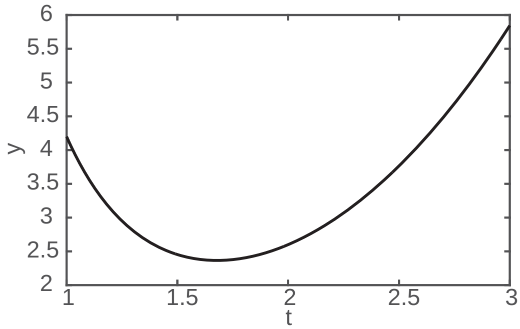
and the solution is plotted with the `plot` command.

```
>> [t y]=ode45(@ODEexp1,[1:0.01:3],4.2);
>> plot(t,y)
>> xlabel('t'), ylabel('y')
```



If the ODE function is written as an anonymous function called `ode1` (see Step 2), then the solution (same as shown above) is obtained by typing:
`[t y]=ode45(ode1,[1:0.5:3],4.2)`

9.5 *Examples of MATLAB Applications*

Sample Problem 9-3: The gas equation

The ideal gas equation relates the volume (V in L), temperature (T in K), pressure (P in atm), and the amount of gas (number of moles n) by:

$$p = \frac{nRT}{V}$$

where R = 0.08206 (L atm)/(mol K) is the gas constant.

The van der Waals equation gives the relationship between these quantities for a real gas by

$$\left(P + \frac{n^2 a}{V^2}\right)(V - nb) = nRT$$

where a and b are constants that are specific for each gas.

Use the `fzero` function to calculate the volume of 2 mol CO_2 at temperature of 50°C, and pressure of 6 atm. For CO_2, a = 3.59 (L^2 atm)/mol^2, and b = 0.0427 L/mol.

Solution

The solution written in a script file is shown below.

```
global P T n a b R
```

```
R=0.08206;
P=6; T=323.2; n=2; a=3.59; b=0.047;
Vest=n*R*T/P;                    Calculating an estimated value for V.
V=fzero(@Waals,Vest)             Function handle @waals is used to pass the
                                 user-defined function waals into fzero.
```

The program first calculates an estimated value of the volume using the ideal gas equation. This value is then used in the `fzero` command for the estimate of the solution. The van der Waals equation is written as a user-defined function named `Waals`, which is shown below:

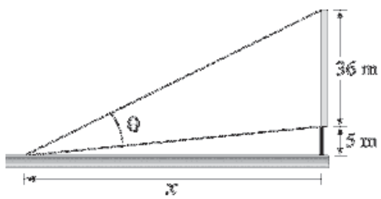
```
function fofx=Waals(x)
global P T n a b R
fofx=(P+n^2*a/x^2)*(x-n*b)-n*R*T;
```

In order for the script and function files to work correctly, the variables `P`, `T`, `n`, `a`, `b`, and `R` are declared global. When the script file (saved as Chap9SamPro3) is executed in the Command Window, the value of V is displayed, as shown next:

```
>> Chap9SamPro3
V =
    8.6613                       The volume of the gas is 8.6613 L.
```

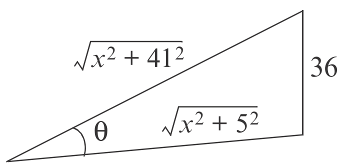
Sample Problem 9-4: Maximum viewing angle

To get the best view of a movie, a person has to sit at a distance x from the screen such that the viewing angle θ is maximum. Determine the distance x for which θ is maximum for the configuration shown in the figure.



