Maple Assignment 1
Due November 22

1. Maple is most useful when algebraic, rather than numerical answers, are needed. Given a periodic function $f(x)$ of period L, its Fourier series is defined to be

$$\frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos(\frac{n\pi x}{L}) + \sum_{n=1}^{\infty} B_n \sin(\frac{n\pi x}{L})$$

where $A_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos\frac{n\pi x}{L}dx$ and $B_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin\frac{n\pi x}{L}dx$ are called Fourier coefficients. Note that $A_0 = \frac{1}{L}\int_{-L}^{L} f(x)dx$.

   (a) Create a worksheet that, given $f(x)$, the period, L, will find find an expression for the Fourier coefficients for all n. Note that results with $\cos n\pi$ or $\sin n\pi$ are unacceptable, so make sure Maple replaces them automatically with more useful expressions.

   (b) Create an animation that, given $f(x)$, L and N, plots the first N Fourier sums (both sine and cosine terms) in succession.

   Good functions to test your work on are the triangle wave (extended periodically, of course):

$$f(x) = \begin{cases} x & \text{if } -1 \leq x < 1 \\ 2 - x & \text{if } 1 \leq x < 2 \\ -2 - x & \text{if } -2 \leq x < -1 \end{cases}$$

   and the square wave: $f(x) = \text{signum}\,(x)$ for $-1 \leq x < 1$ (again extended periodically).

2. The Cornu spiral is a parametric curve whose curvature changes linearly with respect to the parameter. It is given by

$$x(t) = \int_0^t \cos(s^2)ds$$
$$y(t) = \int_0^t \sin(s^2)ds.$$

   Note that the parameter is the upper bound of the integrals.

   (a) Create an animation of the Cornu spiral being traced out for $0 \leq t \leq 5$.

   (b) Create an "anonymous" procedure where "5" is replaced by user-specified parameter.

3. Given a function, f and a number x_0, the recursive sequence defined by f is the sequence

$$x_0,\ f(x_0),\ f(f(x_0),\ f(f(f(x_0)))),\ \cdots$$

These are usually hard to analyze. One way to visualize them is with cobweb diagrams. You create one by first drawing the graph of f and the line $y = x$. Starting at the point $(x_0, 1)$, you proceed vertically until you hit $y = f(x)$ (at the point $\big(x_0, f(x_0)\big)$). You then proceed horizontally until you reach $y = x$ at $\big(f(x_0), f(x_0)\big)$. Proceed vertically until you reach $y = f(x)$ at $\big(f(x_0), f(f(x_0))\big)$, etc. The sequence is given by the points on $y = x$, namely, (x_0, x_0), $\big(f(x_0), f(x_0)\big)$, $\big(f(f(x_0)), f(f(x_0))\big)$, ... If the sequence converges, the lines in the cobweb diagram will narrow in on the intersection point of $y = f(x)$ and $y = x$.

(a) Create a worksheet where the first lines define the function, initial point, and number of iterates, and the rest of the worksheet creates the cobweb diagram.

(b) Animate the diagram being traced out! The number of iterates at the top of the worksheet should determine the number of frames in the animation.

You can test your code on $f(x) = 2x(1 - x)$ and $x_0 = 0.1$. The diagram should spiral in on $(0.5, 0.5)$. For a more interesting picture, try $f(x) = 4x(1 - x)$ and $x_0 = 0.1$.

Maple Assignment # 1

Jenn Causey
Jarren Ralf

```
> restart; with(plots):
```

1.

Maple is most useful when algebraic, rather than numerical answers, are needed. Given a periodic function $f(x)$ of period L, its Fourier series is defined to be

$$\frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\,\pi\,x}{L}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{n\,\pi\,x}{L}\right)$$

where $A_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos\frac{n\,\pi\,x}{L}\,\mathrm{d}x$ and $B_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin\frac{n\,\pi\,x}{L}\,\mathrm{d}x$ are called the Fourier coefficients. Note that $A_0 = \frac{1}{L}\int_{-L}^{L} f(x)\,\mathrm{d}x$.

