Application 7.6B
Comets and Spacecraft

The investigations outlined here are intended as applications of the more sophisticated numerical DE solvers that are "built into" computing systems such as *Maple*, *Mathematica*, and MATLAB (as opposed to the *ad hoc* Runge-Kutta methods of the previous project.) We illustrate these high-precision variable step size solvers by applying them to analyze the elliptical orbit of a satellite — a comet, planet, or spacecraft — around a primary (planet or sun) of mass M. If the attracting primary is located at the origin in *xyz*-space, then the satellite's position functions $x(t)$, $y(t)$, and $z(t)$ satisfy Newton's inverse-square law differential equations

$$\frac{d^2x}{dt^2} = -\frac{\mu x}{r^3}, \qquad \frac{d^2y}{dt^2} = -\frac{\mu y}{r^3}, \qquad \frac{d^2z}{dt^2} = -\frac{\mu z}{r^3} \tag{1}$$

where $\mu = GM$ (G being the gravitational constant) and $r = \sqrt{x^2 + y^2 + z^2}$.

Investigation A

Consider a satellite in elliptical orbit around a planet, and suppose that physical units are so chosen that $\mu = 1$. If the orbit lies in the *xy*-plane so $z(t) \equiv 0$, the Eqs. (1) reduce to

$$\frac{d^2x}{dt^2} = -\frac{x}{r^3}, \qquad \frac{d^2y}{dt^2} = -\frac{y}{r^3}. \tag{2}$$

Let T denote the period of revolution of the satellite in its orbit. Kepler's third law says that the *square* of T is proportional to the *cube* of the major semiaxis a of its elliptical orbit. In particular, if $\mu = 1$, then

$$T^2 = 4\pi^2 a^3. \tag{3}$$

(See Section 12.6 of Edwards and Penney, *Calculus*, 6th ed. (Prentice Hall, 2002).) If the satellite's *x*- and *y*-components of velocity, $x_2 = x' = x_1'$ and $y_2 = y' = y_1'$, are introduced, then the system in (2) translates into the system

$$\begin{aligned} x_1' &= x_2, & x_2' &= -\frac{\mu x_1}{r^3} \\ y_1' &= y_2, & y_2' &= -\frac{\mu y_1}{r^3} \end{aligned} \tag{4}$$

of four first-order equations with $r = \sqrt{x_1^2 + y_1^2}$.

(i) Solve Eqs. (2) or (4) numerically with the initial conditions

$$x(0) = 1, \qquad y(0) = 0, \qquad x'(0) = 0, \qquad y'(0) = 1$$

that correspond theoretically to a circular orbit of radius $a = 1$, in which case Eq. (3) gives $T = 2\pi$. Are your numerical results consistent with this fact?

(ii) Now solve the system numerically with the initial conditions

$$x(0) = 1, \qquad y(0) = 0, \qquad x'(0) = 0, \qquad y'(0) = \tfrac{1}{2}\sqrt{6}$$

that correspond theoretically to an elliptical orbit with major semiaxis $a = 2$, so Eq. (3) gives $T = 4\pi\sqrt{2}$. Do your numerical results agree with this?

(iii) Investigate what happens when both the x-component and the y-component of the initial velocity are nonzero.

Investigation B (Halley's Comet)
Halley's comet last reached perihelion (its point of closest approach to the sun at the origin) on February 9, 1986. Its position and velocity components at this time were

$$\mathbf{p_0} = (0.325514, -0.459460, 0.166229) \qquad \text{and}$$

$$\mathbf{v_0} = (-9.096111, -6.916686, -1.305721)$$

(respectively) with position in AU (Astronomical Units, the unit of distance being equal to the major semiaxis of the earth's orbit about the sun) and time in years. In this unit system, its 3-dimensional equations of motion are as in (1) with $\mu = 4\pi^2$. Then solve Eqs. (1) numerically to verify the appearance of the yz-projection of the orbit of Halley's comet shown in Fig. 7.6.13 in the text. Plot the xy- and xz-projections also.

Figure 7.6.14 in the text shows the graph of the distance $r(t)$ of Halley's comet from the sun. Inspection of this graph indicates that Halley's comet reaches a maximum distance (at aphelion) of about 35 AU in a bit less than 40 years, and returns to perihelion after about three-quarters of a century. The closer look in Fig. 4.3.15 indicates that the period of revolution of Halley's comet is about 76 years. Use your numerical solution to refine these observations. What is your best estimate of the calendar date of the comet's next perihelion passage?

