Chapter 7

Introduction to Systems of Differential Equations

Application 7.1
Kepler's Laws and Planetary Orbits

The Section 7.1 application in the text starts with Newton's inverse-square law of gravitation and outlines a derivation of the polar-coordinate formula

$$r(\theta) = \frac{L}{1+\varepsilon\cos(\theta-\alpha)} \tag{1}$$

describing an elliptical planetary orbit with eccentricity ε and semi-latus rectum L. The angle α is the planet's polar coordinate angle at perihelion — when it is closest to the sun. If the numerical values of L, ε, and α are known, then the ellipse can be plotted in rectangular coordinates by writing

$$x(t) = r(t)\cos t, \qquad y(t) = r(t)\sin t, \qquad 0 \le t \le 2\pi \tag{2}$$

(with parameter t in place of θ).

Here we want to use formulas (1) and (2) to plot some typical planetary orbits, starting with data found in a common source like a world almanac — where a planet's maximum and minimum distances r_M and r_m (respectively) from the sun typically are listed, but its semi-latus rectum is unlikely to be mentioned. But if we take $\alpha = 0$ in (1) then it should be clear that

$$r_M = \frac{L}{1-\varepsilon} \quad \text{and} \quad r_m = \frac{L}{1+\varepsilon}. \tag{3}$$

Upon equating values of L, these two equations are easily solved first for

$$\varepsilon = \frac{r_M - r_m}{r_M + r_m} \tag{4}$$

and then for

$$L = r_M(1-\varepsilon) = r_m(1+\varepsilon). \tag{5}$$

The initial columns in the table below list the maximum and minimum distances from the sun (in astronomical units, where 1 AU = 93 million miles is the mean distance of the Earth from the sun) of the nine planets and Halley's comet. The last two columns list values of ε and L calculated using Eqs. (4) and (5).

Planet	r_M	r_m	ε	L
Mercury	0.467	0.308	0.2056	0.371
Venus	0.728	0.718	0.0067	0.723
Earth	1.017	0.983	0.0172	1.000
Mars	1.667	1.382	0.0935	1.511
Jupiter	5.452	4.953	0.0480	5.190
Saturn	10.081	9.015	0.0558	9.518
Uranus	19.997	17.949	0.0540	18.918
Neptune	30.341	29.682	0.0110	30.008
Pluto	48.940	29.639	0.2456	36.919
Halley	35.304	0.587	0.9673	1.155

In the paragraphs below we illustrate the use of *Maple*, *Mathematica*, and MATLAB to plot typical planetary orbits. You can try these and others.

Using *Maple*

To plot the orbit of the planet Mercury, we first enter its maximum and minimum distances from the sun.

```
r1 := 0.467:
r2 := 0.308:
```

Then we calculate its eccentricity and semi-latus rectum using Eqs (4) and (5).

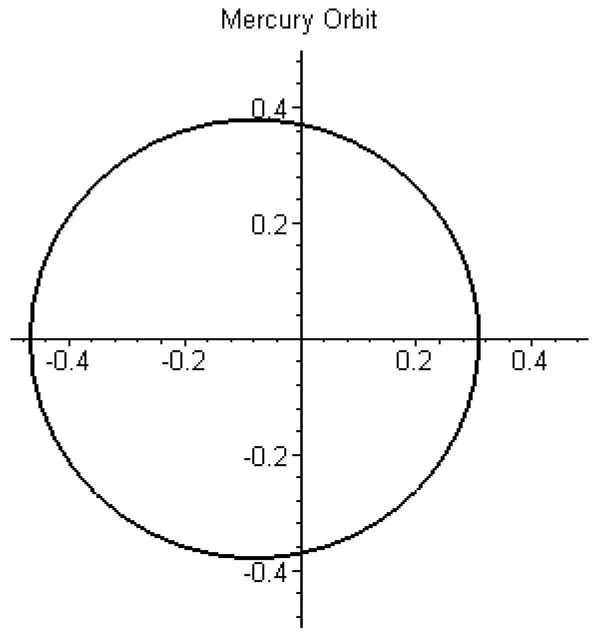
```
e := (r1 - r2)/(r1 + r2):
L := r1*(1-e):
```

Now we can calculate its polar and rectangular coordinate functions using (1) and (2).

```
r := L/(1+e*cos(t)):
x := r*cos(t):
y := r*sin(t):
```

and finally plot the orbit (with $\alpha = 0$).

```
plot([x,y,t=0..2*Pi],
     view=[-0.5..0.5, -0.5..0.5],
     thickness = 2, color = red,
     scaling = constrained,
     title = "Mercury Orbit");
```



The option **`scaling=constrained`** insures equal scales on the *x*- and *y*-axes. We see that the elliptical orbit of Mercury actually looks quite circular. The non-uniformity of the motion of Mercury consists in the facts that this "circle" is off-center from the sun at the origin, and that its speed varies with its position on the orbit, with the planet moving fastest at perihelion and slowest at aphelion.

Using *Mathematica*

To plot the orbit of the planet Mercury, we first enter its maximum and minimum distances from the sun.

```
r1 := 0.467:
r2 := 0.308:
```

Then we calculate its eccentricity and semi-latus rectum using Eq. (4) and (5).

