Application 6.5B
Period Doubling and Chaos in Mechanical Systems

The first objective of this section is the application of the DE plotting techniques of the Section 6.3 and 6.4 applications to the investigation of mechanical systems that exhibit the phenomenon of period-doubling as a selected system parameter is varied.

The Forced Duffing Equation

Section 6.4 in the text introduces the second-order differential equation

$$mx'' + cx' + kx + \beta x^3 = 0$$  \hspace{1cm} (1)

to model the free velocity-damped vibrations of a mass $m$ on a nonlinear spring. Recall that the term $kx$ in Eq. (1) represents the force exerted on the mass by a linear spring, whereas the term $\beta x^3$ represents the nonlinearity of an actual spring. We want now to discuss the forced vibrations that result when an external force $F(t) = F_0 \cos \omega t$ acts on the mass. With such a force adjoined to the system in Eq. (1), we obtain the forced Duffing equation

$$mx'' + cx' + kx + \beta x^3 = F_0 \cos \omega t$$  \hspace{1cm} (2)

for the displacement $x(t)$ of the mass from its equilibrium position.

If $\beta = 0$ in (2) then we have a linear equation with stable periodic solutions. To illustrate the quite different behavior of a nonlinear system, we take $k = -1$ and $m = c = \beta = \omega = 1$ in Eq. (2), so the differential equation is

$$x'' + x' - x + x^3 = F_0 \cos t$$  \hspace{1cm} (3)

As an exercise you may verify that the two critical points $(-1, 0)$ and $(1, 0)$ are stable. We want to examine the dependence of the (presumably steady periodic) response $x(t)$ upon the amplitude $F_0$ of the periodic external force of period $2\pi$.

First verify that the values $F_0 = 0.60$ and $F_0 = 0.70$ yield the two figures shown on the next page, which indicate a simple oscillation about a critical point if $F_0 = 0.60$, and an oscillation with "doubled period" if $F_0 = 0.70$. In each case the equation was solved numerically with initial conditions $x(0) = 1$, $x'(0) = 0$ and the resulting solution plotted for the range $100 \leq t \leq 200$ (to show the steady periodic response remaining after the initial transient response has died out). Use $tx$-plots (as in the predator-prey investigation of the 6.3 application) to verify that the period of the oscillation with
$F_0 = 0.70$ is, indeed, twice the period with $F_0 = 0.60$. Then plot analogous figures with $F_0 = 0.75$ and with $F_0 = 0.80$ to illustrate successive period-doubling and finally chaos as the amplitude of the external force is increased in the range from $F_0 = 0.6$ to $F_0 = 0.8$. This period-doubling toward chaos is a common characteristic of the behavior of a nonlinear mechanical system as an appropriate physical parameter (such as $m$, $c$, $k$, $\beta$, $F_0$, or $\omega$) in Eq. (2) is increased or decreased. No such phenomenon occurs in linear systems.

Then investigate the parameter range $1.00 \leq F_0 \leq 1.10$ for the force constant in Eq. (3). With $F_0 = 1.00$ you should see a period $6\pi$ phase plane trajectory that encircles both stable critical points (as well as the unstable one). The period doubles around $F_0 = 1.07$ and chaos sets in around $F_0 = 1.10$. See whether you can spot a second period-doubling somewhere between $F_0 = 1.07$ and $F_0 = 1.10$. Produce both phase plane trajectories and $tx$-solution curves on which you can measure the periods.

**The Lorenz Strange Attractor**

The genesis of the famous 3-dimensional Lorenz system

\[
\begin{align*}
    x'(t) &= -sx + xy \\
    y'(t) &= -xz + rx - y \\
    z'(t) &= xy - bz
\end{align*}
\]

(4)

of differential equations is discussed in the text. A solution curve in $xyz$-space is best visualized by looking at its projection into some plane, typically one of the three coordinate planes. The figure on the next page shows the projection into the $xz$-plane of the solution obtained by numerical integration from $t = 0$ to $t = 30$ with the parameter
values \( b = 8/3, \ s = 10, \ r = 28 \) and the initial values \( x(0) = -8, \ y(0) = 8, \ z(0) = 27 \). As the projection in this figure is traced in "real time", the moving solution point \( P(x(t), y(t), z(t)) \) appears to undergo a random number of oscillations on the right followed by a random number of oscillations on the left, then a random number of the right followed by a random number on the left, and so on.

A close examination of such projections of the Lorenz trajectory shows that it is not simply oscillating back and forth around a pair of critical points (as the figure may initially suggest). Instead, as \( t \to \infty \), the solution point \( P(t) \) on the trajectory wanders back and forth in space approaching closer and closer to a certain complicated set of points whose detailed structure is not yet fully understood. This elusive set that appears somehow to “attract” the solution point is the famous *Lorenz strange attractor*.

First use an ODE plotting utility to reproduce the \( xz \)-projection of the Lorenz trajectory shown above. Use the parameter values and initial conditions listed above and numerically integrate the Lorenz system on the interval \( 0 \leq t \leq 30 \). Plot also the \( xy \)- and \( yz \)-projections of this same solution. Next, experiment with different parameter values and initial conditions. For instance, see if you can find a periodic solution with \( r = 70 \) (and \( b = 8/3, \ s = 10 \) as before) and initial values \( x_0 = -4 \) and \( z_0 = 64 \). To get a trajectory that almost repeats itself, you will need to try different values of \( y_0 \) in the range \( 0 < y_0 < 10 \) and look at \( xz \)-projections.
Another much-studied nonlinear three-dimensional system is the Rossler system

\[
\begin{align*}
    x'(t) &= -y - z \\
    y'(t) &= x + ay \\
    z'(t) &= b + z(x - c) 
\end{align*}
\] (5)

The figure below shows an \(xy\)-projection of the Rossler band, a chaotic attractor obtained with the values \(a = 0.398\), \(b = 2\), and \(c = 4\) of the parameters in (5). In the \(xy\)-plane the Rossler band looks “folded,” but in space it appears twisted like a Möbius strip. Investigate the period-doubling toward chaos that occurs with the Rossler system as the parameter \(a\) is increased, beginning with \(a = 0.3\), \(a = 0.35\), and \(a = 0.375\) (take \(b = 2\) and \(c = 4\) in all cases).

