(*b*) Use MATLAB to solve the equation derived in part (*a*).
(*c*) For the angle determined in part (*b*), use the `ezplot` command to make a plot of the projectile's trajectory.

Solution

(*a*) The motion of the projectile can be analyzed by considering the horizontal and vertical components. The initial velocity v_0 can be resolved into horizontal and vertical components:

$$v_{0x} = v_0\cos(\theta) \quad \text{and} \quad v_{0y} = v_0\sin(\theta)$$

In the horizontal direction the velocity is constant, and the position of the projectile as a function of time is given by:

$$x = v_{0x}t$$

Substituting $x = 2600$ m for the horizontal distance that the projectile travels to reach the target and $210\cos(\theta)$ for v_{0x}, and solving for t gives:

$$t = \frac{2600}{210\cos(\theta)}$$

In the vertical direction the position of the projectile is given by:

$$y = v_{0y}t - \frac{1}{2}gt^2$$

Substituting $y = 350$ m for the vertical coordinate of the target, $210\sin(\theta)$ for v_{0x}, $g = 9.81$, and t gives:

$$350 = 210\sin\theta\frac{2600}{210\cos\theta} - \frac{1}{2}9.81\left(\frac{2600}{210\cos\theta}\right)^2$$

or:

$$350 = \frac{2600\sqrt{1-\cos^2\theta}}{\cos\theta} - \frac{1}{2}9.81\left(\frac{2600}{210\cos\theta}\right)^2$$

The solution of this equation gives the angle θ at which the projectile has to be fired.

(*b*) A solution of the equation derived in part (*a*) obtained by using the `solve` command (in the Command Window) is:

```
>> syms th
Angle = solve('2600*sqrt(1 - cos(th)^2)/cos(th) -
0.5*9.81*(2600/(210*cos(th)))^2 = 350')
Angle =
    1.2453544972374161683138135806560
  0.45925280703207121277786452037279
 -0.45925280703207121277786452037279
   -1.2453544972374161683138135806560
```

MATLAB displays four solutions. The two positive ones are relevant to the problem.

```
>> Angle1 = Angle(1)*180/pi
```
Converting the solution in the first element of `Angle` from radians to degrees.

```
Angle1 =
224.16380950273491029648644451808/
```
MATLAB displays the answer as a symbolic object in terms of π.

```
>> Angle1=double(Angle1)
Angle1 =
   71.3536
```
Use the `double` command to obtain numerical values for `Angle1`.

```
>> Angle2=Angle(2)*180/pi
```
Converting the solution in the second element of `Angle` from radians to degrees.

```
Angle2 =
82.665505265772818300015613667102/pi
```
MATLAB displays the answer as a symbolic object in terms of π.

```
>> Angle2=double(Angle2)
Angle2 =
   26.3132
```
Use the `double` command to obtain numerical values for `Angle2`.

(*c*) The solution from part (*b*) shows that there are two possible angles and thus two trajectories. In order to make a plot of a trajectory, the x and y coordinates of the projectile are written in terms of t (parametric form):

$$x = v_0\cos(\theta)t \quad \text{and} \quad y = v_0 \sin\theta\, t - \frac{1}{2}gt^2$$

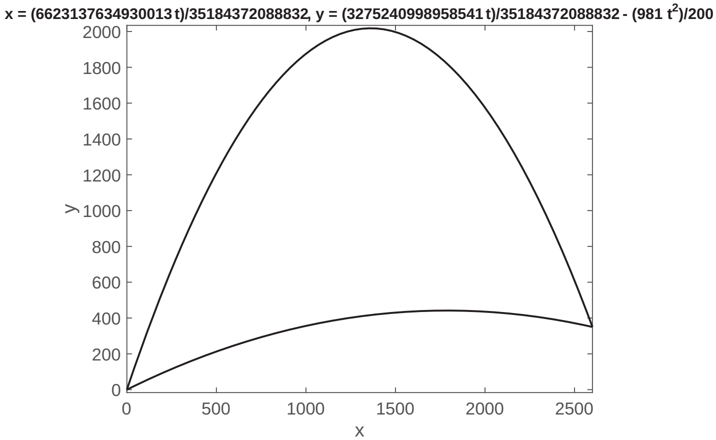
The domain for t is $t = 0$ to $t = \dfrac{2600}{210\cos(\theta)}$.

These equations can be used in the `ezplot` command to make the plots shown in the following program written in a script file.

```
xmax=2600; v0=210; g=9.81;
theta1=1.24535; theta2=.45925;
t1=xmax/(v0*cos(theta1));
t2=xmax/(v0*cos(theta2));
syms t
X1=v0*cos(theta1)*t;
X2=v0*cos(theta2)*t;
Y1=v0*sin(theta1)*t-0.5*g*t^2;
Y2=v0*sin(theta2)*t-0.5*g*t^2;
ezplot(X1,Y1,[0,t1])
hold on
ezplot(X2,Y2,[0,t2])
hold off
```
Assign the two solutions from part (*b*) to `theta1` and `theta2`.

