Cobweb Diagrams With Excel

David Gurney
Department of Mathematics
Southeastern Louisiana University
SLU 10541, 500 Western Ave
Hammond, LA 70402
dgurney@selu.edu

October 27, 2004

Abstract

Iterative functions are often used to simulate the dynamics of
human and animal populations. Unfortunately, if the iterative
function is no more complicated than a two-term quadratic, trying
to predict the results of repeated application of the function is very
difficult. Even studying the numerical results of applying the
function again and again may not provide much insight into the
long-term behavior of the populations. Cobweb diagrams offer a
way around this dilemma. Originally devised for use with rulers
and graph paper, cobweb diagrams provide a relatively quick way
of representing the repeated application of an iterative function.
This talk will demonstrate how to create cobweb diagrams using a
Microsoft Excel spreadsheet. The strong and weak points of this
method of creating cobweb diagrams, as opposed to using the
TI-83 graphing calculator or software programs like Maple and
Mathematica, will also be discussed.

Every once in a while I am given the opportunity to teach a population modeling course at Southeastern Louisiana University. Usually the students who sign up for the course are those who need an extra couple of credit hours in something and their background is not much more than first semester calculus. Therefore I cannot assume the students know much about using Mathematica or Maple or that they even have a certain type of calculator. I usually show students how to use Excel to develop the simple models and also show them how to use Mathematica to help with the analysis of the resulting models.

This summer, however, I decided to restrict myself to using Excel for everything we needed to do. You can do quite a bit of population modeling with just Excel. Time series graphs are easy. Phase plane graphs are not much more difficult. Even putting in randomized birth and death rates is fairly simple. But until this year, I had not been able to construct a cobweb diagram in Excel in spite of some hours trying. I was lucky enough to discover the method described here, however, in the few weeks after spring finals and before my population modeling course began in the summer session.

Function Values for Iterated Logistic	
Iteration	Value
0	0.5000
1	0.8925
2	0.3425
3	0.8040
4	0.5627
5	0.8785
6	0.3811
7	0.8420
8	0.4749
9	0.8902
10	0.3488
11	0.8109
12	0.5474
13	0.8845
14	0.3648
15	0.8272
16	0.5102
17	0.8921
18	0.3436
19	0.8051
20	0.5601

Before hand-held calculators and personal computers, cobweb diagrams were a means to avoid some very messy calculations and still demonstrate the result of successive iteration of functions. Even with calculators and computers, the algebraic result of applying a function as simple as a binomial quadratic to itself over and over again can be very complicated after just a few iterations.

The example I will use for this paper is the function

$$f(x) = 3.57x(1-x)$$

with initial value 0.5. Population modelers call this the logistic function and 3.57 is the logistic constant. The table at right shows the numerical result of twenty iterations of this function. Figure A on the next page shows a plot of the values for each iteration. In either case, it's not very easy to tell what is happening in this process.

There are a few mathematical facts you can use to help predict the resulting behavior, but in the end the best method of seeing what happens is to create a cobweb diagram. I don't know who the person was who invented cobweb diagrams, but they really are a clever idea.

You start with a graph of your iteration function and the line $y = x$, and some initial value on the x-axis (see Figure B). By drawing a vertical line from the initial x-value to the function graph, the y-value of the intersection point is the result of applying the function to the initial x-value. That's the first iteration.

Next you draw a horizontal line from the intersection point on the function graph to the line $y = x$ (see Figure C). This has the effect of converting the first function value to your new x-value. Now you draw a vertical line from the intersection of the horizontal line and $y = x$ to the graph of the function again, and you have your second function value (see Figure D). Then you do it again. Draw a horizontal line from the function graph to the line $y = x$ and then draw a vertical line to the function graph to get your next function value (see Figures E and F). Keep repeating until you see a pattern or the segments fill your diagram.

Sometimes these paths end up converging to a point, sometimes they start cycling through a series of fixed points, sometimes they head off in one direction, and still other times they result in an unpredictable tangle. In any case, the result is called a cobweb diagram.