Investigation C (Your Own Comet)
Lucky you! The night before your birthday in 1997 you set up your telescope on nearby mountaintop. It was a clear night, and at 12:30 am you spotted a new comet. After repeating the observation on successive nights, you were able to calculate its solar system coordinates $\mathbf{p_0} = (x_0, y_0, z_0)$ and its velocity vector $\mathbf{v_0} = (vx_0, vy_0, vz_0)$ on that first night. Using this information, determine this comet's

- perihelion (point nearest sun) and aphelion (farthest from sun),
- its velocity at perihelion and at aphelion,
- its period of revolution about the sun, and
- its next two dates of perihelion passage.

Using length-time units of AU and earth years, the comet's equations of motion are given in (1) with $\mu = 4\pi^2$. For your personal comet, start with random initial position and velocity vectors with the same order of magnitude as those of Halley's comet. Repeat the random selection of initial position and velocity vectors, if necessary, until you get a nice-looking eccentric orbit that goes well outside the earth's orbit (like real comets do).

Using *Maple*

Let's consider a comet orbiting the sun with initial position and velocity vectors

```
p0 := {x(0)=0.2, y(0)=0.4, z(0)=0.2};
v0 := {D(x)(0)=5, D(y)(0)=-7, D(z)(0)=9};
```

at perihelion. For convenience, we combine these initial conditions in the single set

```
inits := p0 union v0;
```

of equations. The comet's equations of motion in (1) with $\mu = 4\pi^2$ are entered as

```
r :=  t->sqrt(x(t)^2 + y(t)^2 + z(t)^2);

de1 := diff(x(t),t$2) = -4*Pi^2*x(t)/r(t)^3:
de2 := diff(y(t),t$2) = -4*Pi^2*y(t)/r(t)^3:
de3 := diff(z(t),t$2) = -4*Pi^2*z(t)/r(t)^3:

deqs := {de1,de2,de3}:
```

The comet's *x*-, *y*-, and *z*-position functions then satisfy the combined set

```
eqs := deqs union inits:
```

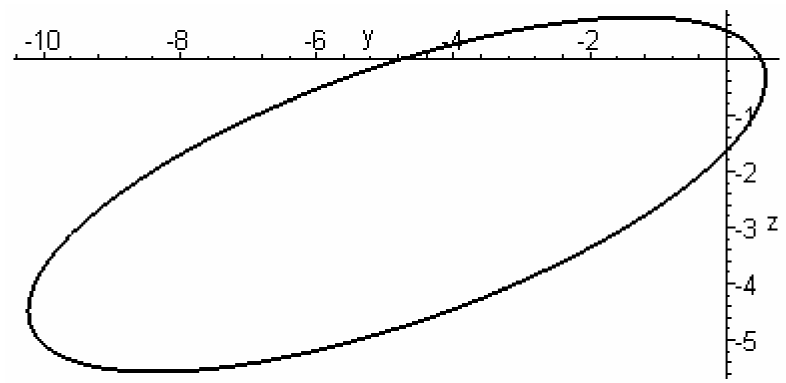
of three second-order differential equations and six initial conditions, which we proceed to solve numerically.

```
soln := dsolve(eqs, {x(t),y(t),z(t)}, type=numeric);
```

soln := **proc**(*rkf45_x*) ... **end**

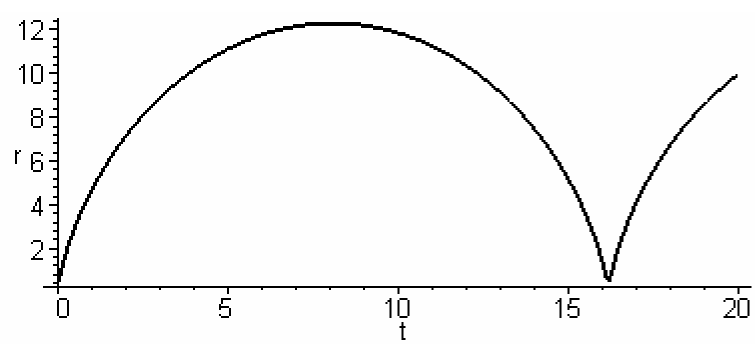
We use the resulting numerical procedure **soln** to plot the *yz*-projection of the comet's orbit for the first 20 years:

```
with(plots):
odeplot(soln, [y(t),z(t)], 0..20, numpoints=1000);
```