(a)

Create a worksheet that, given $f(x)$, the period, L, will find find an expression for the Fourier coeffcients for all n. Note that results with $\cos n\pi$ or $\sin n\pi$ are unacceptable, so make sure Maple replaces them automatically with more useful expressions.

```
> assume(n, integer); assume(N, integer); # N will be the
  number of terms in the sum
```

We will test our fourier series with a triangle wave, given a period of 2.

```
> f := piecewise( -1 < x and x < 1, x, 1 < x and x < 2, 2 - x,
  -2 < x and x < -1, -2 - x):
  L := 2:
```

Next, we will test our fourier series with a square wave, given a period of 1. Just uncomment the following two lines (and comment out the two above), then execute the worksheet again.

```
> #f := signum(x):
  #L := 1:
```

Establishing the fourier coefficients.

```
> A_0 := (1/L)*int(f, x = -L..L):
  A   := (1/L)*int(f*cos((n*Pi*x)/L), x = -L..L):
  B   := (1/L)*int(f*sin((n*Pi*x)/L), x = -L..L):
```

Setting up the finite fourier approximation.

```
> fourier := (x, N) -> A_0/2 + sum(A*cos((n*Pi*x)/L), n = 1..N)
  + sum(B*sin((n*Pi*x)/L), n = 1..N):
```

Taking the inifinite sum of the fourier series.

```
> fourier(x, infinity);
```

$$\sin\left(\frac{1}{2}\,\pi\,x\right) \quad \textbf{(1.1.1)}$$

(b)

Create an animation that, given $f(x)$, L, and N, plots the first N Fourier sums (both sine and cosine terms) in succession.

Choose the number of terms you want in your fourier approximation.

