a b  real
```

```
sl = {a,b};
sn = {4,2};

y1 = 2*b;
y2 = b;

A1 = int(y1,x,0,a);
A2 = int(y2,x,a,2*a);
A = A1+A2;

Mx1 = int(x*y1,x,0,a);
Mx2 = int(x*y2,x,a,2*a);
Mx = Mx1 + Mx2;

xC1 = Mx1/A1;
xC2 = Mx2/A2;

xC = (Mx1+Mx2)/A;

My1 = 0.5*int(y1^2,x,0,a);
My2 = 0.5*int(y2^2,x,a,2*a);
My = My1 + My2;

yC1 = My1/A1;
yC2 = My2/A2;

yC = (My1+My2)/A;

An = subs(A,sl,sn);
xCn = subs(xC,sl,sn);
yCn = subs(yC,sl,sn);

xC1n = subs(xC1,sl,sn);
yC1n = subs(yC1,sl,sn);

xC2n = subs(xC2,sl,sn);
yC2n = subs(yC2,sl,sn);

fprintf('A = ')
pretty(A)
fprintf('\n')
fprintf('A = %4.3f (m^2)\n',double(An))
fprintf('\n')
fprintf('My = ')
pretty(My)
```

```
fprintf('\n')
fprintf('xC = ')
pretty(xC)
fprintf('\n')
fprintf('xC = %4.3f (m)\n',double(xCn))
fprintf('\n')
fprintf('Mx = ')
pretty(Mx)
fprintf('\n')
fprintf('yC = ')
pretty(yC)
fprintf('\n')
fprintf('yC = %4.3f (m)\n',double(yCn))

an = subs(a,sl,sn);
bn = subs(b,sl,sn);
xA = 2*an; yA = 0;
xB = xA;   yB = bn;
xD = an;   yD = yB;
xE = xD;   yE = 2*bn;
xF = 0;    yF = yE;

% axis square

sa = 8;
axis([0 sa  0 sa])
hold on, grid on

line([0 xA],[0 yA],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')
hold on
line([xA xB],[yA yB],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')
hold on
line([xB xD],[yB yD],...
    'Color','r','LineWidth',2,...
    'Marker','.','LineStyle','-')
hold on
line([xD xE],[yD yE],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')
hold on
line([xE xF],[yE yF],...
```

```
    'Color','b','LineWidth',2,...
    'Marker','.','LineStyle','-')
hold on
line([0 xF],[0 yF],...
    'Color','k','LineWidth',1,...
    'Marker','.','LineStyle','-')

plot(xCn,yCn,...
'o','MarkerSize',8,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','r')
text(xCn,yCn,'   C',...
'fontsize',14,'fontweight','b')

plot(xC1n,yC1n,...
'o','MarkerSize',6,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','b')
text(xC1n,yC1n,'   C1',...
'fontsize',14,'fontweight','b')

plot(xC2n,yC2n,...
'o','MarkerSize',6,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','b')
text(xC2n,yC2n,'   C2',...
'fontsize',14,'fontweight','b')

xlabel('x(m)'), ylabel('y(m)')
text(xA,yA,'   A',...
'fontsize',14,'fontweight','b')
text(xB,yB,'   B',...
'fontsize',14,'fontweight','b')
text(xD,yD,'   D',...
'fontsize',14,'fontweight','b')
text(xE,yE,'   E',...
'fontsize',14,'fontweight','b')
text(xF,yF,'   F',...
'fontsize',14,'fontweight','b')
text(0,0,'   O',...
'fontsize',14,'fontweight','b')

% end of program
```

### 3.9.10 Program 3.10

```
% example 3.10

% frustum of a right-circular cone
% volume of a frustum of a cone
% h height
% R radius of large base
% r radius of small base

clear all; clc; close all
% f = y(x)
% V = pi*int(f^2,x,a,b)

syms R r h x  real

f = (R-r)*x/h+r;

% volume of thin disk differential element
% dV = pi f^2 dx

V = pi*int(f^2,x,0,h);

% centroid
% xC = int(x*pi*f^2,x,0,h)/V;

xC = int(x*pi*f^2,x,0,h)/V;
xC = simplify(xC);

fprintf('V = ')
fprintf('\n')
pretty(V)
fprintf('\n')
fprintf('xC = ')
fprintf('\n')
pretty(xC)
fprintf('\n')

ls = {R,r,h};
ln = {2,1,2};
fn = subs(f,ls,ln);
Vn = subs(V,ls,ln);
xCn= subs(xC,ls,ln);
```

```
fprintf('V = %4.3f (m^3)\n',double(Vn))
fprintf('xC = %4.3f (m^3)\n',double(xCn))

g1 = ezplot(fn,[0,2]);
set(g1,'Color','b','LineWidth',2)
grid on

% end of program
```

### *3.9.11 Program 3.11*

```
% example 3.11
% torus volume
% x^2+(y-b)^2-r^2=, b>0
% torus can be generated by revolving
% the circular area of radius r
% through 360 deg. about the x-axis
clear all; clc; close all
% f = f(x)
% V = pi*int(f^2,x,x1,x2)

syms b r x  real

f1 = b-sqrt(r^2-x^2);
f2 = b+sqrt(r^2-x^2);
V = pi*int(f2^2-f1^2,x,-r,r);
fprintf('V = %s \n',char(V))

% theorems of Guldinus-Pappus
%
% S = 2 pi yC L  surface of revolution
% L length of generating curve
% yC centroid of generating curve
%
% V = 2 pi yC A  volume of revolution
% A area of generating plane surface
% yC centroid of generating plane surface

yC = b;
A = pi*r^2;
Vg = 2*pi*yC*A;
L = 2*pi*r;
S = 2*pi*yC*L;
```

```
fprintf('\n')
fprintf('Vg = %s \n',char(Vg))
fprintf('\n')
fprintf('S = %s \n',char(S))
fprintf('\n')