In the following paragraphs we illustrate ODE plotting techniques by showing how to use Maple, Mathematica, and MATLAB to plot the forced Duffing, Lorenz, and Rossler trajectories pictured here.
Using *Maple*

To plot a periodic trajectory for the forced Duffing equation (3) we need only define the system

\[
\begin{align*}
F_0 &:= 0.60; \\
m &:= 1; \quad c := 1; \quad k := -1; \quad b := 1; \quad w := 1; \\
deq1 &:= \text{diff}(x(t),t) = y; \\
deq2 &:= m \text{diff}(y(t),t) = -c y - k x - b x^3 + F_0 \cos(w t);
\end{align*}
\]

and then load the `DEtools` package and use the `DEplot` function. For instance, the command

\[
\text{with(DEtools):} \\
\text{DEplot([deq1,deq2], [x,y],} \\
t=100..200, \{[x(0)=1, y(0)=0]\}, \\
\text{stepsize=0.1,} \\
\text{linecolor=blue, arrows=none);}
\]

plots the periodic trajectory obtained with \( F_0 = 0.60 \).

To picture the Rossler band, we first define the Rossler system

\[
\begin{align*}
a &:= 0.398; \\
b &:= 2; \quad c := 4; \\
deq1 &:= \text{diff}(x(t),t) = -y - z; \\
deq2 &:= \text{diff}(y(t),t) = x + a y; \\
deq3 &:= \text{diff}(z(t),t) = b + z (x - c);
\end{align*}
\]

and then plot the desired \(xy\)-projection:

\[
\text{with(DEtools):} \\
\text{DEplot([deq1,deq2,deq3], [x,y,z],} \\
t=0..200, \{[x(0)=1,y(0)=1,z(0)=1]\}, \\
\text{stepsize=0.1,} \\
\text{scene = [x,y], thickness=0,} \\
\text{linecolor=blue, arrows=none);}
\]

**Using *Mathematica***

To plot a periodic trajectory for the forced Duffing equation (3) we need only define the system

\[
\begin{align*}
F_0 &= 0.60; \\
m &= 1; \quad c = 1; \quad k = -1; \quad b = 1; \quad w = 1;
\end{align*}
\]
deq1 = x'[t] == y[t];
deq2 = m y'[t] == -c y[t] - k x[t] - b x[t]^3 + F_0 \cos[w t];

and then use \texttt{NDSolve} to integrate numerically. For instance, the command

```
soln = NDSolve[ {deq1, deq2,
                  x[0]==1, y[0]==0},
                  {x[t],y[t]}, {t,0,200},
                  MaxSteps->2500 ];
```

yields an approximate solution on the interval \(0 \leq t \leq 200\) satisfying the initial conditions \(x(0)=1, y(0)=0\). Then the command

```
ParametricPlot[

Evaluate[{x[t],y[t]} /. soln], {t,100,200}];
```

plots the periodic trajectory corresponding to \(F_0 = 0.60\).

To picture the Rossler band, we first define the Rossler system

\[
\begin{align*}
a &= 0.398; \\
b &= 2; \quad c = 4; \\
deq1 &= x'[t] == -y[t] - z[t]; \\
deq2 &= y'[t] == x[t] + a y[t]; \\
deq3 &= z'[t] == b + z[t] (x[t] - c);
\end{align*}
\]

solve numerically,

```
soln = NDSolve[ {deq1, deq2, deq3,
                  x[0]==1, y[0]==1, z[0]==1},
                  {x[t],y[t],z[t]}, {t,0,200},
                  MaxSteps->3000];
```

and then plot the xy-projection:

```
ParametricPlot[

Evaluate[{x[t],y[t]} /. soln], {t,0,200}];
```

\textbf{Using MATLAB}

To plot a periodic trajectory for the forced Duffing equation (3) we first define the
MATLAB function

```
```
function yp = duffing(t,y)
F0 = 0.60;
m = 1; c = 1; k = -1; b = 1; w = 1;
yp = y;
x = y(1);
y = y(2);
yp(1) = y;
yp(2) = (-c*y - k*x - b*x^3 + F0*cos(t))/m;

as the m-file duffing.m defining the forced Duffing system. Note that it specifies the desired numerical values of the parameters in Eq. (2). Then the commands

[t,y] = ode45('duffing',0:0.1:100,[1;0]);
n = length(t);
[t,y] = ode45('duffing',100:0.1:200,y(n,:));
plot(y(:,1),y(:,2));

plot the periodic xy-trajectory corresponding to $F_0 = 0.60$. Note that we solve first from $t = 0$ to $t = 100$, then use the final value here as the initial value in solving from $t = 100$ to $t = 200$.

To plot the Lorenz strange attractor, we first define the MATLAB function

function yp = lorenz(t,y)
yp = y;
b = 8/3; s = 10; r = 28;
x = y(1);
z = y(3);
y = y(2);
yp(1) = -s*x + s*y;
yp(2) = -x.*z + r*x - y;
yp(3) = x.*y - b*z;

as the m-file lorenz.m. Then we solve numerically

[t,y] = ode45('lorenz',0:0.005:30,[-9;8;27]);

and finally plot the $xz$-projection:

plot(y(:,1),y(:,3))