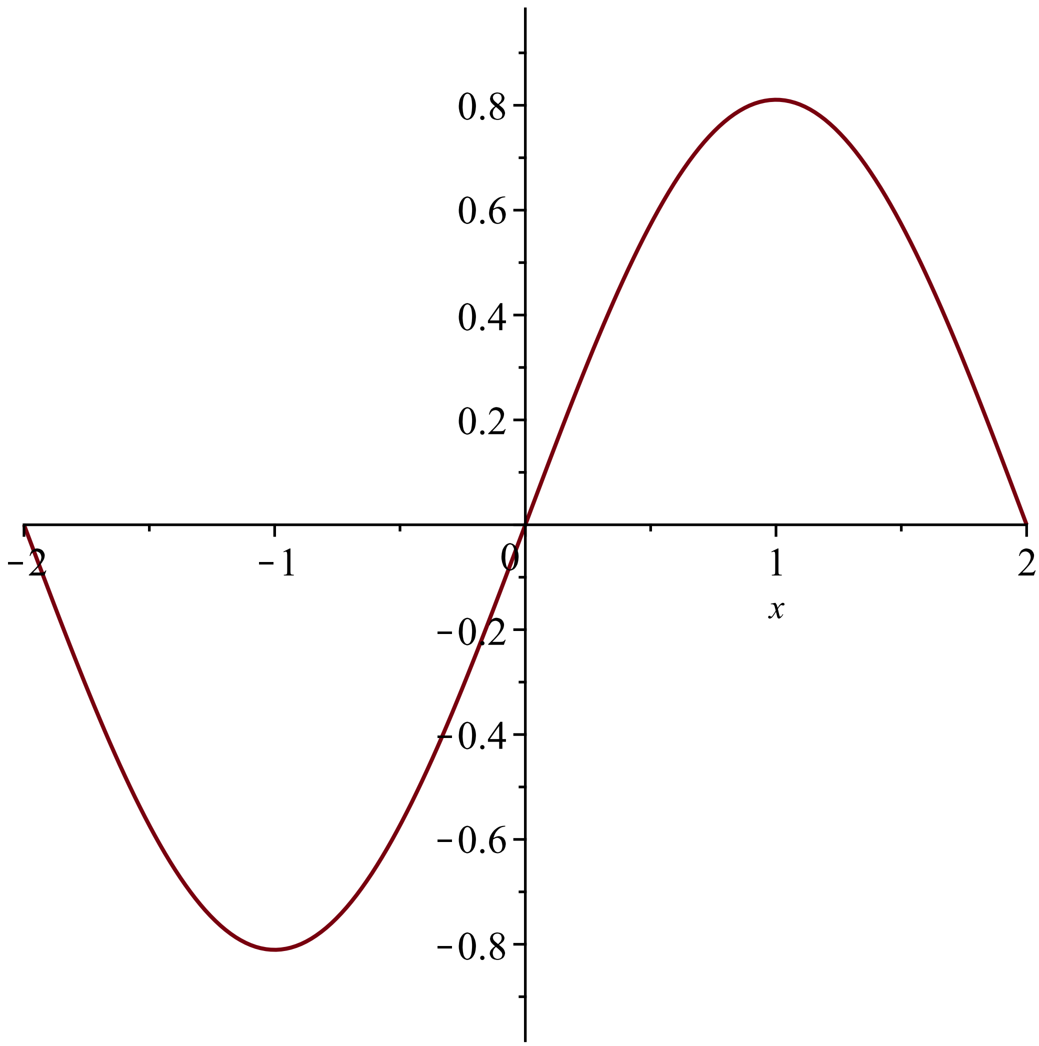
```
> N := 30:
```

Store the plots of each fourier approximation up to N terms.

```
> for i from 1 to N do
      fourierAnimation[i] := plot(fourier(x, i), x = -L..L):
  end do:
```

Display an animation for the first N terms of the fourier approximation.

```