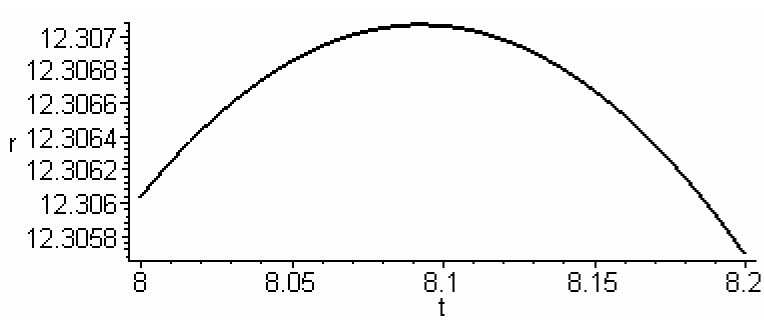
This orbit certainly looks like an ellipse. To investigate the comet's motion on its orbit, we plot its distance r from the sun as a function of t.

```
odeplot(soln, [t,r(t)], 0..20, numpoints=1000);
```



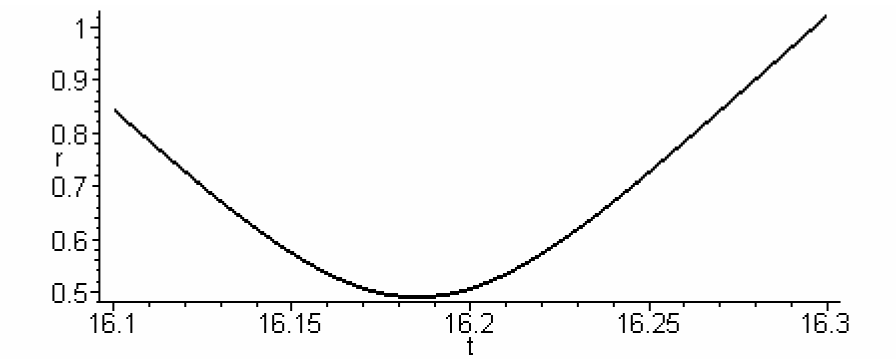
The comet appears to reach aphelion after about 8 years, and to return to perihelion after about 16 years. Zooming in on the aphelion,

```
odeplot(soln,[t,r(t)], 8.0..8.2);
```



we see that the comet reaches a maximal distance from the sun of about 12.31 AU after about 8.09 years. Zooming in on the perihelion,

```
odeplot(soln, [t,r(t)], 16.1..16.3);
```



we see that the comet appears to return to a minimal distance of about 0.49 AU from the sun after about 16.18 years.

Using *Mathematica*

Let's consider a comet orbiting the sun with initial position and velocity vectors

```
p0 = {x[0]==0.2, y[0]==0.4, z[0]==0.2}
v0 = {x'[0]==5, y'[0]==-7, z'[0]==9}
```

at perihelion. For convenience, we combine these initial conditions in the single set

```
inits = Union[p0,v0]
```

of equations. The comet's equations of motion in (1) with $\mu = 4\pi^2$ are entered as

```
r[t_] = Sqrt[x[t]^2 + y[t]^2 + z[t]^2]

de1 =    x''[t] == -4 Pi^2 x[t]/r[t]^3;
de2 =    y''[t] == -4 Pi^2 y[t]/r[t]^3;
de3 =    z''[t] == -4 Pi^2 z[t]/r[t]^3;

deqs = {de1,de2,de3}
```

The comet's *x*-, *y*-, and *z*-position functions then satisfy the combined set

```
eqs = Union[deqs, inits]
```

of three second-order differential equations and six initial conditions, which we proceed to solve numerically.

```
soln = NDSolve[eqs, {x, y, z}, {t, 0, 20}]
```