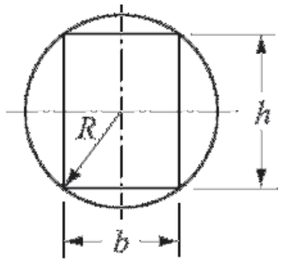
Plot one trajectory. (for `ezplot(X1,Y1,[0,t1])`)

Plot a second trajectory. (for `ezplot(X2,Y2,[0,t2])`) When this program is executed, the following plot is generated in the Figure Window:



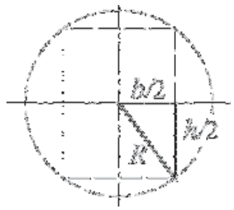
Sample Problem 11-3: Bending resistance of a beam

The bending resistance of a rectangular beam of width b and height h is proportional to the beam's moment of inertia I, defined by $I = \frac{1}{12}bh^3$. A rectangular beam is cut out of a cylindrical log of radius R. Determine b and h (as a function of R) such that the beam will have maximum I.



Solution

The problem is solved by following these steps:
1. Write an equation that relates R, h, and b.
2. Derive an expression for I in terms of h.
3. Take the derivative of I with respect to h.
4. Set the derivative equal to zero and solve for h.
5. Determine the corresponding b.



The first step is carried out by looking at the triangle in the figure. The relationship between R, h, and b is given by the Pythagorean theorem as $\left(\frac{b}{2}\right)^2 + \left(\frac{h}{2}\right)^2 = R^2$. Solving this equation for b gives $b = \sqrt{4R^2 - h^2}$.

The rest of the steps are done using MATLAB:

```
>> syms b h R
>> b=sqrt(4*R^2-h^2);                Create a symbolic expression for b.
>> I=b*h^3/12                        Step 2: Create a symbolic expression for I.
I =
(h^3*(4*R^2-h^2)^(1/2))/12           MATLAB substitutes b in I.
```

```
>> ID=diff(I,h)

ID =
(h^2*(4*R^2-h^2)^(1/2))/4-h^4/(12*(4*R^2-h^2)^(1/2))
```

Step 3: Use the `diff(R)` command to differentiate `I` with respect to *h*.

The derivative of *I* is displayed.

```
>> hs=solve(ID,h)

hs =
          0
  3^(1/2)*R
 -3^(1/2)*R
```

Step 4: Use the `solve` command to solve the equation $ID = 0$ for *h*. Assign the answer to *hs*.

MATLAB displays three solutions. The positive non zero solution $\sqrt{3}R$ is relevant to the problem.

```
>> bs=subs(b,hs(2))
```

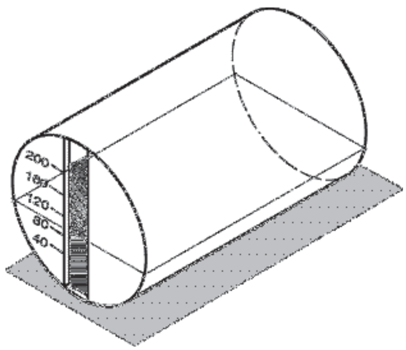
Step 5: Use the `subs` command to determine *b* by substituting the solution for *h* in the expression for

```
bs =
(R^2)^(1/2)
```

The answer for *b* is displayed. (The answer is *R*, but MATLAB displays $(R^2)^{1/2}$.)

Sample Problem 11-4: Fuel level in a tank

The horizontal cylindrical tank shown is used to store fuel. The tank has a diameter of 6 m and is 8 m long. The amount of fuel in the tank can be estimated by looking at the level of the fuel through a narrow vertical glass window at the front of the tank. A scale that is marked next to the window shows the levels of the fuel corresponding to 40, 60, 80, 120, and 160 thousand liters. Determine the vertical positions (measured from the ground) of the lines of the scale.

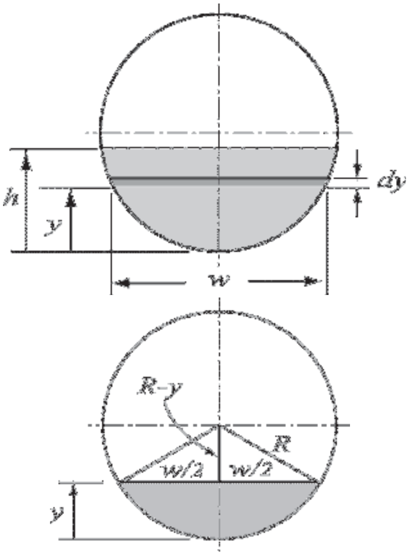


Solution

The relationship between the level of the fuel and its volume can be written in the form of a definite integral. Once the integration is carried out, an equation is obtained for the volume in terms of the fuel's height. The height corresponding to a specific volume can then be determined from solving the equation for the height.

