1. Introduction

In this notebook, we develop some of the basic properties of the tent map. We use the functions defined in DynPac for iterated maps. These functions are discussed in detail in Tutorial 17, which is also available on the examples page (listed as Iterated Maps) of the web site for this course. We begin by defining the mapping for DynPac. The parameter in the mapping is called $\mu$. The command `setmap` tells DynPac that we are studying a mapping rather than a differential equation.

```
setState[{x}]; setparm[{\mu}]; sysname = "Tent Map"; setmap;
```

The slope vector for the tent map is

```
slopevec = {2*\mu*(x + (1 - 2*x)*UnitStep[x - 1/2])};
```

We use `sysreport` to check our definitions.

```
sysreport

SYSTEM DEFINITION (10.67)

System name sysname = Tent Map
State vector statevec = {x}
State units stateunits = {}
Slope vector slopevec = {2*\mu*(x + (1 - 2*x)*UnitStep[-1/2 + x])}
Parameter vector parmvec = {\mu}
Parameter values parmval = {\mu}
Parameter units vector parmunits = {}
Time unit timeunit =
System Type = mapping
```
The domain of the map is $[0,1]$, and for $0 < \mu < 1$ the range is contained in the interval $[0,1]$. We now assign $\mu$ the value 1.

```
parmval = {1};
```

We use the function `viewmap` to plot the map, along with the reference line $x$. The argument 1 in `viewmap` indicates that we want to view the mapping itself. For an argument $n$, we get a view of the $n$th composite mapping.

```
imsize = 380;
viewmap[1];
```

In dynamical systems, we are interested in the result of iterating the map -- that is applying it repeatedly. For example, suppose we start with the value $x = 0.23$. We call this $x_0$. We associate an initial time $t_0$ of 0 with this initial $x$. The first iterate is the mapping function evaluated at 0.23, and the time associated with the first iterate is 1. The DynPac function which carries out iterations is `iterate[x0,t0,n,nthrow]`. Here $n + nthrow$ iterations are calculated, and the first $nthrow$ are discarded (this allows us to discard unwanted transients in certain cases). Thus to get the first 10 iterates and to keep them all, we type

```
iterate[0.23, 0, 10, 0]
```

```
{{0, 0.23}, {1, 0.46}, {2, 0.92}, {3, 0.16}, {4, 0.32}, {5, 0.64},
 {6, 0.72}, {7, 0.56}, {8, 0.88}, {9, 0.24}, {10, 0.48}}
```

We can do this in a less automated way. To get the value of the map at say $x = 0.23$, we can use the function `mapval`:

```
mapval[0.23]
```

```
{0.46}
```

To get directly the 10th iterate, we type
This last is slow, because it is first constructing the 10th composite of the map and then evaluating the composite at 0.23. The function iterate is much faster.

We will shortly see a very nice way to visualize the iterations, using a kind of plot called a cobweb plot. Before we do that, let's consider the basic concepts of orbits, fixed points and stability.

### 2. Fixed Points and Stability

We consider an iteration $x_{n+1} = f(x_n)$, starting with some given initial value $x_0$. The set of points $\{x_0, x_1, \ldots, x_n, \ldots\}$ is called an orbit of the system. For each initial value $x_0$ we get such an orbit. Such orbits display the same diversity of behavior as the orbits of differential equations which we have been studying all semester. A very important special case is the analogue of an equilibrium point for a differential equation. Let's see what form that takes for an iterated mapping. Suppose we have found a point $x_*$ which has the property that $x_* = f(x_*)$. Such a point is called a fixed point of the mapping. Clearly if we start on or land on such a point we will stay there forever, so the point is an equilibrium point of our discrete dynamical system. An easy graphical way to look for fixed points is to plot $f[x]$ and $ref[x] = x$ on the same plot. The plotting function viewmap automatically includes the reference line. Let's look at the picture again.

```math
viewmap[];
```

We see that there are two fixed points. The obvious one at $x = 0$, and another at a point easily calculated to be $x = \frac{2}{3}$. Let's check this.
We can also ask Mathematica to find the fixed points for us. We use the function nfindfix[initialguess].

\begin{verbatim}
mapval[0] = {0}
mapval[2/3] = {2/3}
\end{verbatim}

We can ask Mathematica to find the fixed points for us. We use the function nfindfix[initialguess].

\begin{verbatim}
nfindfix[0.1] = {0.}
nfindfix[0.6] = {0.666667}
\end{verbatim}

Here's an interesting example which has a transient approach to equilibrium. We start with $x = 1/12$, and iterate 10 times.