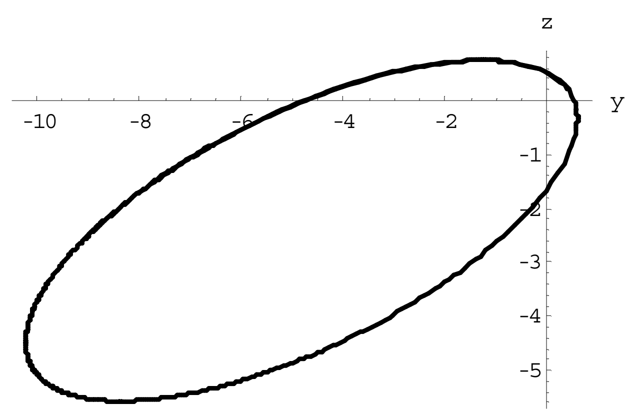
```
{{x -> InterpolatingFunction[{{0.,20.}}, <>],
  y -> InterpolatingFunction[{{0.,20.}}, <>],
  z -> InterpolatingFunction[{{0.,20.}}, <>]}}
```

The result **soln** is a list of three numerical "interpolating functions"

```
x = First[x /. soln];
y = First[y /. soln];
z = First[z /. soln];
```

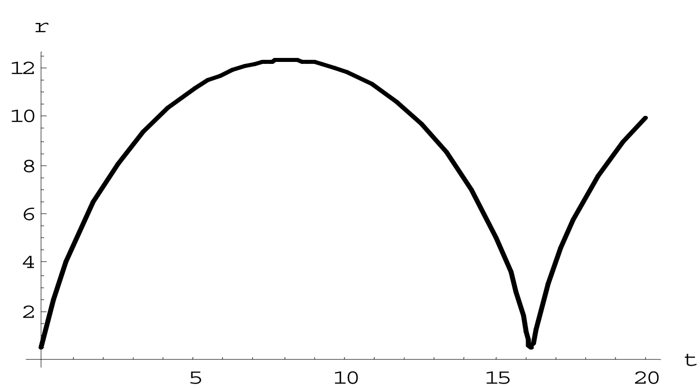
that we can use to plot the *yz*-projection of the comet's orbit for the first 20 years:

```
ParametricPlot[Evaluate[{y[t],z[t]}], {t,0,20}]
```



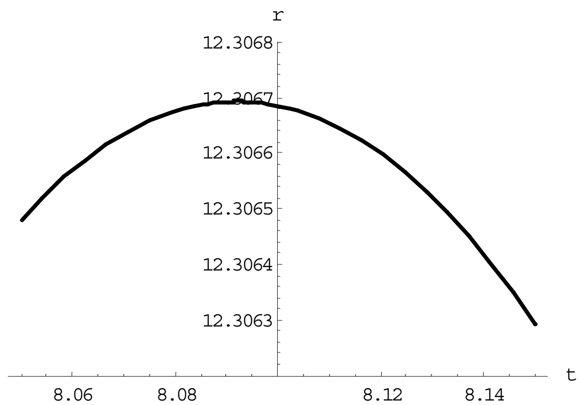
This certainly looks like an ellipse. To investigate the comet's motion on this orbit, we plot its distance r from the sun as a function of t.

```
Plot[Evaluate[r[t]], {t, 0, 20}]
```



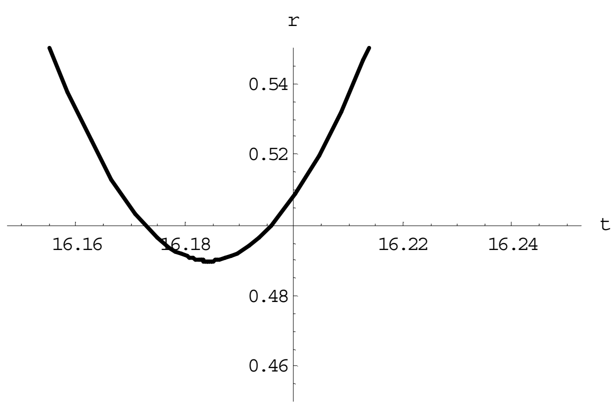
The comet appears to reach aphelion after about 8 years, and to return to perihelion after about 16 years. Zooming in on the aphelion,

```
Plot[Evaluate[r[t]], {t, 8.05, 8.15},
     PlotRange -> {12.3062, 12.3068}]
```



we see that the comet reaches a maximal distance from the sun of about 12.31 AU after about 8.09 years. Zooming in on the perihelion,

```
Plot[Evaluate[r[t]], {t, 16.15, 16.25},
     PlotRange -> {0.45, 0.55},
```



we see that the comet appears to return to a minimal distance of about 0.49 AU from the sun after about 16.18 years.