Figure A

Iterated Logistic

Figure B

Cobweb Building - Step 1

Figure C

Cobweb Building - Step 2

Figure D

Cobweb Building - Step 3

Figure E

Cobweb Building - Step 4
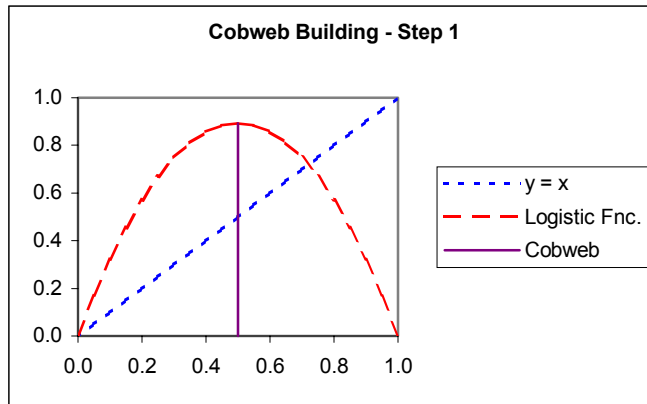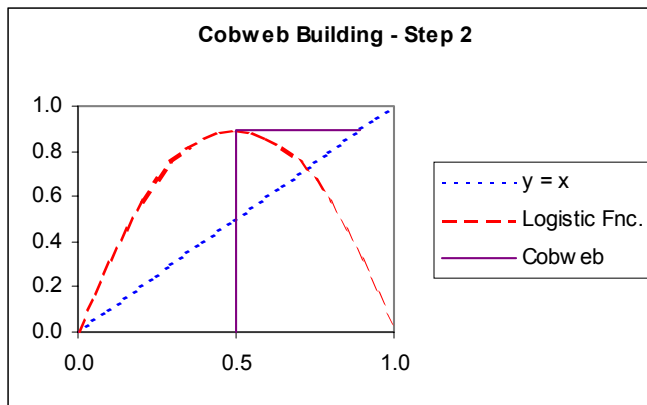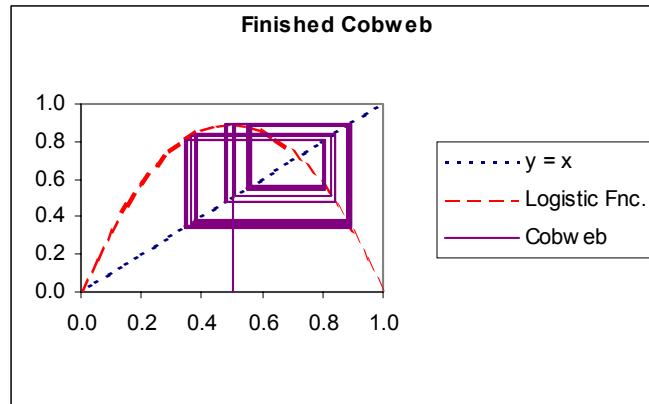
Figure F

Cobweb Building - Step 5

Figure G

Setting up the Excel Worksheet (see Figure H)

 Create columns corresponding to your x-values, the line $y = x$, and the values of the iteration function. In this example, I am using the logistic function as the iteration function.

 Create two more columns for the cobweb-x values and the cobweb-y values. Put the initial x-value in the first cobweb-x cell, and zero in the first cobweb-y cell. The second x-value will be the same as the first, but the next y-value will be the result of applying the iteration function to the initial x-value. The third x- and y-values are the same as the second y-value.

Figure H: Cobweb Setup

	A	B	C	D	E	F	G	H	I
1	3.57	**Logistic Constant**			0.500	**Initial Value**			**Formula in B5**
2									=A5
3	**Cobweb Frame**				**Iteration**	**Cobweb x**	**Cobweb y**		
4	**x**	**y = x**	**Logistic Fnc.**		0	0.5000	0.0000		**Formula in C5**
5	0.00	0.00	0.0000			0.5000	0.8925		=A$1*A5*(1-A5)
6	0.05	0.05	0.1696		1	0.8925	0.8925		
7	0.10	0.10	0.3213			0.8925	0.3425		**Formula in F5**
8	0.15	0.15	0.4552		2	0.3425	0.3425		=F4
9	0.20	0.20	0.5712			0.3425	0.8040		
10	0.25	0.25	0.6694		3	0.8040	0.8040		**Formula in F6**
11	0.30	0.30	0.7497			0.8040	0.5627		=G5
12	0.35	0.35	0.8122		4	0.5627	0.5627		
13	0.40	0.40	0.8568			0.5627	0.8785		**Formula in G5**
14	0.45	0.45	0.8836		5	0.8785	0.8785		=A$1*F4*(1-F4)
15	0.50	0.50	0.8925			0.8785	0.3811		
16	0.55	0.55	0.8836		6	0.3811	0.3811		**Formula in G6**
17	0.60	0.60	0.8568			0.3811	0.8420		=G5
18	0.65	0.65	0.8122		7	0.8420	0.8420		
19	0.70	0.70	0.7497			0.8420	0.4749		
20	0.75	0.75	0.6694		8	0.4749	0.4749		
21	0.80	0.80	0.5712			0.4749	0.8902		
22	0.85	0.85	0.4552		9	0.8902	0.8902		
23	0.90	0.90	0.3213			0.8902	0.3488		
24	0.95	0.95	0.1696		10	0.3488	0.3488		
25	1.00	1.00	0.0000			0.3488	0.8109		