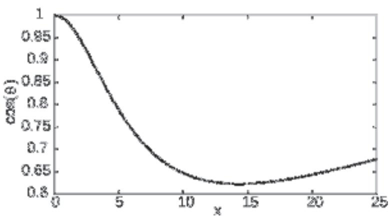
Solution

The problem is solved by writing a function for the angle θ in terms of x, and then finding the x for which the angle is maximum. In the triangle that includes θ, one side is given (the height of the screen), and the other two sides can be written in terms of x, as shown in the figure. One way in which θ can be written in terms of x is by using the Law of Cosines:



$$\cos\theta = \frac{(x^2+5^2)+(x^2+41^2)-36^2}{2\sqrt{x^2+5^2}\sqrt{x^2+41^2}}$$

The angle θ is expected to be between 0 and π/2. Since cos(0) = 1 and the cosine is decreasing with increasing θ, the maximum angle corresponds to the smallest cos(θ). A plot of cos(θ) as a function of x shows that the function has a minimum between 10 and 20. The commands for the plot are:




```
>>fplot('((x^2+5^2)+(x^2+41^2)-36^2)/(2*sqrt(x^2+ 5^2)*sqrt(x^2+
                                                     41^2))',[0 25])
>> xlabel('x'); ylabel('cos(\theta)')
```

The minimum can be determined with the `fminbnd` command:

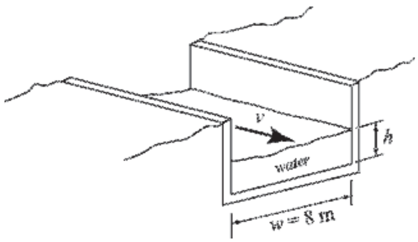
```
>>[x anglecos]=fminbnd('((x^2+5^2)+(x^2+41^2)-36^2)/
                    (2*sqrt(x^2+5^2)*sqrt(x^2+41^2))',10,20)
x =
   14.3178
anglecos =
    0.6225
>> angle=anglecos*180/pi
angle =
   35.6674
```

The minimum is at x = 14.3178 m. At this point cos(θ) = 0.6225.

In degrees the angle is 35.6674°.

Sample Problem 9-5: Water flow in a river

To estimate the amount of water that flows in a river during a year, a section of the river is made to have a rectangular cross section as shown. In the beginning of every month (starting at January 1st) the height h of the water and the speed v of the water flow are measured. The first day of measurement is taken as 1, and the last day—which is January 1st of the next year—is day 366. The following data was measured:



Day	1	32	60	91	121	152	182	213	244	274	305	335	366
h (m)	2.0	2.1	2.3	2.4	3.0	2.9	2.7	2.6	2.5	2.3	2.2	2.1	2.0
v (m/s)	2.0	2.2	2.5	2.7	5	4.7	4.1	3.8	3.7	2.8	2.5	2.3	2.0

Use the data to calculate the flow rate, and then integrate the flow rate to obtain an estimate of the total amount of water that flows in the river during a year.

Solution

The flow rate, Q (volume of water per second), at each data point is obtained by multiplying the water speed by the width and height of the cross-sectional area of the water that flows in the channel:

$$Q = vwh \quad (\mathrm{m^3/s})$$

The total amount of water that flows is estimated by the integral:

$$V = (60 \cdot 60 \cdot 24)\int_{t_1}^{t_2} Q\,dt$$

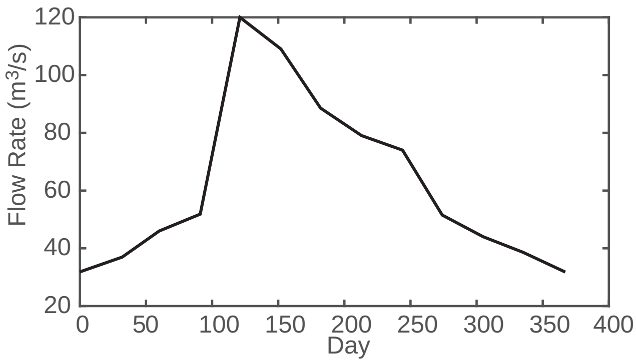
The flow rate is given in cubic meters per second, which means that time must have units of seconds. Since the data is given in terms of days, the integral is multiplied by $(60 \cdot 60 \cdot 24)$ s/day.

The following is a program written in a script file that first calculates Q and then carries out the integration using the `trapz` command. The program also generates a plot of the flow rate versus time.

```
w=8;
d=[1 32 60 91 121 152 182 213 244 274 305 335 366];
h=[2 2.1 2.3 2.4 3.0 2.9 2.7 2.6 2.5 2.3 2.2 2.1 2.0];
speed=[2 2.2 2.5 2.7 5 4.7 4.1 3.8 3.7 2.8 2.5 2.3 2];
Q=speed.*w.*h;
Vol=60*60*24*trapz(d,Q);
fprintf('The estimated amount of water that flows in the
river in a year is %g cubic meters.',Vol)
plot(d,Q)
xlabel('Day'), ylabel('Flow Rate (m^3/s)')
```

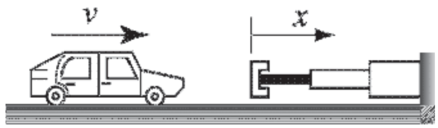
When the file (saved as Chap9SamPro5) is executed in the Command Window, the estimated amount of water is displayed and the plot is generated. Both are shown below:.

```
>> Chap9SamPro5
The estimated amount of water that flows in the river in a
year is 2.03095e+009 cubic meters.
```



Sample Problem 9-6: Car crash into a safety bumper

A safety bumper is placed at the end of a racetrack to stop out-of-control cars. The bumper is designed such that the force that the bumper applies to the car is a function of the velocity v and the displacement x of the front edge of the bumper according to the equation:



$$F = Kv^3(x+1)^3$$

where $K = 30$ (s kg)/m^5 is a constant.

A car with a mass m of 1,500 kg hits the bumper at a speed of 90 km/h. Determine and plot the velocity of the car as a function of its position for $0 \le x \le 3$ m.

Solution

The deceleration of the car once it hits the bumper can be calculated from Newton's second law of motion,

$$ma = -Kv^3(x+1)^3$$

which can be solved for the acceleration a as a function of v and x:

$$a = \frac{-Kv^3(x+1)^3}{m}$$

The velocity as a function of x can be calculated by substituting the acceleration in the equation

$$vdv = adx$$

which gives:

$$\frac{dv}{dx} = \frac{-Kv^3(x+1)^3}{m}$$

The last equation is a first-order ODE that needs to be solved for the interval $0 \le x \le 3$ with the initial condition $v = 90$ km/h at $x = 0$.

A numerical solution of the differential equation with MATLAB is shown in the following program, which is written in a script file:

```
global k m
k=30; m=1500; v0=90;
xspan=[0:0.2:3];                 A vector that specifies the interval of the solution.
v0mps=v0*1000/3600;              Changing the units of v0 to m/s.
[x v]=ode45(@bumper,xspan,v0mps)                  Solving the ODE.
plot(x,v)
xlabel('x (m)'); ylabel('velocity (m/s)')
```

Note that the function handle `@bumper` is used for passing the user-defined function `bumper` into `ode45`. The listing of the user-defined function with the differential equation, named `bumper`, is:

```
function dvdx=bumper(x,v)
global k m
dvdx=-(k*v^2*(x+1)^3)/m;
```

When the script file executes (saved as Chap9SamPro6) the vectors `x` and `v` are displayed in the Command Window (actually, they are displayed on the screen one after the other, but to save room they are displayed below next to each other).

```
>> Chap9SamPro6
x =          v =
        0      25.0000
   0.2000      22.0420
   0.4000      18.4478
   0.6000      14.7561
   0.8000      11.4302
   1.0000       8.6954
   1.2000       6.5733
   1.4000       4.9793
   1.6000       3.7960
   1.8000       2.9220
   2.0000       2.2737
   2.2000       1.7886
   2.4000       1.4226
   2.6000       1.1435
   2.8000       0.9283
```