\begin{verbatim}
iterate[1/12, 0, 10, 0]
\end{verbatim}

\begin{verbatim}
iterate[1/12, 0, 60, 0]
\end{verbatim}

We reach the fixed point after three steps and then stay there. Notice that we stay exactly on the fixed point once we reach it. Does that mean the fixed point is stable? Surprisingly, the answer is no. Mathematica does exact arithmetic on rational numbers, so that there is no numerical error in the above calculation. If we use decimal numbers, we get quite a different result.
Now we see that eventually we have drift away from the fixed point at 2/3 and end up at the fixed point 0. Does this mean that the fixed point at 0 is stable? Once again the answer is no. To show this numerically, we start at 0.00001 and iterate 50 times.

```
iterate[0.00001, 0, 50, 0]
```

```
{{0, 0.00001}, {1, 0.00002}, {2, 0.00004}, {3, 0.00008}, {4, 0.00016}, {5, 0.00032}, {6, 0.00064}, {7, 0.00128}, {8, 0.00256}, {9, 0.00512}, {10, 0.01024}, {11, 0.02048}, {12, 0.04096}, {13, 0.08192}, {14, 0.16384}, {15, 0.32768}, {16, 0.65536}, {17, 0.68928}, {18, 0.62144}, {19, 0.57572}, {20, 0.48576}, {21, 0.97152}, {22, 0.05696}, {23, 0.11392}, {24, 0.22784}, {25, 0.45568}, {26, 0.91136}, {27, 0.17728}, {28, 0.35456}, {29, 0.70912}, {30, 0.58176}, {31, 0.83648}, {32, 0.32704}, {33, 0.65408}, {34, 0.69184}, {35, 0.61632}, {36, 0.76736}, {37, 0.46528}, {38, 0.93056}, {39, 0.13888}, {40, 0.27776}, {41, 0.55552}, {42, 0.88896}, {43, 0.22208}, {44, 0.44416}, {45, 0.88832}, {46, 0.22336}, {47, 0.44672}, {48, 0.89344}, {49, 0.21312}, {50, 0.426241}}
```

We have moved away from the fixed point at 0. Thus neither of the two fixed points is stable. Could we have predicted these instabilities directly from the mapping function? As we showed in class, the answer is yes, and the key quantity is the derivative of the mapping function. We repeat here the argument given in class. Consider the difference between two successive iterates.

\[
x_{n+2} - x_{n+1} = f(x_{n+1}) - f(x_n) = (x_{n+1} - x_n) f'(\bar{x})
\]

where \( \bar{x} \) is some point between \( x_n \) and \( x_{n+1} \). If we are near the fixed point \( x_*= \) and the iterates are close, then \( \bar{x} = x_* \). Thus the difference between successive iterates is either magnified or diminished in absolute value according as \( |f'(x_*)| \) is either greater than 1 or less than 1. If the difference is magnified, then the fixed point is unstable. Our basic test is to evaluate \( |f'(x_*)| \). If it is less than one, the fixed point is stable, and if it is greater than 1 the fixed point is unstable. Only if it is equal to 1 is there no conclusion from the test. For the tent map with \( \mu = 1 \), we have

```
D[First[slopeval], x] /. x -> 0
```

```
2
```

```
D[First[slopeval], x] /. x -> 2/3
```

```
-2
```

Thus our test tells us unambiguously that these are unstable fixed points. Of course this conclusion depends on the value of \( \mu \). For general \( \mu \) we have

```
D[First[slopevec], x]
```

```
2 \mu \left( 1 + 2 (1 - 2 x) \text{DiracDelta}[-1 + 2 x] - 2 \text{UnitStep}\left[-\frac{1}{2} + x\right] \right)
```
Thus the value of the derivative is either $2\mu$ or $-2\mu$, depending on the value of $x$. In either case the absolute value of the derivative is $2\mu$, and so any fixed points are unstable for $\mu > 1/2$ and stable for $\mu < 1/2$. We look at this in detail for $\mu = 1/3$.

parmval = {1/3};
viewmap[];

\[2\mu \left(1 - 2 \text{UnitStep}\left[-\frac{1}{2} + x\right]\right), \quad (\mu) = \left\{\frac{1.00}{3.00}\right\}\]

The only fixed point is $x = 0$, and it is clear that this is true for any $\mu < 1/2$. According to our analysis, this fixed point should be stable. We look at some iterates starting at $x = 0.2$:

iterate[0.2, 0, 20, 0]

\[
\{(0, 0.2), (1, 0.133333), (2, 0.0888889), (3, 0.0592593), (4, 0.0395062), (5, 0.0263374), (6, 0.0175583), (7, 0.0117055), (8, 0.00780369), (9, 0.00520246), (10, 0.00346831), (11, 0.0023122), (12, 0.00154147), (13, 0.00102765), (14, 0.000685097), (15, 0.000456732), (16, 0.000304488), (17, 0.000202992), (18, 0.000135328), (19, 0.0000902186), (20, 0.0000601457)\}
\]

This is clearly converging to 0.