> display([seq(fourierAnimation[i], i = 1..N)], insequence =
  true);
```



2.

The Cornu spiral is a parametric curve whose curvature changes linearly with respect to the parameter. It is given by

$$x(t) = \int_0^t \cos(s^2)\, ds$$

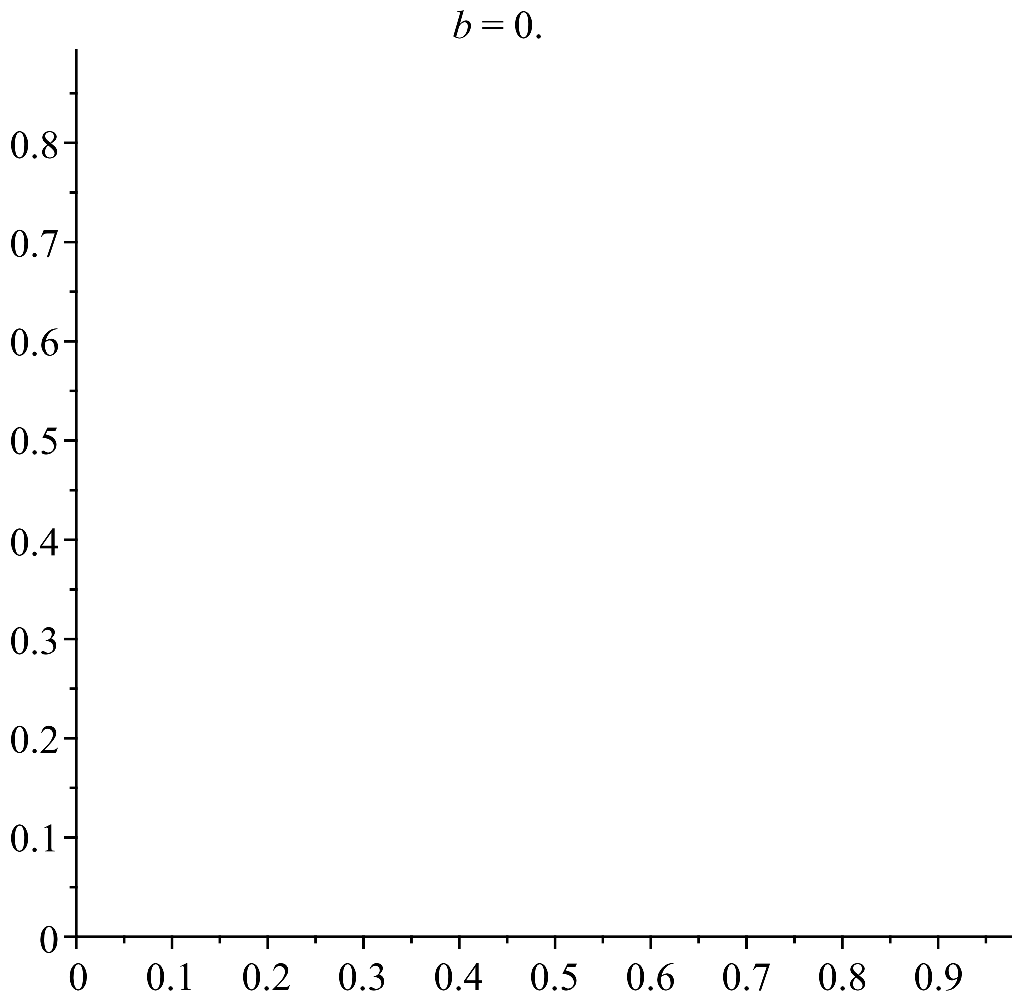
$$y(t) = \int_0^t \sin(s^2)\, \mathrm{d}s.$$