The volume of the fuel V can be determined by multiplying the area of the cross section of the fuel A (the shaded area) by the length of the tank L. The cross-sectional area can be calculated by integration.

$$V = AL = L\int_0^h w\,dy$$



The width w of the top surface of the fuel can be written as a function of y. From the triangle in the figure on the right, the variables y, w, and R are related by:

$$\left(\frac{w}{2}\right)^2 + (R-y)^2 = R^2$$

Solving this equation for w gives:

$$w = 2\sqrt{R^2 - (R-y)^2}$$

The volume of the fuel at height h can now be calculated by substituting w in the integral in the equation for the volume and carrying out the integration. The result is an equation that gives the volume V as a function of h. The value of h for a given V is obtained by solving the equation for h. In the present problem values of h have to be determined for volumes of 40, 60, 80, 120, and 160 thousand liters. The solution is given in the following MATLAB program (script file):

`R=3; L=8;`	
`syms w y h`	
`w=2*sqrt(R^2-(R-y)^2)`	Create a symbolic expression for w.
`S = L*w`	Create the expression that will be integrated.
`V = int(S,y,0,h)`	Use the `int` command to integrate `S` from 0 to h. The result gives V as a function of h.
`Vscale=[40:40:200]`	Create a vector with the values of V in the scale.
`for i=1:5`	Each pass in the loop solves h for one value of V.
`    Veq=V-Vscale(i);`	Create the equation for h that has to be solved.
`    h_ans(i)=solve(Veq);`	Use the `solve` command to solve for h.
`end`	`h_ans` is a vector (symbolic with numbers) with the values of h that correspond to the values of V in the vector `Vscale`.
`h_scale=double(h_ans)`	Use the `double` command to obtain numerical values for the elements of vector `h_ans`.

When the script file is executed, the outcomes from commands that don't have a semicolon at the end are displayed. The display in the Command Window is:

```
>> w =
2*(9-(y-3)^2)^(1/2)
S =
16*(9-(y-3)^2)^(1/2)
V =
36*pi+72*asin(h/3-1)+8*(9-(h-3)^2)^(1/2)*(h-3)

Vscale =
  40   80   120   160   200
h_scale =
  1.3972   2.3042   3.1439   3.9957   4.9608
```

The symbolic expression for w is displayed.

S is the expression that will be integrated.

The result from the integration; V as a function of h.

The values of V in the scale are displayed.

The positions of the lines in the scale are displayed.

Units: The unit for length in the solution is meters, which correspond to m3 for the volume (1 m3 = 1,000 L).

Sample Problem 11-5: Amount of medication in the body

The amount M of medication present in the body depends on the rate at which the medication is consumed by the body and on the rate at which the medication enters the body, where the rate at which the medication is consumed is proportional to the amount present in the body. A differential equation for M is

$$\frac{dM}{dt} = -km + p$$

where k is the proportionality constant and p is the rate at which the medication is injected into the body.

(*a*) Determine k if the half-life of the medication is 3 hours.

(*b*) A patient is admitted to a hospital and the medication is given at a rate of 50 mg per hour. (Initially there is no medication in the patient's body.) Derive an expression for M as a function of time.

(*c*) Plot M as a function of time for the first 24 hours.

Solution

(*a*) The proportionality constant can be determined from considering the case in which the medication is consumed by the body and no new medication is given. In this case the differential equation is:

$$\frac{dM}{dt} = -kM$$

The equation can be solved with the initial condition $M = M_0$ at $t = 0$:

```
>> syms M M0 k t
```

```
>> Mt=dsolve('DM=-k*M','M(0)=M0')
Mt =
M0/exp(k*t)
```

Use the `dsolve` command to solve $\frac{dM}{dt} = -kM$.

The solution gives M as a function of time:

$$M(t) = \frac{M_0}{e^{kt}}$$

A half-life of 3 hours means that at $t = 3$ hours $M(t) = \frac{1}{2}M_0$. Substituting this information in the solution gives $0.5 = \frac{1}{e^{3k}}$, and the constant k is determined from solving this equation: $0.5 = e^{-3k}$

```
ks=solve('0.5=1/exp(k*3)')
ks =
.231049060186648436472410707l5273
```

Use the `solve` command to solve $0.5 = e^{-3k}$.

(*b*) For this part the differential equation for M is:

$$\frac{dM}{dt} = -kM + p$$

The constant k is known from part (*a*), and $p = 50$ mg/h is given. The initial condition is that in the beginning there is no medication in the patient's body, or $M = 0$ at $t = 0$. The solution of this equation with MATLAB is:

```
>> syms p
>> Mtb=dsolve('DM=-k*M+p','M(0)=0')
Mtb =
(p-p/exp(k*t))/k
```

Use the `dsolve` command to solve $\frac{dM}{dt} = -kM + p$.

(*c*) A plot of `Mtb` as a function of time for $0 \le t \le 24$ can be done by using the `ezplot` command:

```
>> pgiven=50;
>> Mtt=subs(Mtb,{p,k},{pgiven,ks})
Mtt =
216.404-216.404/exp(0.231049*t)
>> ezplot(Mtt,[0,24])
```

Substitute numerical values for p and k.

In the actual display of the last expression that was generated by MATLAB (`Mtt = . . .`) the numbers have many more decimal digits than shown above. The numbers were shortened so that they will fit on the page.
The plot that is generated is: