Iterated Maps

*sysid*

Mathematica 6.0.3, DynPac 11.01, 1/13/2009

plotreset;

intreset;

- **Functions and Variables Used in This Tutorial**

asprat, axon, bifurcmap, bifurc3Dmap, bimap, boxrat, classifymap, cobweb, colorvec, eigsysmap, eigvalmap, findpolyfix, frameon, imsize, intreset, iterate, jacob, jacobval, mapcomp, mapval, nfndfix, nfndpolyfix, outbound, parmval, parmvec, periodmap, phaser, phaser3D, plotreset, plrange, plrange3D, pointcon, portraitmap, portrait3Dmap, psize, rangeflag, ranger, residualfix, setback, setcolor, setde, setmap, setparm, setstate, show, slopevec, staterange, statevec, stripsol, sysid, sysreport, timeplot, and viewmap.

- **Description of Systems Used in This Tutorial**

In this tutorial, our objective is to illustrate the use of the functions defined for iterated mappings. For examples, we will use the logistic map for a 1D case, the Henon map for a 2D case, and a combination of the two for a 3D case. In many cases, it is useful to apply some of the functions directly to iterates of the map. For example, if we are studying a map f[x], then one way of finding orbits of period two is to look for fixed points of f[f[x]]. Most of the functions used in DynPac for mappings allow an optional final argument which is the level of composition desired. We will see a number of examples of this below.

- **Logistic Map**

The logistic map is discussed in many references. A very complete and readable discussion is given in Chapter 10 of *Nonlinear Dynamics and Chaos* by Steven Strogatz, Addison-Wesley, 1994. Many of the interesting properties of the map were discovered by the mathematical biologist Robert May ("Simple Mathematical Models with Very Complicated Dynamics," Nature 261, 459, 1976.) The basic form of the map is

\[ x_{n+1} = rx_n(1 - x_n) \]

As is well-known this map exhibits a wide and interesting range of behavior as \( r \) is varied. We define the system for
DynPac, starting by the setmap command. This command tells DynPac that we are working with a mapping rather than a differential equation.

```
setmap;
setstate[{x}]; setparm[{r}]; parmval = {3.2}; slopevec = {r * x * (1 - x)};
```

```
sysreport
SYSTEM DEFINITION (11.01)
```

```
System name: sysname = System
State vector: statevec = {x}
State units: stateunits = {}
Slope vector: slopevec = {r (1 - x) x}
Parameter vector: parmvec = {r}
Parameter values: parmval = {3.2}
Parameter units vector: parmunits = {}
Time unit: timeunit =
System Type: sysmode = mapping
```

We could use this same function as the slope for a differential equation. The command setde switches back to differential equation mode. The primary difference in the two modes is the actual stepping algorithm used in constructing solutions -- a Runge-Kutta step for a differential equation, and a map iteration for the mapping. It is only at that basic level of code that the two modes differ.

```
setde;
```
sysreport

SYSTEM DEFINITION (11.01)

System name: sysname = System
State vector: statevec = \{x\}
State units: stateunits = {} 
Slope vector: slopevec = \{r(1-x)x\}
Parameter vector: parmvec = \{r\}
Parameter values: parmval = \{3.2\}
Parameter units vector: parmunits = {} 
Time unit: timeunit =
System Type: sysmode = differential equation

We return to the map setting.

setmap;

We start by viewing the map.

imsize = 250;

viewmap[]

\[ r(1-x)x, \{r\} = \{3.2\} \]

The picture suggests that there are two fixed points -- one at 0 and one between 0.6 and 0.8. We find these. Because the mapping is a polynomial, we can use findpolyfix or nfindpolyfix. We can also use the more general nfindfix, which requires an initial guess.
\[ \text{findpolyfix[]} \]
\[ \{0, \left\{ \frac{1 + r}{r} \right\} \} \]

This gives the answer in terms of the parameter \( r \). To find numerical values we can use \text{nfindpolyfix} or \text{nfindfix}:

\[ \text{nfindpolyfix[]} \]
\[ \{0., \{0.6875\}\} \]
\[ \text{nfindfix[0.5]} \]
\[ \{0.6875\} \]

We can check the accuracy of the fixed point by finding the residual with \text{residualfix}.

\[ \text{residualfix[\{0.6875\}] } \]
\[ \{0.\} \]

Alternatively, we can evaluate the map at the fixed point.

\[ \text{mapval[\{0.6875\}]} \]
\[ \{0.6875\} \]

We check the stability of these two fixed points.

\[ \text{classifymap[\{0\}]} \]
\[ \text{unstable} \]
\[ \text{classifymap[\{0.6875\}] } \]
\[ \text{unstable} \]

Thus both of the fixed points are unstable. This could also be determined by the eigenvalues at those points. Any eigenvalue greater than one in magnitude indicates instability.

\[ \text{eigvalmap[\{0\}]} \]
\[ \{3.2\} \]
\[ \text{eigvalmap[\{0.6875\}] } \]
\[ \{-1.2\} \]

As neither fixed point is stable for this value of \( r \), there might be a periodic orbit. Let's perform a short iteration with a more or less arbitrary initial condition of 0.23. The four arguments of \text{iterate} are the initial value, the initial time, the total number of iterates, and the number thrown away before keeping the results. In this case, we start at 0.23, do 20 iterations, and keep them all.
We see four fixed points. Of course two will be the fixed points of the original map, but the other two should be the points on the period-two orbit of the original map. We check this.

\texttt{nfindpolyfix[2]}

\{(0.), (0.513045), (0.6875), (0.799455)\}

We see the same two values that showed up explicitly in the orbit calculated above. We check the stability of the period.
two orbit by checking the stability of these as fixed points of the second iterated mapping.

\[ \text{classifymap}[[0.513045], 2] \]
strictly stable

\[ \text{classifymap}[[0.799455], 2] \]
strictly stable

Thus the period two orbit is stable.

The last few function evaluations have provided examples of applying functions to higher compositions of the map -- in this case the second composition. It is the optional last argument 2 that causes this. We haven't yet looked explicitly at the formula for the second composition, although we can do that easily with \text{mapcomp}[n], which returns the nth composition.

\[ \text{mapcomp}[2] \]
\[ \{ r^2 (1 - x) x (1 - r (1 - x) x) \} \]

Now we look at something new. We construct a cobweb plot to show the approach to the stable orbit of period 2. The function to do this is \text{cobweb}[initial, niter, ntoss], where initial is the starting point for the iteration, niter is the number of iterations to plot, and ntoss is the number to calculate and throw away first (to eliminate transients). We start with an initial condition 0.23, and we ask for 100 iterations, throwing none away before plotting.

\[ \text{cobweb}[[0.23], 100, 0] \]

We see the eventual approach to the orbit. We can get the pure orbit by throwing away the transients. We perform the same calculation, only now throwing away 100 points first.
Now we get the pure orbit.

Let's increase the value of \( r \) to 3.5, and carry out a short iteration.

\[
\text{parmval} = \{3.5\};
\]
\[
\text{sol2} = \text{iterate}[\{0.23\}, 0.0, 30, 0]
\]
\[
\\{\{0., 0.23\}, \{1., 0.61985\}, \{2., 0.824726\}, \{3., 0.505936\}, \{4., 0.874877\},
\{5., 0.383136\}, \{6., 0.8272\}, \{7., 0.500291\}, \{8., 0.875\}, \{9., 0.38282\},
\{10., 0.826935\}, \{11., 0.500896\}, \{12., 0.874997\}, \{13., 0.38282\},
\{14., 0.826941\}, \{15., 0.500884\}, \{16., 0.874997\}, \{17., 0.38282\},
\{18., 0.826941\}, \{19., 0.500884\}, \{20., 0.874997\}, \{21., 0.38282\},
\{22., 0.826941\}, \{23., 0.500884\}, \{24., 0.874997\}, \{25., 0.38282\},
\{26., 0.826941\}, \{27., 0.500884\}, \{28., 0.874997\}, \{29., 0.38282\}, \{30., 0.826941\}\}
\]

A close inspection shows an orbit of period 4. We verify that.

\[
\text{periodmap[sol2]}
\]
Solution contains a periodic orbit; period = 4

The points on the 4-orbit should be fixed points of the fourth composition of the map.

\[
\text{nfindpolyfix[4]}
\]
\[
\\{\{0., \{0.049385 - 0.0241573 i\}, \{0.049385 + 0.0241573 i\}, \{0.166354 - 0.0761994 i\},
\{0.166354 + 0.0761994 i\}, \{0.38282\}, \{0.428571\}, \{0.500884\},
\{0.505703 - 0.177965 i\}, \{0.505703 + 0.177965 i\}, \{0.714286\}, \{0.826941\},
\{0.857143\}, \{0.874997\}, \{0.985737 - 0.00710473 i\}, \{0.985737 + 0.00710473 i\}\}\]
We find lots of roots. In the interpretation, it helps to look also at the fixed points of the basic mapping and the mapping iterated once.

\[
\text{nfindpolyfix[2]}\]
\[
\{\{0.\}, \{0.428571\}, \{0.714286\}, \{0.857143\}\}
\]
\[
\text{nfindpolyfix[]}\]
\[
\{\{0.\}, \{0.714286\}\}
\]

By comparing these, we conclude that (1) the basic map has fixed points

\[
f_{11} = 0.0; f_{12} = 0.714286;
\]

(2) the second composition has in addition a period two orbit

\[
f_{21} = 0.428571; f_{22} = 0.857143;
\]

and (3) the fourth composition has a period four orbit

\[
f_{41} = 0.38282; f_{42} = 0.500884; f_{43} = 0.826941; f_{44} = 0.874997;
\]

Thus for \( r = 3.5 \), the map has two fixed points, a period two orbit and a period four orbit. Yet all we saw in the iteration was the period four orbit. This strongly suggests that it is the only stable attractor. We check the stability of all of these now.

\[
\text{classifymap[\{f11\}]}
\]
unstable

\[
\text{classifymap[\{f12\}]}
\]
unstable

\[
\text{classifymap[\{f21\}, 2]}
\]
unstable

\[
\text{classifymap[\{f22\}, 2]}
\]
unstable

\[
\text{classifymap[\{f41\}, 4]}
\]
strictly stable

\[
\text{classifymap[\{f42\}, 4]}
\]
strictly stable

\[
\text{classifymap[\{f43\}, 4]}
\]
strictly stable
classifymap[{f44}, 4]

strictly stable

Let's look at a cobweb of this stable period four orbit.

cobweb[{0.23}, 100, 0]

Now we look at the pure orbit.
We increase $r$ again, this time to $3.7$.

```
parmval = {3.7};

sol3 = iterate[{0.23}, 0, 100, 0]
```

{{0, 0.23}, {1, 0.65527}, {2, 0.835798}, {3, 0.507788}, {4, 0.924776}, {5, 0.257393}, {6, 0.707225}, {7, 0.766114}, {8, 0.662979}, {9, 0.82672}, {10, 0.530039}, {11, 0.921661}, {12, 0.267146}, {13, 0.724383}, {14, 0.738713}, {15, 0.714159}, {16, 0.755303}, {17, 0.683835}, {18, 0.799958}, {19, 0.592094}, {20, 0.893619}, {21, 0.351736}, {22, 0.843666}, {23, 0.488007}, {24, 0.924468}, {25, 0.25836}, {26, 0.708958}, {27, 0.763446}, {28, 0.668206}, {29, 0.820315}, {30, 0.545374}, {31, 0.917383}, {32, 0.28043}, {33, 0.746619}, {34, 0.699663}, {35, 0.77055}, {36, 0.64099}, {37, 0.851451}, {38, 0.467984}, {39, 0.921207}, {40, 0.268562}, {41, 0.726815}, {42, 0.734653}, {43, 0.72127}, {44, 0.743846}, {45, 0.704995}, {46, 0.769515}, {47, 0.656238}, {48, 0.834682}, {49, 0.510555}, {50, 0.924588}, {51, 0.257983}, {52, 0.708283}, {53, 0.764487}, {54, 0.666172}, {55, 0.822831}, {56, 0.539387}, {57, 0.91926}, {58, 0.274618}, {59, 0.73705}, {60, 0.717086}, {61, 0.750632}, {62, 0.69258}, {63, 0.787779}, {64, 0.618579}, {65, 0.872974}, {66, 0.410293}, {67, 0.895225}, {68, 0.347049}, {69, 0.838443}, {70, 0.501189}, {71, 0.924995}, {72, 0.256704}, {73, 0.705986}, {74, 0.766008}, {75, 0.659235}, {76, 0.831183}, {77, 0.519175}, {78, 0.92364}, {79, 0.260959}, {80, 0.71358}, {81, 0.756219}, {82, 0.82102}, {83, 0.802034}, {84, 0.586865}, {85, 0.897082}, {86, 0.341606}, {87, 0.832172}, {88, 0.516748}, {89, 0.923962}, {90, 0.259997}, {91, 0.711787}, {92, 0.759042}, {93, 0.67672}, {94, 0.809494}, {95, 0.570692}, {96, 0.90651}, {97, 0.313575}, {98, 0.796409}, {99, 0.599925}, {100, 0.888056}}

Now there is no obvious repetition. We check it for periodicity anyway.
Solution does not contain a periodic orbit.

Are there fixed points?

\[
\text{nfindpolyfix[1]} \\quad \{\{0.\}, \{0.72973\}\}
\]

\[
\text{nfindpolyfix[2]} \\quad \{\{0.\}, \{0.390022, 0.72973, 0.880248\}\}
\]

\[
\text{nfindpolyfix[4]} \\quad \{\{0.\}, \{0.0447134 - 0.015327 \, i, 0.0447134 + 0.015327 \, i, 0.158911 - 0.0516384 \, i, 0.158911 + 0.0516384 \, i, 0.321626, 0.390022, 0.504402 - 0.130338 \, i, 0.504402 + 0.130338 \, i, 0.575652, 0.72973, 0.807276, 0.880248, 0.903824, 0.987784 - 0.00424622 \, i, 0.987784 + 0.00424622 \, i\}\}
\]

There are two fixed points, a period two orbit and a period four orbit. Presumably they are all unstable, since they don't show up in our iteration. Are there orbits of higher period? If we try nfindpolyfix[8], \textit{Mathematica} goes away for a very long time and returns with a large list of roots, mostly complex, and missing most of the real roots that we already know from above. Let's look at the stability of the orbits we have found. It is sufficient to check just one point on each orbit if that one point is unstable.

\[
\text{classifymap[\{0\}]} \quad \text{unstable}
\]

\[
\text{classifymap[\{0.72973\}]} \quad \text{unstable}
\]

\[
\text{classifymap[\{0.390022\}, 2]} \quad \text{unstable}
\]

\[
\text{classifymap[\{0.321626\}, 4]} \quad \text{unstable}
\]

If this system with \(r = 3.7\) has a stable attractor, we haven't found it yet. We try a cobweb plot.
This could be chaotic. Let's do a fancy cobweb plot now with color, showing the sensitive dependence on initial conditions. We take 26 initial points in the range \([0.20,0.25]\), and we assign a gradually varying color from red to blue. First we construct the color list.

```
collist = Table[RGBColor[(1 - i), 0, i], \{i, 0, 1, 0.04\}];
```

Because we don't have colors specified by name, we will assign the list directly to colorvec, rather than using the command setcolor.

```
colorvec = collist; collpoint = 1;
```

Now we construct a list of initial conditions.

```
initvec = Table[0.2 + i, \{i, 0, 0.05, 0.002\}];
```

Now we construct the cobweb plot.

```
plrange = \{\{0, 1\}, \{0, 1\}\}; asprat = 1;
imsize = 380;
```
We can also make a plot of $x$ versus time by using timeplot. We construct a solution with 50 iterates and then plot it as a function of time.

```
plrange = {[0, 50], [0, 1]}; asprat = 0.7; setcolor[Black];
soltime = iterate[{0.23}, 0, 50, 0];
pointcon = True;
```
Both the time plot and the cobweb plot show the chaotic nature of this orbit. In particular the cobweb plot shows clearly the spread of an initially compact set of initial conditions. With a bifurcation diagram, we can get an overview of how the system behavior depends on the parameter $r$. This is done with the function bimap, which produces a graph in which the abcissa is the parameter being varied, and along the ordinate, the iterates are plotted. If we throw away initial transients, the result is a plot of the attractor of the system as a function of the parameter. The function which does this is

$$\text{bimap}[\text{npts}, \text{ntoss}, \text{nparm}, \text{xname}, \text{xrange}, \text{initvec}, \text{pname}, \text{prange}, \text{ncomp}]$$

The first two arguments are npts and ntoss. The total number of iterates calculate is npts + ntoss. Then the first ntoss are thrown away, and the remaining iterates (npts in number) are plotted. The number of parameter values plotted is nparm. The name of the state variable plotted is xname, and its plotting range is xrange. The initial condition for the iteration is initvec, which may contain parameter symbols. The name of the parameter being varied is pname, and it is varied through the range prange. The final argument ncomp is optional and specifies the level of function composition of the basic map. Generally for the logistic map about 200 iterates have to be thrown away to get good results, and about 100 to 200 iterates have to be plotted, at about 100 to 200 parameter values. We carry this out for $r$ in the range 2.8 to 4. We ask for a larger image size, for a background color of Wheat, and points plotted in Blue.

$$\text{setback}[@\text{Wheat}]; \text{imsize} = 400; \text{setcolor}[@\{\text{Blue}\}]$$

$$\text{ptsiz}e = 0.002; \text{asprat} = 0.7$$
We see clearly the bifurcation from a stable fixed point to a stable orbit of period 2 at \( r = 3 \), and then the bifurcation from period two to period four at \( r \) between 3.4 and 3.5. The further period doublings occur at decreasing increments in \( r \), and the orbit becomes chaotic for \( r \approx 3.57 \). Note the intriguing window just beyond 3.8. Let’s explore this briefly. We set \( r \) to 3.83. We iterate and throw away 100 initial points in an effort to get rid of the transients.

\[
\text{parmval} = \{3.83\};
\]

\[
\text{sol4} = \text{iterate}[\{0.23\}, 0, 30, 100]
\]

\[
\{(100, 0.957417), (101, 0.156149), (102, 0.504666), (103, 0.957417), (104, 0.156149),
(105, 0.504666), (106, 0.957417), (107, 0.156149), (108, 0.504666), (109, 0.957417),
(110, 0.156149), (111, 0.504666), (112, 0.957417), (113, 0.156149), (114, 0.504666),
(115, 0.957417), (116, 0.156149), (117, 0.504666), (118, 0.957417),
(119, 0.156149), (120, 0.504666), (121, 0.957417), (122, 0.156149),
(123, 0.504666), (124, 0.957417), (125, 0.156149), (126, 0.504666),
(127, 0.957417), (128, 0.156149), (129, 0.504666), (130, 0.957417)\}
\]

A surprising result -- a period 3 orbit!

\[
\text{periodmap[sol4]}
\]

Solution is periodic; period = 3

Because we saw it, it surely is stable, but we can check that.
classifymap[0.504666, 3]
strictly stable

classifymap[0.957414, 3]
strictly stable

classifymap[0.156149, 3]
strictly stable

Let's look at the cobweb plot for this.

asprat = 1;
plrange = {{0, 1}, {0, 1}};
cobweb[0.23, 50, 0]
Now we throw away the transients and look at the periodic orbit.

```math
\text{cobweb}[\{0.23\}, 15, 100]
```

We look at the third iterated mapping.
The third iterated map has 8 fixed points. Two of these are unstable fixed points of the basic map. The other six turn out to be the components of two period 3 orbits, one stable and one unstable. From the iteration carried out above, we know that the components of the stable orbit are 0.156149, 0.504666, and 0.957417. The unstable period 3 orbit then is \{0.16357, 0.524001, and 0.955294\}. Let's start on this orbit and iterate, and see what happens.
Although it isn't very strongly unstable, we can see that it is drifting off the orbit. Here's a longer run.

```math
sol7 = iterate[{0.16357}, 0.0, 1000, 0];
lastx
```

Thus after a 1000 steps starting on the unstable period 3 orbit, we end up on the stable period 3 orbit.

### Hénon Map

The Hénon map is a two-dimensional map developed by Michel Hénon to study chaos and strange attractors ("A Two-Dimensional Mapping with a Strange Attractor," Commun. Math. Phys. 50, 69, 1976). It is discussed in many books on dynamical systems -- for example in section 12.2 of *Nonlinear Dynamics and Chaos* by Steven Strogatz, Addison-Wesley, 1994. The Hénon mapping provides a computationally straightforward way to study the complexities of chaos. The mapping has two parameters $a$ and $b$. It is given by

$$ x_{n+1} = y_n + 1 - ax_n^2, \quad y_{n+1} = bx_n. $$

We define the system for DynPac.

```math
setstate[{x, y}]; setparm[{a, b}]; sysname = "Henon"; setmap;
slopevec = {y + 1 - a x^2, b x}; parmval = {1.4, 0.3};
```
sysreport

SYSTEM DEFINITION (11.01)

System name: sysname = Henon
State vector: statevec = \{x, y\}
State units: stateunits = \{, \}
Slope vector: slopevec = \{-a x^2 + y + 1, b x\}
Parameter vector: parmvec = \{a, b\}
Parameter values: parmval = \{1.4, 0.3\}
Parameter units vector: parmunits = \{, \}
Time unit: timeunit =
System Type: sysmode = mapping

These parameter values were found by Hénon to produce a chaotic attractor.

We begin by calculating the Jacobian of the map.

```math
jacob
-b
```

```math
jacobval
-0.3
```

Thus the map will be dissipative (giving contracting areas in phase space) when \(|b| < 1\). We construct a very lengthy iteration -- 50000 points. We throw away the first 100 points. One reason for calculating so many points is that we plan to zoom in on the structure of the attractor produced. Notice how much quicker this is than solving a differential equation for the same number of time steps.

```math
solhen = iterate\{0, 0\}, 0.0, 50000, 100\};
```

We use staterange to determine a suitable plotting window.

```math
staterange[solhen]
\{\{x, \{-1.28466, 10.581.\}, \{1.27297, 43.604.\}\},
\{y, \{-0.385397, 10.582.\}, \{0.381892, 43.605.\}\}\}
```

```math
plrange = \{\{-1.5, 1.5\}, \{-0.5, 0.5\}\}; axon = False; frameon = True; asprat = 0.7;
```
This graph shows the chaotic attractor for the map. There is an incredible fine structure which does not show up on this scale. We zoom in by choosing a plotting window near the point \(\{0.5, 0.2\}\). In this zoom process, we follow closely the presentation in Strogatz’ book of Hénon’s work. We define a graphics zoombox that will put a box on a graph to show the window used in the following graph.

```math
zoombox[rng_] := Graphics[
{GrayLevel[0.7], Rectangle[First[rng], Last[rng]]}, DisplayFunction -> Identity]
```
The grey rectangle is the entire window for the next zoomed graph.

```
plrange = {{0.54, 0.71}, {0.15, 0.21}};
```
We zoom in once again.
Once again, the grey rectangle is the window for the next (and last) plot.

```
plrange = {{0.62, 0.64}, {0.185, 0.191}};
```
The unfolding of structure is amazing. In each case what appears to be three lines, becomes, when viewed more closely, a set of 6 lines, grouped as three, two and one. The fractal nature of the attractor is such that this unfolding continues forever. Of course whether or not we see it in our graphs depends on how many iterates we have calculated. With our 50000 points we can't go much further than we have here.

We can explore the sensitive dependence on initial position by looking at two orbits starting close together. We use red and blue for the orbit colors, and we carry out the construction using the function portraitmap. We throw away the first 100 iterations and then keep 1000 iterates.

```
setcolor[{Red, Blue}];
pts = 0.008; plrange = {{-1.5, 1.5}, {-0.5, 0.5}};
```
We can see from the colors that both orbits are spread over the attractor. We can also see the sensitive dependence from a time plot.

```math
plrange = {{0, 50}, {-1.52, 1.52}};
sol1 = iterate[{{0, 0}, 0.0}, 50, 0];
sol2 = iterate[{{0.01, 0.01}, 0.0}, 50, 0];
pointcon = True;
```
It is of interest to see how the iterates of the Hénon map depend on the parameter $a$. We leave $b$ fixed and vary $a$ through 5 values, constructing a phase plot for each value. We accomplish this in an automated way by using the function \texttt{bifurcmap[intlist,t0,niter,ntoss,i,j,parmlist,ncomp]}. The argument \texttt{intlist} is the list of initial conditions to be used in each graph. In this case, we use only one initial condition. The initial time is $t_0$ and the number of iterations is $n_{iter}$, with $n_{toss}$ being thrown away first. The components $i$ and $j$ are to be plotted. The list of parameter values to be used is in \texttt{parmlist}. The final argument \texttt{ncomp} is optional. It is the level of function composition to be used with the basic map, and the default is 1.

\begin{verbatim}
plrange = ((-1.5, 1.5), (-0.5, 0.5)); pointcon = False;

parmlist = {{0.4, 0.3}, {0.6, 0.3}, {0.8, 0.3}, {1.0, 0.3}, {1.2, 0.3}, {1.4, 0.3}};

pts,axis = 0.008; setcolor([Black]);

bifurcmap[{{0, 0}}, 0.0, 1000, 100, 1, 2, parmlist]
\end{verbatim}
Henon \( \{a, b\} = \{0.40, 0.30\} \)

Henon \( \{a, b\} = \{0.60, 0.30\} \)
Henon \( \{a, b\} = \{0.80, 0.30\}\)

Henon \( \{a, b\} = \{1.00, 0.30\}\)
An interesting sequence. Looks like stable period two orbits for $a = 0.4, 0.6, \text{ and } 0.8$, followed by a period four orbit for $a = 1.0$, and then chaotic orbits for $a = 1.2$ and $1.4$. Let's look in more detail at the period four orbit.
parmval = {1.0, 0.3};

henfour = iterate[{0, 0}, 0.0, 20, 200]

{{200., -0.656352, 0.382493}, {201., 0.951695, -0.196906}, {202., -0.102629, 0.285508},
 {203., 1.27498, -0.0307886}, {204., -0.656352, 0.382493}, {205., 0.951695, -0.196906},
 {206., -0.102629, 0.285508}, {207., 1.27498, -0.0307886}, {208., -0.656352, 0.382493},
 {209., 0.951695, -0.196906}, {210., -0.102629, 0.285508}, {211., 1.27498, -0.0307886},
 {212., -0.656352, 0.382493}, {213., 0.951695, -0.196906}, {214., -0.102629, 0.285508},
 {215., 1.27498, -0.0307886}, {216., -0.656352, 0.382493}, {217., 0.951695, -0.196906},
 {218., -0.102629, 0.285508}, {219., 1.27498, -0.0307886}, {220., -0.656352, 0.382493}}

Clearly period 4, as DynPac also tells us:

periodmap[henfour]

Solution is periodic; period = 4

We name the four points of the orbit.

pt1 = Drop[First[henfour], 1]

{-0.656352, 0.382493}

pt2 = Drop[henfour[[2]], 1]

{0.951695, -0.196906}

pt3 = Drop[henfour[[3]], 1]

{-0.102629, 0.285508}

pt4 = Drop[henfour[[4]], 1]

{1.27498, -0.0307886}

We check by hand the orbit.

mapval[pt1]

{0.951695, -0.196906}

mapval[pt2]

{-0.102629, 0.285508}

mapval[pt3]

{1.27498, -0.0307886}

mapval[pt4]

{-0.656352, 0.382493}

From the calculation above, it is clear that the orbit is stable, but we check one point on it anyway.
From what we have found, it appears that the Hénon map goes through a period-doubling sequence to chaos. We construct a view of this with a bifurcation diagram, varying $a$ from 0.2 to 1.4, and plotting $x$ versus $a$. We throw away the first 300 points and then plot 200, for each of 200 values of the parameter.

```math
asprat = 0.7; ptsize = 0.002;
setcolor[{Blue}];
biouthen = bimap[200, 300, 200, x, {-1.5, 1.5}, {0, 0}, a, {0.4, 1.4}]
```

This diagram suggests lots of other things to explore later, but we leave it for now.

One final point on the Hénon map. Not all of the solutions are bounded. For some initial conditions and some parameter values, the iterates run off to infinity. We explore this briefly.

```math
parmval = {1.7, 0.3};
```
We set a range flag to True and specify the range on 3D plots.

```plaintext
iterate[{1, 1}, 0.0, 20, 0]

{0., 1, 1}, {1., 0.3, 0.3}, {2., 1.147, 0.09}, {3., -1.14654, 0.3441},
{4., -0.890623, -0.343961}, {5., -0.692418, -0.267187},
{6., -0.082239, -0.207725}, {7., 0.780777, -0.0246717}, {8., -0.0610137, 0.234233},
{9., 1.2279, -0.0183041}, {10., -1.58148, 0.368371}, {11., -2.88346, -0.474444},
{15., -4.81911\times10^{10}, -50.510.3}, {16., -3.94805\times10^{21}, -1.44573\times10^{10}},
{17., -2.64981\times10^{42}, -1.18442\times10^{71}}, {18., -1.19365\times10^{87}, -7.94942\times10^{62}},
```