Using MATLAB

Let's consider a comet orbiting the sun with initial position and velocity column vectors

```
r0 = [0.2; 0.4; 0.2];
v0 = [5; -7; 9];
```

at perihelion. We combine these initial values into the single 6-component vector

```
inits = [p0; v0];
```

The following MATLAB function saved as **`ypcomet.m`** serves to define the comet's equations of motion in (1) with $\mu = 4\pi^2$.

```
function  yp = ypcomet(t,y)
yp = y;
vx = y(4);  vy = y(5);  vz = y(6);% velocity comps
x  = y(1);  z  = y(3);  y  = y(2);% coordinates
r  = sqrt( x*x + y*y + z*z );     % radius
r3  = r*r*r;                      % r-cubed
k = 4*pi^2;                       % for AU-yr units
yp(1) =  vx;
yp(2) =  vy;
yp(3) =  vz;
yp(4) =  -k*x/r3;
yp(5) =  -k*y/r3;
yp(6) =  -k*z/r3;
```

We proceed to solve these differential equations numerically with the given initial conditions.

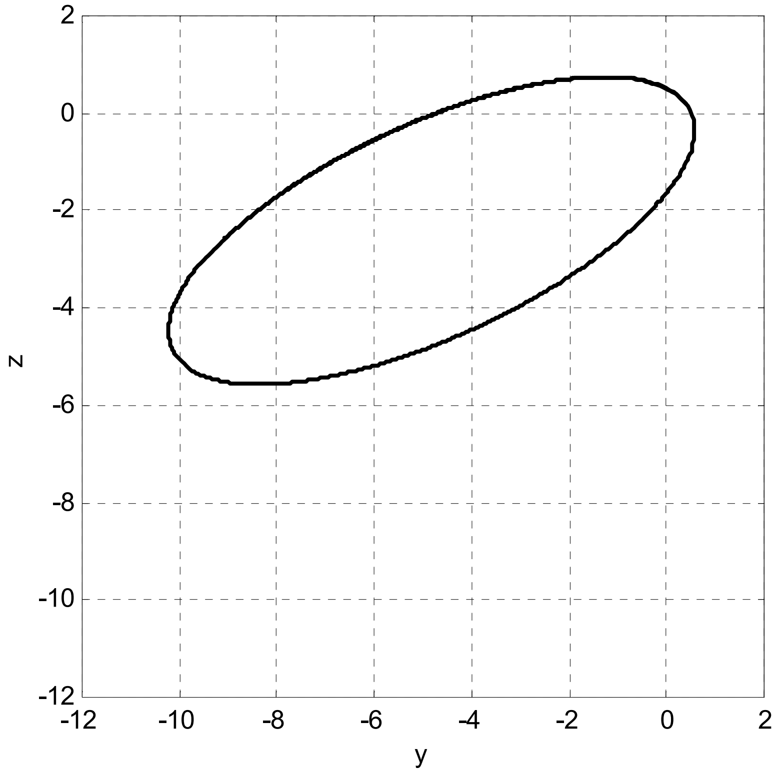
```
options = odeset('reltol',1e-6);     % error tolerance
tspan = 0 : 0.01 : 20;  % from t=0 to t=20 with dt=0.01

[t,y] = ode45('ypcomet',0:0.01:20, inits, options);
```

Here **`t`** is the vector of times and **`y`** is a matrix whose first 3 column vectors give the corresponding position coordinates of the comet. We need only plot the second and third of these vectors against each other to see the *yz*-projection of the comet's orbit for the first 20 years.

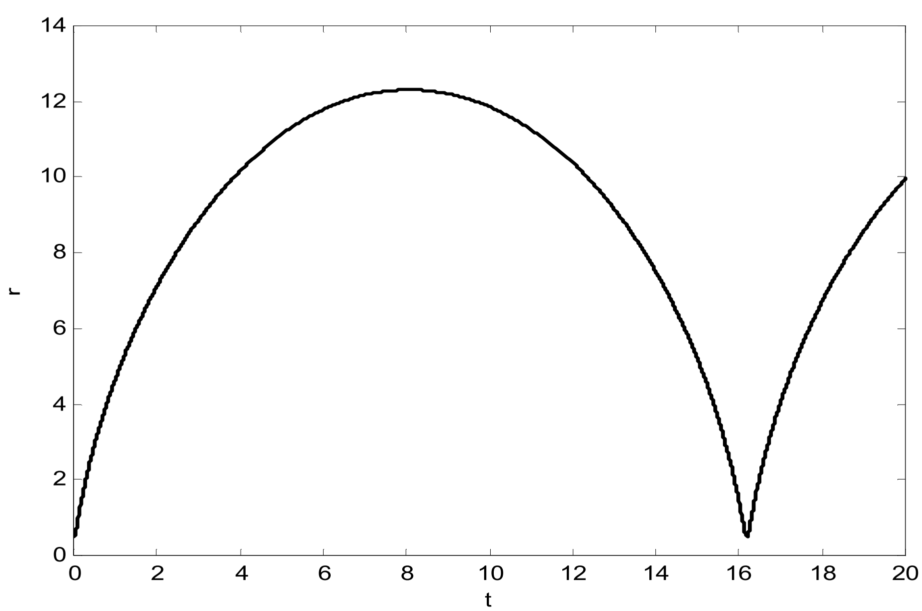
```
plot(y(:,2),y(:,3)),
axis([-12 2 -12 2]), axis square
```

The resulting orbit (at the top of the next page) certainly looks like an ellipse.



To investigate the comet's motion on this orbit, we plot its distance r from the sun as a function of t.

```
r = sqrt(y(:,1).^2 + y(:,2).^2 + y(:,3).^2);
plot(t, r)
```



The comet appears to reach aphelion after about 8 years, and to return to perihelion after about 16 years. We can zoom in on the aphelion with the command

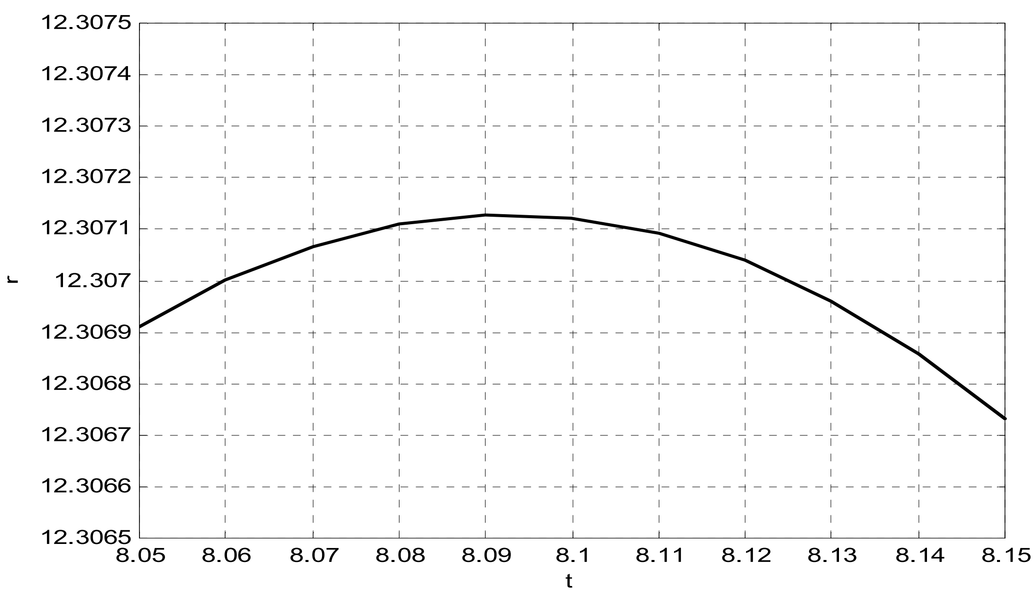
```
axis([8.05 8.15 12.3065 12.3075]), grid on
```

and see (in the figure below) that the comet reaches a maximal distance from the sun of about 12.31 AU after about 8.09 years.