Note that the parameter is the upper bound of the integrals.

(a)

Create an animation of the Cornu spiral being traced out for $0 \le t \le 5$.

```
> xParametric := t -> int(cos(s^2), s = 0..t):
  yParametric := t -> int(sin(s^2), s = 0..t):
  animate(plot, [[xParametric(t), yParametric(t), t = 0..b]], b
  = 0..5,  frames = 50);
```



(b)

Create an "anonymous" procedure where "5" is replaced by user-specified parameter.

The following is the procedure to animate the plot of the Cornu for a user specified upper bound.

```
> #cornuSpiral := proc(upperBound)
  #    local xParametric, yParametric;
  #    xParametric := t -> int(cos(s^2), s = 0..t):
  #    yParametric := t -> int(sin(s^2), s = 0..t):
  #    animate(plot, [[xParametric(t), yParametric(t), t = 0..
  b]], b = 0..upperBound,  frames = 50);
```

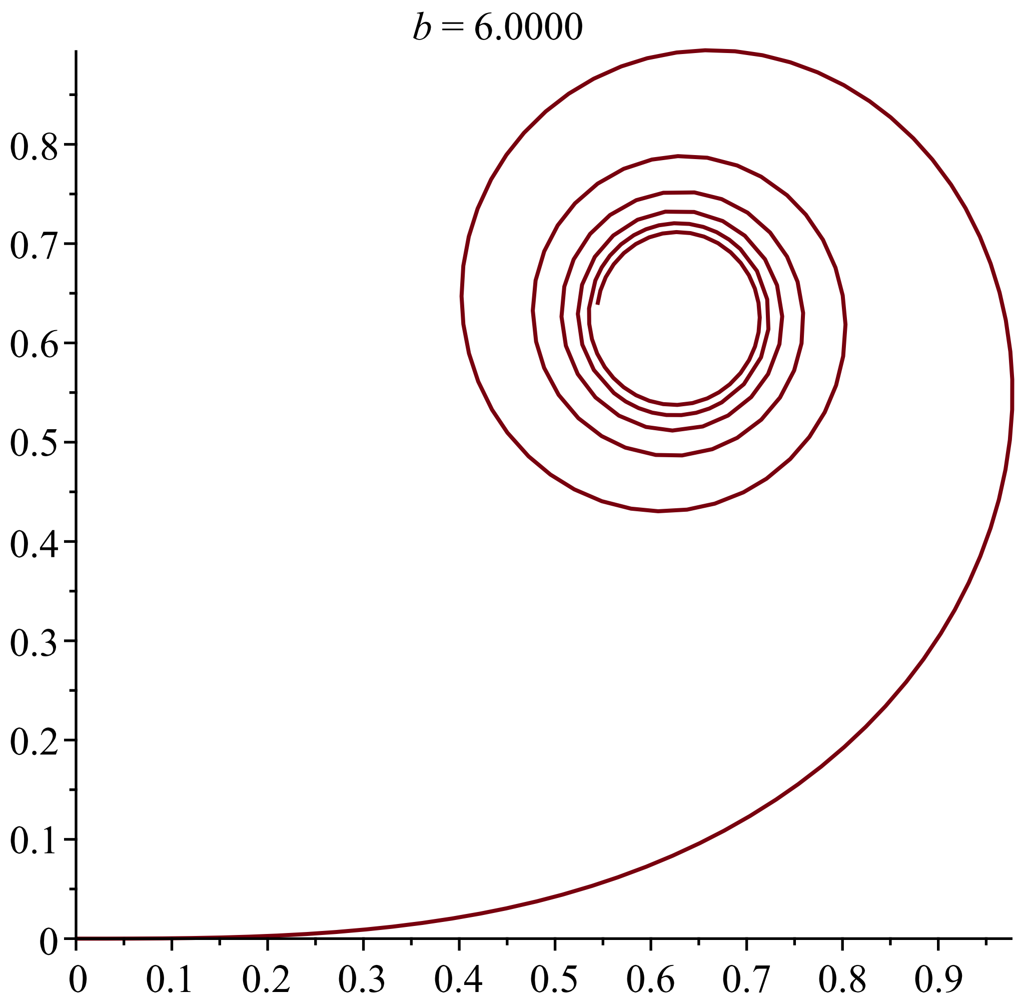
```
#end proc:
```

The "anonymous" procedure for animating the plot of the Cornu for a user specified upper bound.

```
> cornuSpiral := upperBound -> animate(plot, [[xParametric(t),
  yParametric(t), t = 0..b]], b = 0..upperBound,
                             frames = 50):
```

Call the function choosing your desired upperbound.

```
> cornuSpiral(6);
```



3.

Given a function, f, and a number x_0, the recursive sequence defined by f is the sequence

$$x_0,\ f(x_0),\ f(f(x_0)),\ f(f(f(x_0))),\ \ldots$$

These are usually hard to analyze. One way to visualize them is with cobweb diagrams. You create one by first drawing the graph of f and the line $y = x$. Starting at the point $(x_0, 0)$, you proceed vertically until you hit $y = f(x)$ (at the point $(x_o, f(x_0))$). You then proceed horizontally until you reach $y = x$ at $(f(x_0), f(x_0))$. Proceed vertically until you reach $y = f(x)$ at $(f(f(x_0)), f(f(x_0)))$, etc. The sequence is given by the points on $y = x$, namely, (x_0, x_0), $(f(x_0), f(x_0))$, $(f(f(x_0)), f(f(x_0)))$, ... If the sequence converges, the lines in the cobweb diagram will narrow in on the intersection point of $y=f(x)$ and $y=x$.

(a)

Create a worksheet where the first lines define the function, initial point, and number of iterates, and the rest of the worksheet creates the cobweb diagram.

Set your logistic constant.

```
> logisticConstant := 4:
```

Modify your function if needed.

```
> f := x -> logisticConstant*x*(1 - x):
```

Choose the x-value of your starting point.

```
> startingPoint := 0.1:
```

Choose the number of iterations you want.

```
> iterations := 50:
```

Here the procedure begins to create a cob web diagram and animation.

```
> cobWeb := matrix(iterations*2 + 1, 2): # Create a matrix of x
  and y coordinates
```

Establish the starting point.

```
> cobWeb[1,1] := startingPoint:
  cobWeb[1,2] := 0:
```

Store the points of the Cob Webs as well as the sequence of plot structures.

```
> for j from 1 to iterations do
      cobWeb[2*j, 1]      := cobWeb[2*j - 1,1]:
      cobWeb[2*j, 2]      := f(cobWeb[2*j, 1]):
      cobWeb[2*j + 1, 1]  := cobWeb[2*j, 2]:
      cobWeb[2*j + 1, 2]  := cobWeb[2*j, 2]:
      plotMatrix[2*j - 1] := plot([seq([cobWeb[k, 1], cobWeb[k,
  2]], k = 1..2*j - 1)]):
      plotMatrix[2*j]     := plot([seq([cobWeb[k, 1], cobWeb[k,
  2]], k = 1..2*j)]):
  end do:
```

Store the plot stuctures for the line $y=x$ and the chosen function.