As an alternative approach to stability, we may use the DynPac function classifymap. This function is analogous to the function classify for equilibria of differential equations. We check our fixed point $x = 0$:

classifymap[0]

strictly stable
3. Periodic Orbits

What is the analogue, for a mapping, of a periodic orbit for a differential equation? The key concept is that the state repeats after a finite time. Thus a periodic orbit here is one in which a value of $x$ is repeated after a certain number of iterations. To make our task specific, we will search for orbits of the tent map of period two. That is, the orbit (apart from any transient approach) will consist of two points $a$ and $b$, such that $a = f(b)$ and $b = f(a)$. How do we find such a pair of points? There is a simple observation which makes it easier, namely that $a = f(b) = f(f(a))$. Thus any point of an orbit of period two will be a fixed point of $f \circ f$. The only minor downside of this approach is that we will also pick up the fixed points of $f$ itself. Let's try this after resetting $\mu$ to 1. We use the function `viewmap[2]` to plot the second function composition of the current mapping.

```
parmval = (1);
viewmap[2];
```

We see four fixed points. All of them are either equilibria or orbits of period two. It is obvious from the graph that two of the points are 0 and 2/3, the fixed points of the basic mapping. The other two points must be the two points on a single orbit of period two. It is a simple calculation to show that those two points are 2/5 and 4/5. We check those values, using `mapval` for the second composition.

```
mapval[2 / 5, 2]

\{ \frac{2}{5} \}

mapval[4 / 5, 2]

\{ \frac{4}{5} \}
```
We can also use iterate to see the orbit:

\[
\text{iterate[2/5, 0, 10, 0]}
\]

\[
\{ \{0, \frac{2}{5}\}, \{1, \frac{4}{5}\}, \{2, \frac{2}{5}\}, \{3, \frac{4}{5}\}, \{4, \frac{2}{5}\}, \{5, \frac{4}{5}\}, \{6, \frac{2}{5}\}, \{7, \frac{4}{5}\}, \{8, \frac{2}{5}\}, \{9, \frac{4}{5}\}, \{10, \frac{2}{5}\} \}
\]

There is our orbit of period two. Is the orbit stable? We can answer that by looking at the derivative of the second composition at either point. The derivative of a composite function \(f(f(x))\) is \(f'(f(x)) \cdot f'(x)\). For the tent map \(f'\) is \(\pm 2\), so the absolute value of the derivative of the composite is 4, and the orbit is unstable. The instability will show up if we use a decimal number rather than an exact integer and if we iterate long enough.

\[
\text{iterate[0.4, 0, 50, 0]}
\]

\[
\{ \{0, 0.4\}, \{1, 0.8\}, \{2, 0.4\}, \{3, 0.8\}, \{4, 0.4\}, \{5, 0.8\}, \{6, 0.4\}, \{7, 0.8\}, \{8, 0.4\}, \{9, 0.8\}, \{10, 0.4\}, \{11, 0.8\}, \{12, 0.4\}, \{13, 0.8\}, \{14, 0.4\}, \{15, 0.8\}, \{16, 0.4\}, \{17, 0.8\}, \{18, 0.4\}, \{19, 0.8\}, \{20, 0.4\}, \{21, 0.8\}, \{22, 0.4\}, \{23, 0.8\}, \{24, 0.4\}, \{25, 0.8\}, \{26, 0.4\}, \{27, 0.8\}, \{28, 0.4\}, \{29, 0.8\}, \{30, 0.4\}, \{31, 0.8\}, \{32, 0.4\}, \{33, 0.8\}, \{34, 0.4\}, \{35, 0.799999\}, \{36, 0.400002\}, \{37, 0.800003\}, \{38, 0.399994\}, \{39, 0.799988\}, \{40, 0.400024\}, \{41, 0.800049\}, \{42, 0.399902\}, \{43, 0.799805\}, \{44, 0.400391\}, \{45, 0.800781\}, \{46, 0.398438\}, \{47, 0.796875\}, \{48, 0.40625\}, \{49, 0.8125\}, \{50, 0.375\}\}
\]

Clearly we have drifted off the orbit. There are orbits of higher period also, and you can use the same techniques to find them.