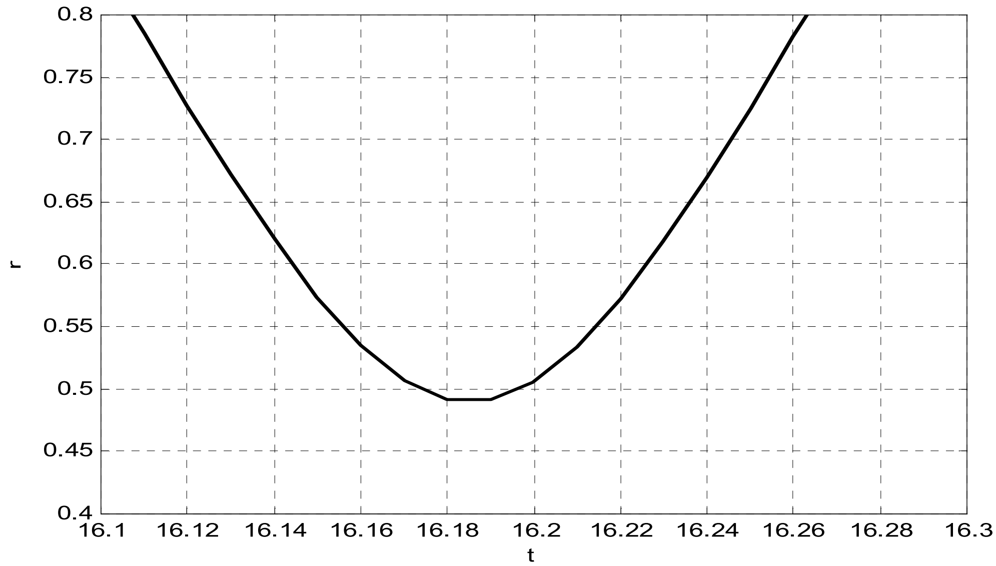
We zoom in on the perihelion with the command

```
axis([16.1 16.3 0.4 0.8]), grid on
```

and see (in the figure at the top of the next page) that it appears to return to a minimal distance of about 0.49 AU from the sun after about 16.18 years.



Zooming in on the comet's aphelion



Zooming in on the comet's perihelion

Earth-Moon Satellite Orbits

We consider finally an Apollo satellite in orbit about the Earth *E* and the Moon *M*. Figure 7.6.8 in the text shows an x_1x_2-coordinate system whose origin lies at the center of mass of the Earth and the Moon, and which rotates at the rate of one revolution per "moon month" of approximately $\tau = 27.32$ days, so the Earth and Moon remain fixed in their positions on the x_1-axis. If we take as unit distance the distance between the Earth and Moon centers, then their coordinates are $E(-\mu, 0)$ and $M(1-\mu, 0)$, where $\mu = m_M/(m_E + m_M)$ in terms of the Earth mass m_E and the Moon mass m_M. If we take the total mass $m_E + m_M$ as the unit of mass and $\tau/2\pi \approx 4.348$ days as the unit of time, then the gravitational constant has value $G = 1$, and the equations of motion of the satellite position $S(x_1, x_2)$ are

$$x_1'' = x_1 + 2x_2' - \frac{(1-\mu)(x_1+\mu)}{r_E^3} - \frac{\mu(x_1 - 1 + \mu)}{r_M^3} \tag{1a}$$

and

$$x_2'' = x_2 - 2x_1' - \frac{(1-\mu)x_2}{r_E^3} - \frac{\mu x_2}{r_M^3} \tag{1b}$$

where $r_E = \sqrt{(x_1+\mu)^2 + x_2^2}$ and $r_M = \sqrt{(x_1+\mu-1)^2 + x_2^2}$ denote the satellite's distance to the Earth and Moon, respectively. The initial two terms on the right-hand side of each equation in (1) result from the rotation of the coordinate system. In the system of units described here, the lunar mass is approximately $\mu = m_M = 0.012277471$. The second-order system in (1) can be converted to a first-order system by substituting

$$x_1' = x_3, \qquad x_2' = x_4$$

so

$$x_3' = x_1'', \qquad x_4' = x_2''. \tag{2}$$

This system is defined in the MATLAB function

```
function  yp = ypmoon(t,y)

m1 = 0.012277471;              % mass of moon
m2 = 0.987722529;              % mass of earth

r1 = norm([y(1)+m1, y(2)]);    % Distance to the earth
r2 = norm([y(1)-m2, y(2)]);    % Distance to the moon

yp = [ y(3); y(4); 0; 0 ];     % Column 4-vector

yp(3)  = y(1)+2*y(4)-m2*(y(1)+m1)/r1^3-m1*(y(1)-m2)/r2^3;

yp(4)  = y(2)-2*y(3)  - m2*y(2)/r1^3  - m1*y(2)/r2^3;
```