At this point you can copy the formulas in the last two cells for *x* and the last two cells for *y* to as many of the following two-by-two blocks of cells as you like. Each of these two-by-two blocks of cells represents a step in the iteration process, and so each block you paste onto will produce another two strands of your cobweb.

Creating the Diagram

Highlight the columns with headings "x", "y = x", and "Logistic Fnc." Include the column headings in the highlighted area. Click on the chart wizard and you'll see the window for step 1 of the creation process (see Figure I). Choose "XY (Scatter)" and the chart image in the lower right for points connected by lines with no point markers. Then click "Next". You should see a graph of your iteration function and the line $y = x$ in the chart preview window (see Figure J).
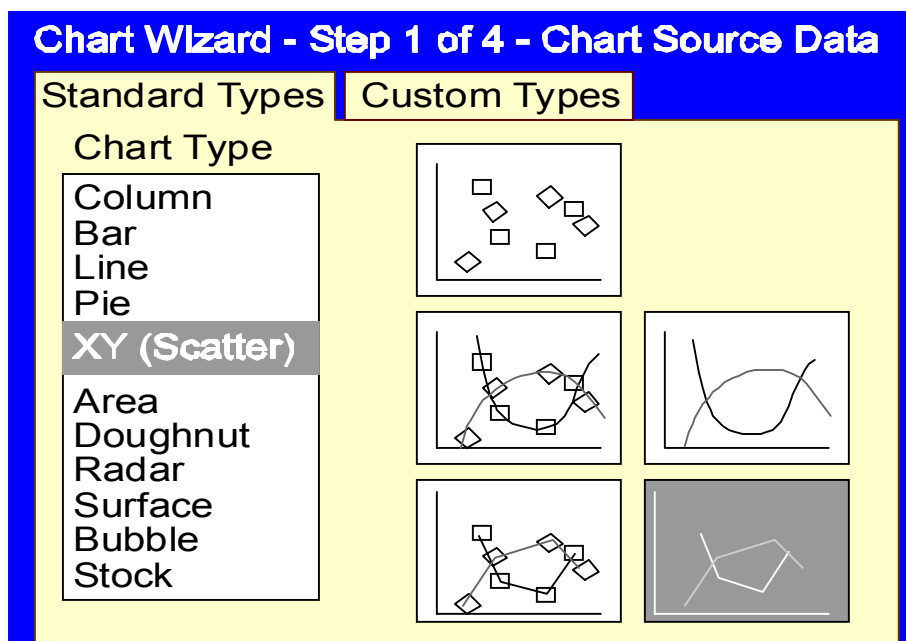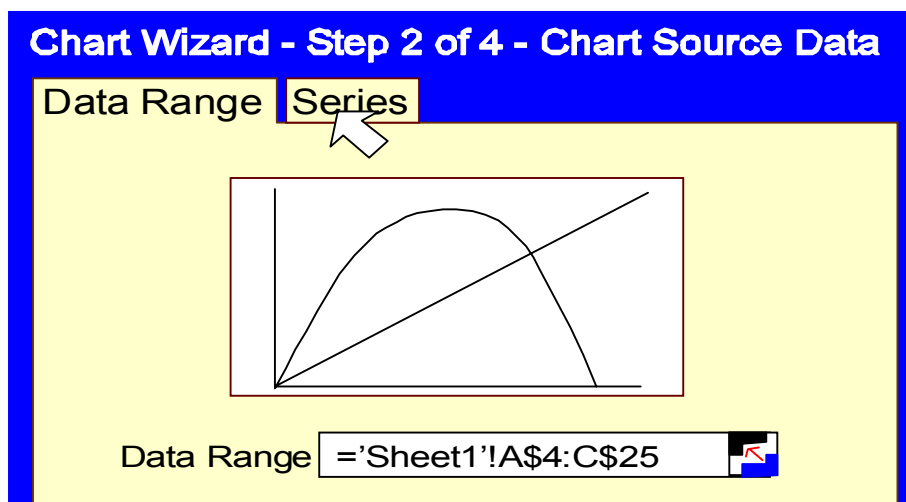
Figure I

Chart Wizard - Step 1 of 4 - Chart Source Data

Standard Types | Custom Types

Chart Type

Column
Bar
Line
Pie
XY (Scatter)
Area
Doughnut
Radar
Surface
Bubble
Stock

Figure J

Chart Wizard - Step 2 of 4 - Chart Source Data

Data Range | Series

Data Range ='Sheet1'!A$4:C$25

Click on the "Series" tab to open the "Source Data" window. Highlight the "Logistic" series in the "Series" box and click on the "Add" button (see Figure K). "Series 3" will appear. (see Figure L). In the name box, write "Cobweb".
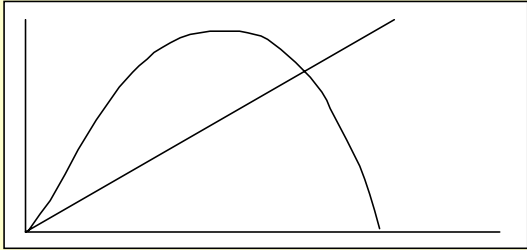
Next clear the "x-values" box and then click the cell-reference button at the right end of the box. You will be returned to your original spreadsheet and a "Source Data x-values" window will appear near the top. In the original spreadsheet, highlight the numbers in the column under "cobweb x" and click the cell-reference button in the "Source Data x-values" window to return to the "Source Data" window".

Now clear the "y-values" box and then click the cell reference button at the right end of the box. Again you are returned to your original spreadsheet, and this time with a "Source Values" window appears near the top. Highlight the numbers in the column under "cobweb y" and click the cell-reference button in the "Source Values" window to be returned to "Source Data" window. You should now see the cobweb in the chart preview window. Click "Next".

Figure K

Figure L

Elements of Style

The difficult part is now finished. The rest is just stylistic decisions. I usually add a fairly descriptive title, but then I call the *x*-axis X and the *y*-axis Y. I will also click on the "Gridlines" tab and remove the gridlines because there are usually so many lines in the finished product that it's hard to tell what's what otherwise. Click "Next" and you can tell Excel where to put the finished graph, in a current spreadsheet or in it's own sheet. Click "Next" again and the full-sized graph appears.

You can still edit the graph at this point. By clicking on any of the labels, you can change the font. The title font is usually set at 21.5 points and the other font is set to 18 point. To allow room for a bigger graph, I usually convert all the font to regular (non-bold) 12 point. By clicking on the *x*-axis or the *y*-axis, you can change the font and scale of the axis numbering. By clicking on the axis labels and the legend, you can change these fonts as well.

Finally, you can click on any of the curves you have graphed to change their color and style. I use colors like red, blue and violet for the graphs of "y = x", the iteration function and the cobweb, respectively. I also use dotted and dashed lines respectively for "y = x" and the iteration function so that one can still identify the different parts when the diagram is printed in black and white.

Figure M

Excel Logistic Cobweb: Constant = 3.57, Initial Value = 0.5
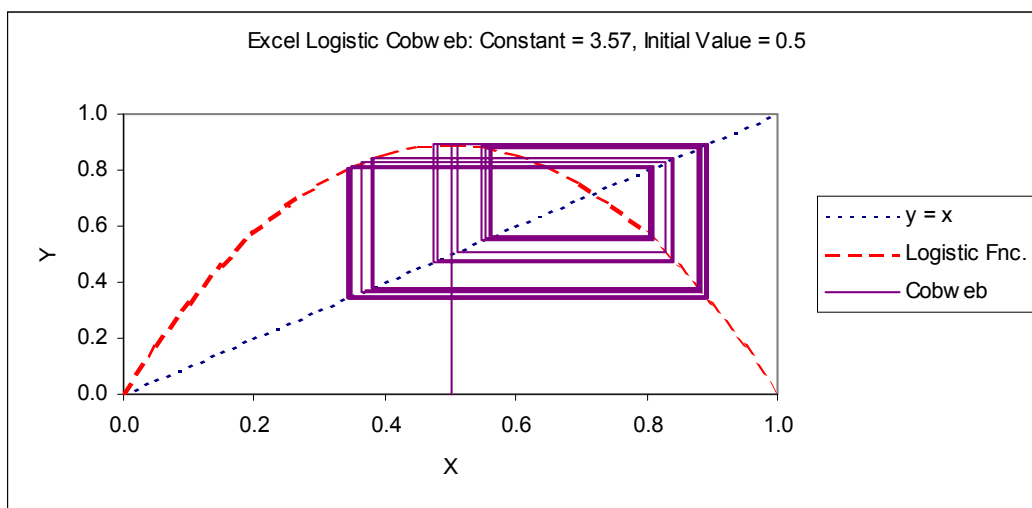
Creating Cobwebs on the TI-83 and TI-84

Press the $\boxed{\text{MODE}}$ button. In the row of options starting "Func", choose "Seq" and press $\boxed{\text{ENTER}}$. Press the $\boxed{\text{2nd}}$ button and then the $\boxed{\text{ZOOM}}$ button to select the "Format" option. Select the "Web" option in the first row and press $\boxed{\text{ENTER}}$.

Press $\boxed{\text{2nd}}$ and the $\boxed{\text{Y=}}$ buttons to reach the "Stat Plots" screen. Select the "PlotsOff" option and press $\boxed{\text{ENTER}}$ twice. Press the $\boxed{\text{Y=}}$ button and clear all functions.

Next to "u(n) =" type the following: K*u(n –1)*(1–u(n –1))

Use the $\boxed{\text{ALPHA}}$ button and the $\boxed{\text{(}}$ button to type the K. Use the $\boxed{\text{2nd}}$ button and the $\boxed{7}$ button to type the u's. Use the $\boxed{\text{X,T,θ,n}}$ button to type the n's. *Do not use spaces in this formula.*

Next to "u(nMin)=" type your initial value, 0.5. Then press $\boxed{\text{2nd}}$ and the $\boxed{\text{MODE}}$ button to quit this window. To set the logistic constant type 3.57 and then press $\boxed{\text{STO} \rightarrow}$, press $\boxed{\text{ALPHA}}$ and press $\boxed{\text{(}}$ to get $3.57 \rightarrow K$. Next press $\boxed{\text{ENTER}}$ to store the value 3.57 in the constant K .

Now press the $\boxed{\text{WINDOW}}$ button and set the window variables as shown below.

nMin = 0	nMax =10	PlotStart =1	PlotStep = 1	Xmin = 0
Xmax =1	Xscl =1	Ymin= -0.1	Ymax=1.1	Yscl=1

Next press $\boxed{\text{TRACE}}$ to show the cobweb framework. Finally, by pressing the $\boxed{\triangleright}$ button repeatedly, the cobweb is constructed segment by segment.

Figure N: Cobweb Diagram Created by TI-83

```
u=K*u(n-1)*(1-u(n-1))




n=50
X=.88737178  Y=.35479801
```
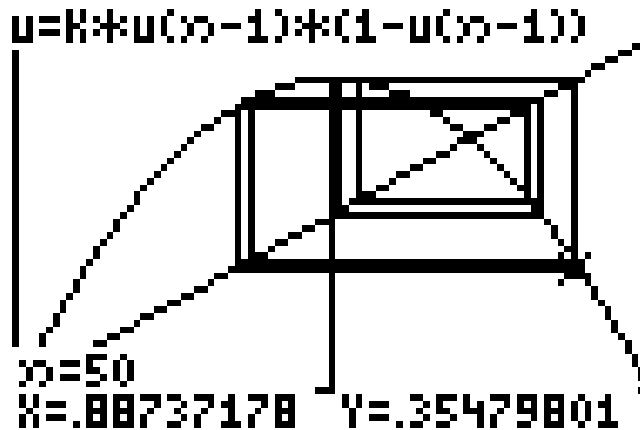
The Texas Instruments approach allows students to make cobweb diagrams with just their graphing calculators. Making a hard copy is a little more difficult, but −at least at my school − there are a few people around who have the software needed to print copies of the screen output from Texas Instrument calculators. The biggest drawback from my perspective is the lack of labeling on the axes. The procedure is given in the TI-83 handbook. To follow this procedure, you really need to know some basics about operating the calculator, including things like storing values in a constant and how to type in "u" for the iteration function, and this may require some time studying the rest of the manual. Once you know these basics, the procedure is easy to follow.