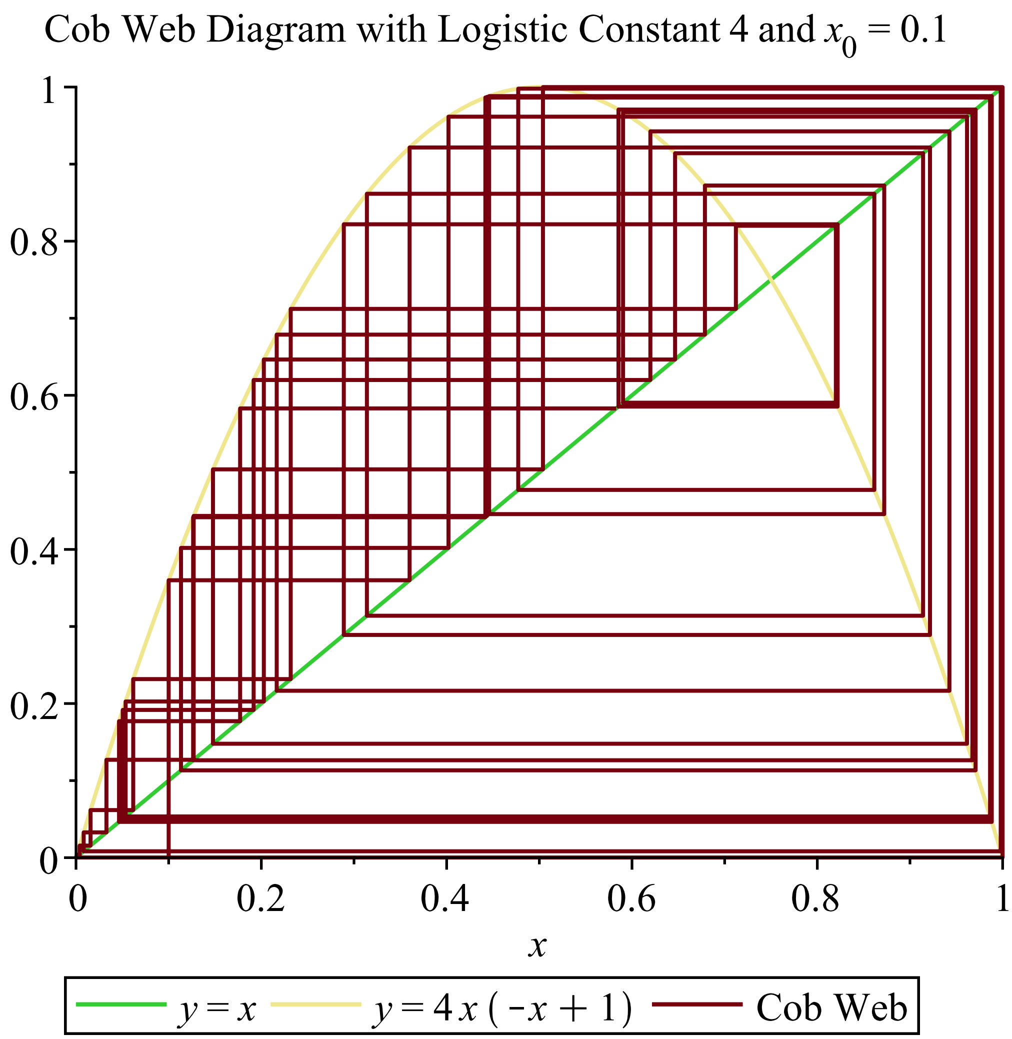
```
> identityFunction := plot(x, x = 0..1, color = "LimeGreen",
  legend = typeset(y = x)):
  chosenFunction   := plot(f(x), x = 0..1, color = "Khaki",
  legend = typeset(y = f(x))):
```

Store the plot stuctures for the cob web diagram and animation.

```
> cobWebDiagram    := plot([seq([cobWeb[k, 1], cobWeb[k, 2]], k
  = 1..2*iterations + 1)], legend = "Cob Web"):
  cobWebAnimation  := display([seq(plotMatrix[j], j = 1..2*
  iterations)], insequence = true):
```

Display the cob web diagram.