We can also check the stability with classifymap, applied to the second composition.

\[
\text{classifymap[2/5, 2]}
\]

unstable

Of course we will get the same result at 4/5:

\[
\text{classifymap[4/5, 2]}
\]

unstable

## 4. Cobweb Plots

Now we are going to introduce a standard and very helpful graphical view of the iteration process. The cobweb plot makes use of a plot of the mapping function and a plot of the reference line \(f(x) = x\). From these two curves one may construct graphically an iteration sequence. We construct the plot first and then explain it. We do this for the case of three iterations. The DynPac function which does this is cobweb[init,niter,nthrow,ncomp]. Here init is the starting value of \(x\), niter is the number of iterations, nthrow is the number to be thrown away before constructing the plot, and ncomp is the level of function composition. The argument ncomp is optional, and if it is omitted a default of 1 is used.
This shows a cobweb plot of three iterations for the tent map. The starting value is $x = 0.23$. Starting at that value on the $x$-axis, we move vertically upward until we hit the graph of $\text{tent}[x]$. The $y$-value at that point is the value of the first iterate. Now we move horizontally until we hit the reference line $y = x$. The $x$-coordinate of that point is the value of the iterate. Next we move vertically until we reach the graph of $\text{tent}[x]$. The $y$-coordinate of that point is the second iterate. We repeat the steps to get the third iterate.

Now let's use the function cobweb to look at a few typical events. We begin by reducing $\mu$ to a value of 1/3, for which we have a single stable equilibrium. We then construct the cobweb plot of the approach to this equilibrium.

```
cobweb[0.23, 3, 0];
```

```
2 \mu \left( x + (1 - 2 x) \text{UnitStep}\left[ -\frac{1}{2} + x \right] \right), \quad \{\mu\} = \{ 1.00 \}
```

```
parmval = \{ 1 / 3 \};
```
This shows nicely the approach to the origin. Now let's increase $\mu$ and look at a periodic orbit.

```
parmval = {1};
```
In a way we cheated by starting on the orbit. Let's do this again, but this time we start off the orbit.
Now we see the transient approach to the orbit.

So far we have seen examples of equilibrium and of a periodic orbit. Is there any analogue here to the chaos that we saw with the Lorenz equations? The picture below answers that question. We choose a rational number for the initial x so that all calculations are done exactly.
This is a chaotic orbit for the tent map. This is the typical result for the typical initial condition when $\mu$ exceeds 1/2. For special choices of initial conditions we can land on one of the unstable periodic orbits.

### 5. Bifurcation Diagram

Often we want to know how system behavior depends on parameters. In the case of the tent map, we have a single parameter $\mu$, and we already know that there is a stable equilibrium for $\mu<1/2$. We also know that for $\mu>1/2$, there are two unstable equilibria, some unstable periodic solutions, and, as we have just seen, chaotic solutions. We can get an overview of how all of this depends on the parameter $\mu$ by a bifurcation diagram. That is a plot in which the abscissa is the value of $\mu$, and on the ordinate we plot all of the $x$-values from an orbit for that value of $\mu$. Fixed points will show up as a single point, a periodic orbit as several points, and a chaotic orbit as a band or several bands of points. The function which does this in DynPac is

$$\text{bimap}[\text{npts}, \text{ntthrow}, \text{nparm}, \text{xname}, \text{xrange}, \text{init}, \text{pname}, \text{range}, \text{ncomp}].$$
Here the number of values calculated for each parameter value is \( npts + n\text{throw} \), and the first \( n\text{throw} \) are discarded. The values are calculated beginning with \( \text{init} \). The number of parameter values for which this is done is \( n\text{parm} \). The range of parameter variation is \( \text{prange} \). The final argument \( n\text{comp} \) is an optional argument specifying the level of function composition. The default is 1. Now we apply this function to the tent map, choosing to keep 250 points and throw away the first 100 for each \( \mu \), starting with the value 1/2, and considering 200 \( \mu \) values in the \( \mu \) range \{1/2,3/4\}.

\[
\text{imsizes} = 420;
\]

\[
\text{asprat} = 0.7;
\]

\[
\text{bimap}[250, 100, 200, x, \{0.3, 0.8\}, \{1/2\}, \mu, \{1/2, 3/4\}];
\]

The dark area is the attractor for the chaotic motion. Next we do a closeup of the first part of this diagram.
We see from the closeup that there is fine structure in the attractor.