Mathematica Program for Creating Cobweb Diagram

```
(* Clear out the old function values *)
Clear[f]
Clear[g]
(* Set the initial coordinates *)
x0=0.5;y0=0;
(* Define the logistic function *)
f[x_]:=3.55x-3.55x^2
(* Define the segment builder function *)
g[{{x_,y_},{z_,w_}}]:={{z,w},{f[x],f[z]}}
(* Define the cobweb building process *)
CW=NestList[g,{{x0,y0},{x0,f[x0]}},100];
(* Create the iteration function plot *)
IF=Plot[f[x],{x,0,1},PlotStyle->{RGBColor[0,0,1],Dashing[{0.04,0.01}]},
        DisplayFunction->Identity];
```

(* Create the plot of y = x . *)

ID=Plot[x,{x,0,1},

> **PlotStyle->{RGBColor[1,0,0],Dashing[{0.01,0.01,0.04,0.01}]},**

> **DisplayFunction ->Identity];**

(* Plot the iteration and y = x together with the cobweb. *)

Show[{IF,ID,Graphics[{RGBColor[0,1,0],Map[Line,CW]}]},

> **DisplayFunction->$DisplayFunction,**

> **PlotLabel->"Mathematica Logistic Cobweb:**

> **Constant 3.55, Initial Value 0.5"]**

Figure O: Mathematica Cobweb Diagram
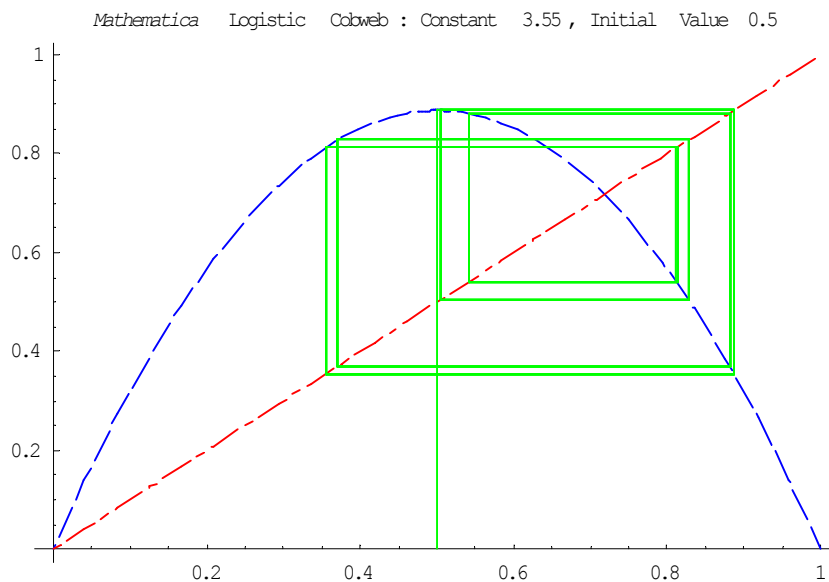
This Mathemtica program creates a nice diagram with labeled axes and a title. The amount of code involved is not to bad. The biggest problem would be explaining the cobweb building section of the program to someone unfamiliar with Mathematica. I spent some time studying Wolfram's *The Mathematica Book* (1996) in an effort to write such a program, but I eventually asked Tilak deAlwis of Southeastern Louisiana University for help and he wrote the original version. The current version has comments added and uses different variable names.

Maple Program for Creating Cobweb Diagram

A program to plot cobweb diagrams.

> with(plots):

Set the number of iterations.

> Steps:=30:

Set the logistic constant.

```
> L:=3.55:
```

Define the logistic iteration function.