This is clearly heading for an overflow! To prevent overflows and possible crashes, we can use range checking, in which we specify a box (named ranger) to contain the solution. The iteration is stopped as soon as the solution leaves the box. To turn on range checking, we set rangeflag = True.

```plaintext
rangeflag = True; ranger = {{-1000, 1000}, {-1000, 1000}};

iterate[{1, 1}, 0.0, 20, 0]

{0., 1, 1}, {1., 0.3, 0.3}, {2., 1.147, 0.09}, {3., -1.14654, 0.3441},
{4., -0.890623, -0.343961}, {5., -0.692418, -0.267187},
{6., -0.082239, -0.207725}, {7., 0.780777, -0.0246717}, {8., -0.0610137, 0.234233},
{9., 1.2279, -0.0183041}, {10., -1.58148, 0.368371}, {11., -2.88346, -0.474444},
{15., -4.81911\times10^{10}, -50.510.3}, {16., -3.94805\times10^{21}, -1.44573\times10^{10}},
{17., -2.64981\times10^{42}, -1.18442\times10^{71}}, {18., -1.19365\times10^{87}, -7.94942\times10^{62}},
```

We see that the iteration was stopped when a solution point first left the box. The first point outside the box is stored in the variable outbound.

```plaintext
outbound

{{14., -168.368, -94.4109}}

rangeflag = False;
```

A Three-Dimensional Map

Most published work on iterated maps deals with one- and two-dimensional maps. Such maps can exhibit the full range of complexities, unlike the situation with sets of autonomous differential equations where dimension three is the minimum dimension for chaos. All of this is by way of saying that there is no natural or well-studied example to use here. In order to illustrate some of the 3D plotting features for maps, we define a somewhat artificial 3D map consisting of the logistic map in one dimension and the Hénon map in the other two. The mapping is defined by

\[ x_{n+1} = y_n + 1 - ax_n^2, \quad y_{n+1} = bx_n, \quad z_{n+1} = rz_n(1 - z_n). \]

We define this system for Dynpac.

```plaintext
Setstate[{x, y, z}]; Setparm[{a, b, r}]; Sysname = "Hybrid"; Parmval = {0.8, 0.3, 2.8}; Slopevec = {y + 1 - a x^2, b x, r z (1 - z)}; Setmap;
```
**sysreport**

**SYSTEM DEFINITION (11.01)**

- **System name**: \( \text{sysname} = \text{Hybrid} \)
- **State vector**: \( \text{statevec} = \{x, y, z\} \)
- **State units**: \( \text{stateunits} = \{, , \} \)
- **Slope vector**: \( \text{slopevec} = \{-ax^2 + y + 1, bx, r(1 - z)z\} \)
- **Parameter vector**: \( \text{parmvec} = \{a, b, r\} \)
- **Parameter values**: \( \text{parmval} = \{0.8, 0.3, 2.8\} \)
- **Parameter units vector**: \( \text{parmunits} = \{, , \} \)
- **Time unit**: \( \text{timeunit} = \)
- **System Type**: \( \text{sysmode} = \text{mapping} \)

On the basis of work done earlier, we expect for the present parameter values to get an orbit of period 2, in which \( z \) is stationary. Let's try it.
solhyl = iterate[{0, 0, 0.5}, 0.0, 75, 0]