% end of program
```

## References


1. P. Appell, *Traité de mécanique rationnelle* (Gauthier-Villars, Paris, 1955)
2. M. Atanasiu, *Mechanics* (EDP, Bucharest, 1973)
3. H. Baruh, *Analytical Dynamics* (WCB/McGraw-Hill, Boston, 1999)
4. G. Baumann, *Mathematica for Theoretical Physics: Classical Mechanics and Nonlinear Dynamics* (Springer-Verlag, New York, 2005a)
5. G. Baumann, *Mathematica for Theoretical Physics: Electrodynamics, Quantum Mechanics, General Relativity and Fractals* (Springer-Verlag, NewYork, 2005b)
6. F.P. Beer, E.R. Johnston, *Vector Mechanics for Engineers: Statics and Dynamics 5/e* (McGraw-Hill Publishing Company, New York, 1988)
7. F.P. Beer, E.R. Johnston, D.F. Mazurek, *Vector Mechanics for Engineers: Statics, 10/e* (McGraw-Hill Publishing Company, New York, 2012)
8. A.M. Bedford, W. Fowler, K.M. Liechti, *Statics and Mechanics of Materials* (Prentice Hall, Inc., Upper Saddle River, 2002)
9. A.M. Bedford, W. Fowler, *Engineering Mechanics: Statics, 5/e* (Prentice Hall, Inc., Upper Saddle River, 2007)
10. A.P. Boresi and R.J. Schmidt, *Engineering Mechanics: Statics* (PWS Publishing Company, Boston, 2000)
11. M.I. Buculei, *Mechanics* (University of Craiova Press, Craiova, 1974)
12. M.I. Buculei, D. Bagnaru, G. Nanu, D.B. Marghitu, *Analysis of Mechanisms with Bars* (Scrisul romanesc, Craiova, 1986)
13. I. Stroe et al., Analytical Mechanics Problems, University Politehnica of Bucharest, 1997
14. V. Ceausu, N. Enescu, F. Ceausu, *Mechanics Problems* (Printech, Bucuresti, 1999)
15. S.J. Chapman, *MATLAB Programming for Engineers* (Thomson Learning, Pacific Grove, CA, 2002)
16. S. Attaway, *MATLAB: A Practical Introduction to Programming and Problem Solving* (Butterworth-Heinemann, Elsevier, Amsterdam, 2012)
17. D.M. Etter, D.C. Kuncicky, *Introduction to MATLAB for Engineers and Scientists* (Prentice Hall, Inc., Upper Saddle River, 1996)
18. C. Iacob, *Theoretical Mechanics* (EDP, Bucharest, 1980)
19. J.H. Ginsberg, *Advanced Engineering Dynamics* (Cambridge University Press, Cambridge, 1995)
20. D.T. Greenwood, *Principles of Dynamics* (Prentice-Hall, Inc., Englewood Cliffs, 1998)
21. L.E. Goodman, W.H. Warner, *Statics* (Dover Publications Inc., Mineola, 2001)
22. R.C. Hibbeler, *Engineering Mechanics: Statics and Dynamics 13/e* (Prentice-Hall Inc., Upper Saddle River, 2013)
23. T.R. Kane, *Analytical Elements of Mechanics*, vol. 1 (Academic Press, New York, 1959)
24. T.R. Kane, *Analytical Elements of Mechanics*, vol. 2 (Academic Press, New York, 1961)
25. T.R. Kane, P.W. Likins, D.A. Levinson, *Spacecraft Dynamics* (McGraw-Hill, New York, 1983)
26. T.R. Kane, D.A. Levinson, *Dynamics* (McGraw-Hill, New York, 1985)

27. R. Maeder, *Programming in Mathematica* (Addison-Wesley Publishing Company, Redwood City, 1990)
28. N.H. Madsen, Statics and Dynamics, class notes, http://www.eng.auburn.edu/users/nmadsen/
29. D.B. Marghitu, *Mechanical Engineer's Handbook* (Academic Press, San Diego, 2001)
30. D.B. Marghitu, M.J. Crocker, *Analytical Elements of Mechanisms* (Cambridge University Press, Cambridge, 2001)
31. D.B. Marghitu, *Kinematic Chains and Machine Component Design* (Elsevier, Amsterdam, 2005)
32. D.B. Marghitu, *Mechanisms and Robots Analysis with MATLAB* (Springer, New York, 2009)
33. D.B. Marghitu, M. Dupac, *Advanced Dynamics: Analytical and Numerical Calculations with MATLAB* (Springer, New York, 2012)
34. D.B. Marghitu, Statics and Dynamics, class notes, http://www.eng.auburn.edu/users/marghitu/
35. D.J. McGill, W.W. King, *Engineering Mechanics: Statics and An Introduction to Dynamics* (PWS Publishing Company, Boston, 1995)
36. J.L. Meriam, L.G. Kraige, *Engineering Mechanics: Statics, 7/e* (Wiley, New York, 2011)
37. R.L. Mott, *Machine Elements in Mechanical Design* (Prentice Hall, Upper Saddle River, 1999)
38. R.L. Norton, *Machine Design* (Prentice-Hall, Upper Saddle River, 1996)
39. L.A. Pars, *A Treatise on Analytical Dynamics* (Wiley, New York, 1965)
40. M. Plesha, G. Gray, F. Costanzo, *Engineering Mechanics: Statics, 2/e* (McGraw-Hill Publishing Company, New York, 2012)
41. M. Radoi, E. Deciu, *Mechanics* (EDP, Bucharest, 1981)
42. W.F. Riley, L.D. Sturges, *Engineering Mechanics: Statics, 2/e* (Wiley, New York, 1995)
43. A. Ruina, R. Pratap, *Introduction to Statics and Dynamics* (Oxford University Press, Oxford, 2002)
44. A. Ripianu, P. Popescu, B. Balan, *Technical Mechanics* (EDP, Bucharest, 1979)
45. I.H. Shames, *Engineering Mechanics Statics, 4/e* (Prentice Hall Inc., Upper Saddle River, 1996)
46. S.D. Sheppard, B.H. Tongue, *Statics: Analysis and Design of Systems in Equilibrium* (Wiley, New York, 2005)
47. D. Smith, *Engineering Computation with MATLAB* (Pearson Education, Upper Saddle River, 2008)
48. R.W. Soutas-Little, D.J. Inman, *Engineering Mechanics: Statics and Dynamics* (Prentice-Hall, Upper Saddle River, 1999)
49. R.W. Soutas-Little, D.J. Inman, D. Balint, *Engineering Mechanics: Statics* (Cengage Learning, Independence, KY, 2007)
50. S. Staicu, *Theoretical Mechanics* (EDP, Bucharest, 1998)
51. A. Stan, M. Grumazescu, *Mechanics Problems* (EDP, Bucharest, 1973)
52. J. Sticklen, M.T. Eskil, *An Introduction to Technical Problem Solving with MATLAB* (Great Lakes Press, Wildwood, 2006)
53. A. Stoenescu, G. Silas, *Theoretical Mechanics* (ET, Bucharest, 1957)
54. J.H. Jackson, H.G. Wirtz, *Schaum's Outline of Theory and Problems of Statics and Strength of Materials* (McGraw-Hill Publishing Company, New York, 1983)
55. The MathWorks, https://www.mathworks.com/
56. Statics eBook, https://ecourses.ou.edu/
57. R. Voinea, D. Voiculescu, V. Ceausu, *Mechanics* (EDP, Bucharest, 1983)
58. V. Valcovici, S. Balan, R. Voinea, *Theoretical Mechanics* (ET, Bucharest, 1959)
59. K.J. Waldron, G.L. Kinzel, *Kinematics, Dynamics, and Design of Machinery* (Wiley, New York, 1999)
60. H.B. Wilson, L.H. Turcotte, D. Halpern, *Advanced Mathematics and Mechanics Applications Using MATLAB* (Chapman & Hall/CRC, Boca Raton, 2003)
61. J.H. Williams Jr, *Fundamentals of Applied Dynamics* (Wiley, New York, 1996)
62. S. Wolfram, *Mathematica, Wolfram Media* (Cambridge University Press, Cambridge, 1999)