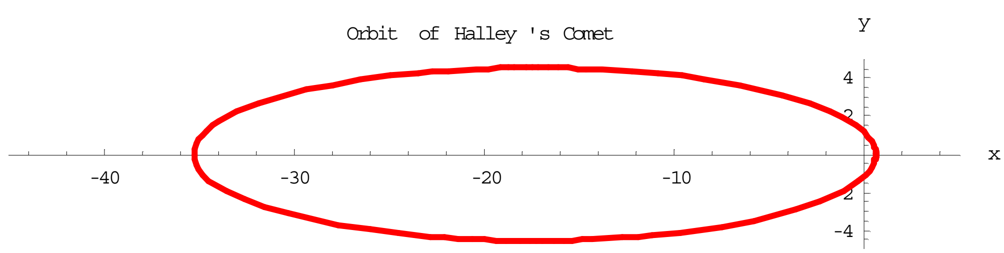
```
e := (r1 - r2)/(r1 + r2):
L := r1*(1-e):
```

Now we can calculate its polar and rectangular coordinate functions using (1) and (2).

```
r := L/(1+e*cos(t)):
x := r*cos(t):
y := r*sin(t):
```

and finally plot the orbit (with $\alpha = 0$).

```
ParametricPlot[{x, y}, {t, 0, 2*Pi},
   PlotRange -> {{-45, 5}, {-5, 5}},
   AspectRatio -> 0.2, AxesLabel -> {"x", "y"},
   PlotStyle -> {Thickness[0.0065],RGBColor[1,0, 0]},
        PlotLabel -> "Orbit of Halley's Comet"];
```



The option **AspectRatio->1** serves to ensure equal scales on the *x*- and *y*-axes. The plotted orbit certainly looks rather eccentric, though perhaps not so much as the actual eccentricity of $\varepsilon \approx 0.97$ might lead on to expect.

Using MATLAB

To plot the Earth's orbit, we first enter its maximum and minimum distances from the sun.

```
r1 = 1.017;
r2 = 0.983;
```

Then we calculate its eccentricity and semi-latus rectum using Eqs. (4) and (5).

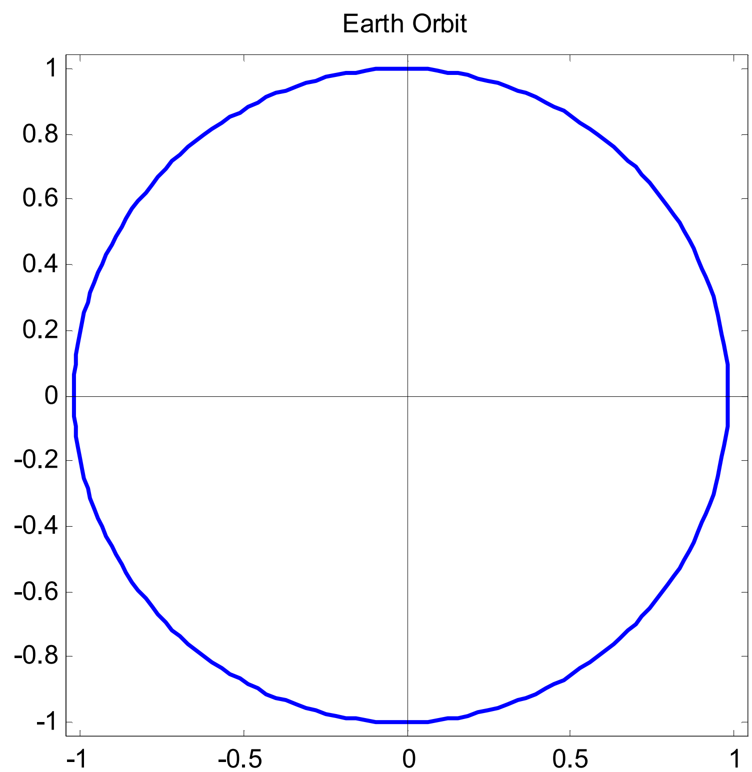
```
e = (r1 - r2)/(r1 + r2);
L = r1*(1-e);
```

Now we can calculate its polar and rectangular coordinate functions using (1) and (2).

```
t = 0 : pi/100 : 2*pi;
r = L./(1+e*cos(t));
x = r.*cos(t);
y = r.*sin(t);
```

and finally plot the orbit (with $\alpha = 0$).

```
h = plot(x,y,'b');
set(h,'linewidth',2);
w = 0.025;  % to set viewing window
axis(w*[-r1 r1 -r1 r1]), axis square
hold on
plot(w*[-r1 r1],[0 0],'k')
plot([0 0],w*[-r1 r1;],'k')
title('Earth Orbit')
```



The fact that the Earth's orbit is elliptical is so engrained in modern minds that the seemingly circular appearance of the orbit (on any reasonable scale) may come as a surprise. We have plotted the orbit in the square viewing window $-1.025 \le x, y \le 1.025$ so that a careful examination of the figure will reveal that this "circular" orbit is visibly off-center from the sun at the origin. This fact, together with the non-uniformity of the Earth's speed in its orbit — it moves fastest at perihelion and slowest at aphelion — constitutes the actual ellipticity of the motion.