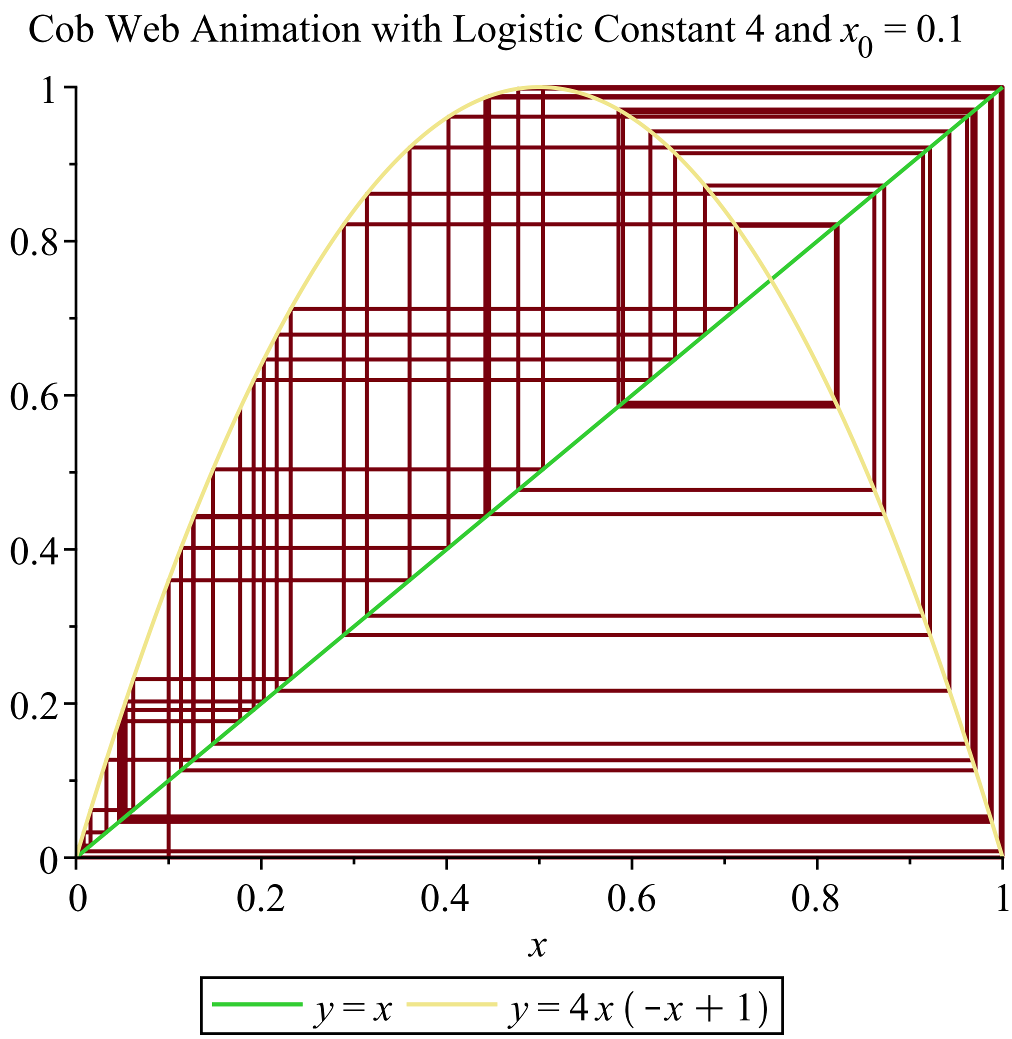
```
> display(identityFunction, chosenFunction, cobWebDiagram,
  title = typeset("Cob Web Diagram with Logistic Constant ",
  logisticConstant, " and ", x[0] = startingPoint));
```



(b)

Animate the diagram being traced out! The number of iterates at the top of the worksheet should determine the number of frames in the animation.

```
> display(identityFunction, chosenFunction, cobWebAnimation,
  title = typeset("Cob Web Animation with Logistic Constant ",
  logisticConstant, " and ", x[0] = startingPoint));
```



Cob Web Animation with Logistic Constant 4 and $x_0 = 0.1$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
x
$y = x$
$y = 4x(-x+1)$