{{0.0, 0.0, 0.5}, {1.0, 1.0, 0.7}, {2.0, 0.2, 0.3, 0.588}, {3.0, 1.268, 0.06, 0.678317},
 {4.0, -0.226259, 0.3804, 0.610969}, {5.0, 1.33945, -0.067878, 0.665521},
 {6.0, -0.503169, 0.401834, 0.623288}, {7.0, 1.19929, -0.150951, 0.657444},
 {8.0, -0.301589, 0.359787, 0.630595}, {9.0, 1.28702, -0.0904766, 0.652246},
 {10.0, -0.415618, 0.386107, 0.6351}, {11.0, 1.24792, -0.124685, 0.648895},
 {12.0, -0.370521, 0.374375, 0.637925}, {13.0, 1.26455, -0.111156, 0.646735},
 {14.0, -0.390418, 0.379364, 0.639713}, {15.0, 1.25742, -0.117125, 0.645345},
 {16.0, -0.382015, 0.377227, 0.64085}, {17.0, 1.26048, -0.114605, 0.644452},
 {18.0, -0.305649, 0.378143, 0.641574}, {19.0, 1.25916, -0.115695, 0.643879},
 {20.0, -0.384089, 0.377749, 0.642037}, {21.0, 1.25973, -0.115227, 0.643511},
 {22.0, -0.384762, 0.377919, 0.642332}, {23.0, 1.25949, -0.115429, 0.643276},
 {24.0, -0.384472, 0.377846, 0.642521}, {25.0, 1.25959, -0.115342, 0.643125},
 {26.0, -0.384597, 0.377877, 0.642642}, {27.0, 1.25955, -0.115379, 0.643029},
 {28.0, -0.384543, 0.377864, 0.642727}, {29.0, 1.25957, -0.115363, 0.642967},
 {30.0, -0.384566, 0.37787, 0.642769}, {31.0, 1.25956, -0.11537, 0.642927},
 {32.0, -0.384556, 0.377867, 0.642801}, {33.0, 1.25956, -0.115367, 0.642902},
 {34.0, -0.38456, 0.377868, 0.642821}, {35.0, 1.25956, -0.115368, 0.642886},
 {36.0, -0.384559, 0.377868, 0.642834}, {37.0, 1.25956, -0.115368, 0.642876},
 {38.0, -0.384559, 0.377868, 0.642842}, {39.0, 1.25956, -0.115368, 0.642869},
 {40.0, -0.384559, 0.377868, 0.642848}, {41.0, 1.25956, -0.115368, 0.642865},
 {42.0, -0.384559, 0.377868, 0.642851}, {43.0, 1.25956, -0.115368, 0.642862},
 {44.0, -0.384559, 0.377868, 0.642853}, {45.0, 1.25956, -0.115368, 0.64286},
 {46.0, -0.384559, 0.377868, 0.642855}, {47.0, 1.25956, -0.115368, 0.642859},
 {48.0, -0.384559, 0.377868, 0.64286}, {49.0, 1.25956, -0.115368, 0.642858},
 {50.0, -0.384559, 0.377868, 0.642866}, {51.0, 1.25956, -0.115368, 0.642858},
 {52.0, -0.384559, 0.377868, 0.642866}, {53.0, 1.25956, -0.115368, 0.642858},
 {54.0, -0.384559, 0.377868, 0.642876}, {55.0, 1.25956, -0.115368, 0.642858},
 {56.0, -0.384559, 0.377868, 0.642876}, {57.0, 1.25956, -0.115368, 0.642858},
 {58.0, -0.384559, 0.377868, 0.642876}, {59.0, 1.25956, -0.115368, 0.642858},
 {60.0, -0.384559, 0.377868, 0.642876}, {61.0, 1.25956, -0.115368, 0.642858},
 {62.0, -0.384559, 0.377868, 0.642876}, {63.0, 1.25956, -0.115368, 0.642858},
 {64.0, -0.384559, 0.377868, 0.642876}, {65.0, 1.25956, -0.115368, 0.642858},
 {66.0, -0.384559, 0.377868, 0.642876}, {67.0, 1.25956, -0.115368, 0.642858},
 {68.0, -0.384559, 0.377868, 0.642876}, {69.0, 1.25956, -0.115368, 0.642858},
 {70.0, -0.384559, 0.377868, 0.642876}, {71.0, 1.25956, -0.115368, 0.642858},
 {72.0, -0.384559, 0.377868, 0.642876}, {73.0, 1.25956, -0.115368, 0.642858},
 {74.0, -0.384559, 0.377868, 0.642876}, {75.0, 1.25956, -0.115368, 0.642858}}

periodmap[solhyl]

Solution contains a periodic orbit; period = 2

Let's plot the approach to this in 3D.

plotreset; pointcon = False; boxrat = (1, 1, 1);
ptsise = 0.01; p医疗器械3D = {{-1.5, 1.5}, {-0.5, 0.5}, {0, 1}};
A projection on the xy plane:

\texttt{frameon = True; asprat = 1; axon = False;}

\begin{verbatim}
phaser3D[solhy1]
\end{verbatim}

\texttt{Hybrid \{a, b, r\} = \{ 0.80, 0.30, 2.80\}}
To get the pure orbit, we iterate again, this time throwing away the first 100 iterates.

```math
solhy2 = iterate[{0, 0, 0.5}, 0.0, 20, 100];

periodmap[solhy2]
```

Solution is periodic; period = 2
We see the two points of the orbit in diagonally opposite corners. Now we try a portrait in 3D with several different initial conditions. We also change $a$ so that the Hénon map is chaotic.

```plaintext
parmval = {1.4, 0.3, 2.8};

plotreset; ptsize = 0.008; pointcon = False; boxrat = {1, 1, 1};
setcolor[{Red, Blue}]; plrange3D = {{-1.5, 1.5}, {-0.5, 0.5}, {0, 1}};
```
Finally we will do a short bifurcation sequence, letting both \( r \) and \( a \) vary, using the command \texttt{bifurc3Dmap}.

\begin{verbatim}
setback[Wheat];
parmlist = {{1.0, 0.3, 2.8}, {1.2, 0.3, 3.5}, {1.4, 0.3, 3.7}};
bifurc3Dmap[{{0, 0, 0.23}, {0.01, 0.01, 0.25}}, 0.0, 2000, 200, 1, 2, 3, parmlist]
\end{verbatim}

Bifurcation sequence for \( \text{parmlist} = \{\{1.0, 0.3, 2.8\}, \{1.2, 0.3, 3.5\}, \{1.4, 0.3, 3.7\}\} \)
Hybrid \{a, b, r\} = \{ 1.00, 0.30, 2.80\}
Hybrid \( \{a, b, r\} = \{1.20, 0.30, 3.50\} \)
Hybrid \( \{a, b, r\} = \{1.40, 0.30, 3.70\} \)

This last graph shows a kind of double chaos, and perhaps that is as good a place to quit as any.