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

Set the initial value.

```
> IV:=0.5:
```

Define the matrix for holding cobweb points.

```
> CW:=matrix(Steps*2+1,2):
```

Set the starting point of the cobweb.

```
> CW[1,1]:=IV:
> CW[1,2]:=0:
```

Give the procedure for filling in other cobweb points.

```
> for j from 1 to Steps do

          CW[2*j,1]:=CW[2*j-1,1]:
          CW[2*j,2]:=f(CW[2*j,1]):
          CW[2*j+1,1]:=CW[2*j,2]:
          CW[2*j+1,2]:=CW[2*j,2]:

   end do:
```

Create the iteration function plot.

```
> ItFnc:=plot(f(x),x=0..1,color=navy,linestyle=2):
```

Create the identity function plot.

```
> Id:=plot(x,x=0..1,color=green,linestyle=3):
```

Create the cobweb diagram plot.

```
> CbWb:=[seq([CW[n,1],CW[n,2]],n=1..2*Steps+1)]:
> CWP:=plot(CbWb):
```

Create a title for the plot.

```
> Ttl:=textplot([.52,.98,`Maple Logistic Cobweb:
  Constant 3.55, Initial Value 0.5`]):
```

Put all three plots on the same graph and add the title.

```
> display(ItFnc,Id,CWP,Ttl);
```

Again the graph using Maple (shown in figure P below) is clear with labeled axes and a title. There is definitely more coding involved, but the program steps are fairly easy to follow for students who know a standard programming language. I wrote the original version in less than two hours using Garvan's *The Maple Book* (2002) as my only reference**.**

Figure P: Maple Cobweb Diagram

Maple Logistic Cobweb: Constant 3.55, Initial Value 0.5
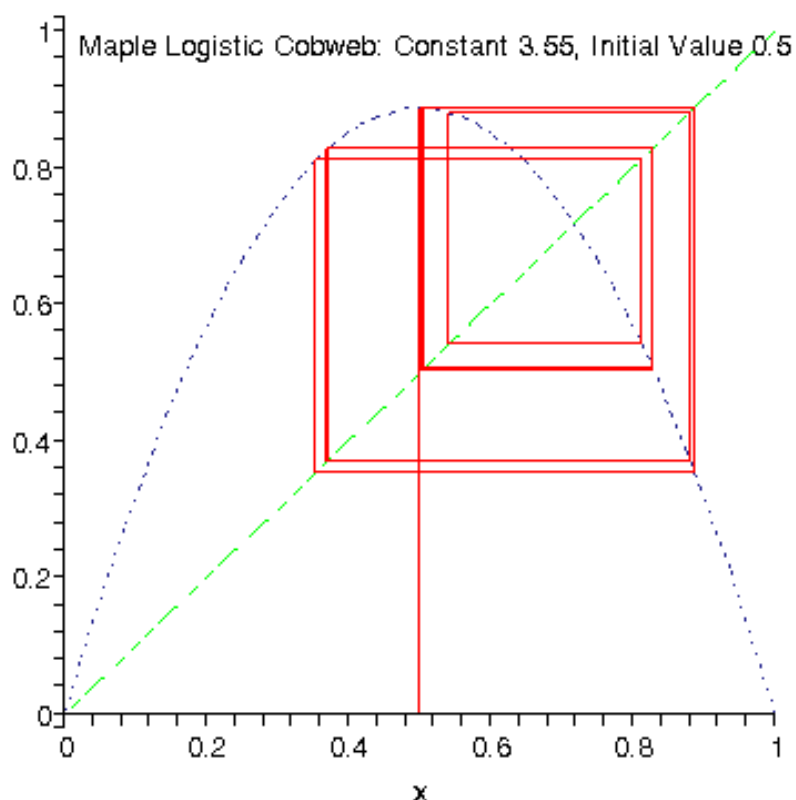
Having tried all these methods, I really prefer the Excel approach. One reason is that the set-up is fairly simple and students need just a basic knowledge of Excel to understand how the procedure works. Identifying and explaining the important parts of the Excel spreadsheet is much easier than trying to explain the programming steps in Maple or Mathematica. Since the procedure is built into the TI-83 (+) and TI-84 (+), there is not much that can be explained in that approach. Another reason to prefer the Exel approach is that when initial conditions or constants are changed in the spreadsheet, the change appears almost instantly in the cobweb diagram. Changing initial conditions or constants with the other approaches requires running through the procedures again to recreate the diagrams.

References
Frank Garvan, *The Maple Book*, Chapman & Hall/CRC, Boca Raton, FL 2002
Daniel Kaplan & Leon Glass, *Understanding Nonlinear Dynamics*, Springer-Verlag, New York
        1995
TI-83 Plus Graphing Calculator Guidebook, Texas Instruments Inc. 1999
Stephen Wolfram, *The Mathematica Book*, 3rd Edition, Wolfram Media/Cambridge University
        Press, Champaign, IL 1996