Suppose that the satellite initially is in a clockwise circular orbit of radius about 1500 miles about the Moon. At its farthest point from the Earth ($x_1 = 0.994$) it is "launched" into Earth-Moon orbit with initial velocity v_0. We then want to solve the system in (2) — with the right-hand functions in (1) substituted for x_1'' and x_2'' — with the initial conditions

$$x_1(0) = 0.994, \quad x_2(0) = 0, \quad x_3(0) = 0, \quad x_4(0) = -v_0. \tag{3}$$

In the system of units used here, the unit of velocity is approximately 2289 miles per hour. Some initial conditions and final times of particular interest are defined by the function

```
function  [tf,y0] = mooninit(k)

% Initial conditions for k-looped Apollo orbit
```

```
if  k == 2,
   tf = 5.436795439260;
   y0 = [ 0.994 0 0 -2.113898796695 ]';

elseif  k == 3,
   tf = 11.124340337266;
   y0 = [ 0.994 0 0 -2.031732629557 ]';

elseif  k == 4,
   tf = 17.065216560158;
   y0 = [ 0.994 0 0 -2.001585106379 ]';
   end
```

The first two components of **`y0`** are the coordinates of the initial position, and the last two components are the components of the initial velocity; **`tf`** is then the time required to complete one orbit. The cases $k = 3$ and $k = 4$ yield Figures 7.6.9 and 7.6.10 (respectively) in the text. The following commands (with $k = 2$) yield the figure shown on the next page, and illustrate how such figures are plotted.

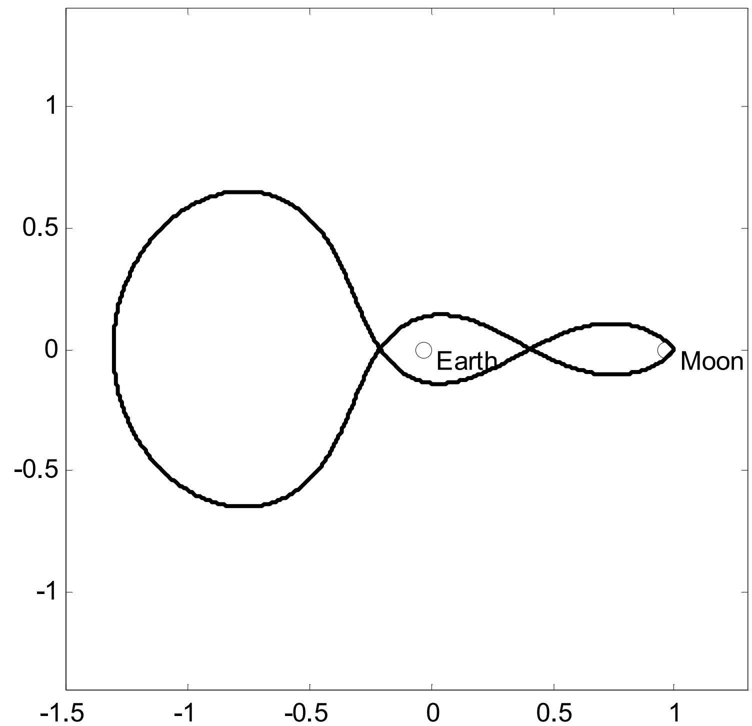
```
[tf,y0] = mooninit(2);
options = odeset('RelTol',1e-9,'AbsTol',1e-12);
[t,y] = ode45('ypmoon', [0,tf], y0, options);
plot(y(:,1), y(:,2));
axis([-1.5 1.3 -1.4 1.4]), axis square
```

The small relative and absolute error tolerances are needed to insure that the orbit closes smoothly when the satellite returns to its initial position.

You might like to try the values $k = 3$ and $k = 4$ to generate the analogous 3- and 4-looped orbits. A more substantial project would be to search empirically for initial velocities yielding periodic orbits with more than 4 loops.

Further Investigations

See the Application 7.6C page at the web site **www.prenhall.com/edwards** for additional investigations of comets, satellites, and trajectories of baseballs with air resistance (as in Example 4 of Section 7.6 in the text).



Earth
Moon