Example of Nonlinear Dynamics

Andrea De Paoli

Begin with a calculator (usually scientific) that has $\ln(x)$ and x^y function, and consider the function $f(x) = \ln(x^y)$.

Try a few examples, and use the answer from each operation as an input for the next one.

There are different results depending on the y used, and on the initial x used (for each y).

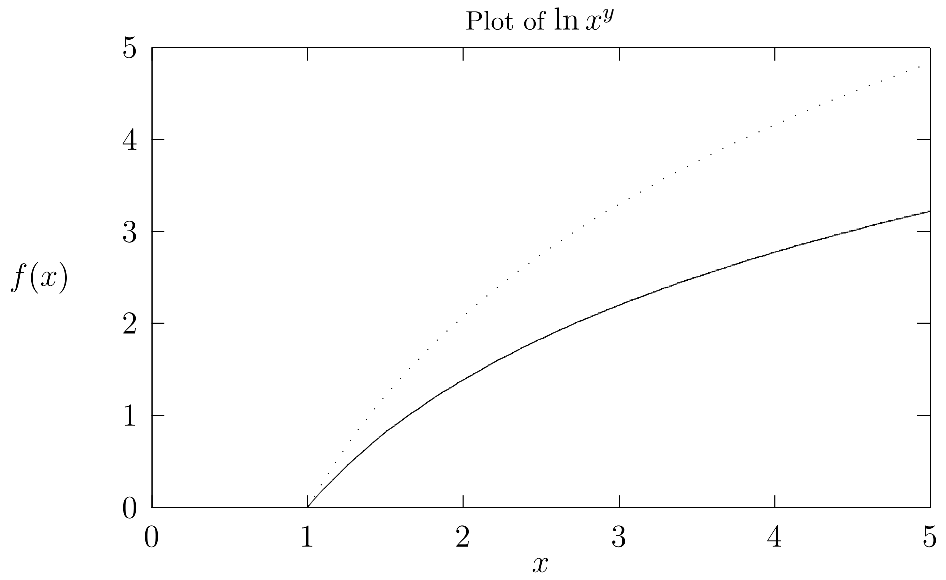
With a PC we can use a simple algorithm to keep track of the numerical results, and notice if there are any patterns.

Algorithm

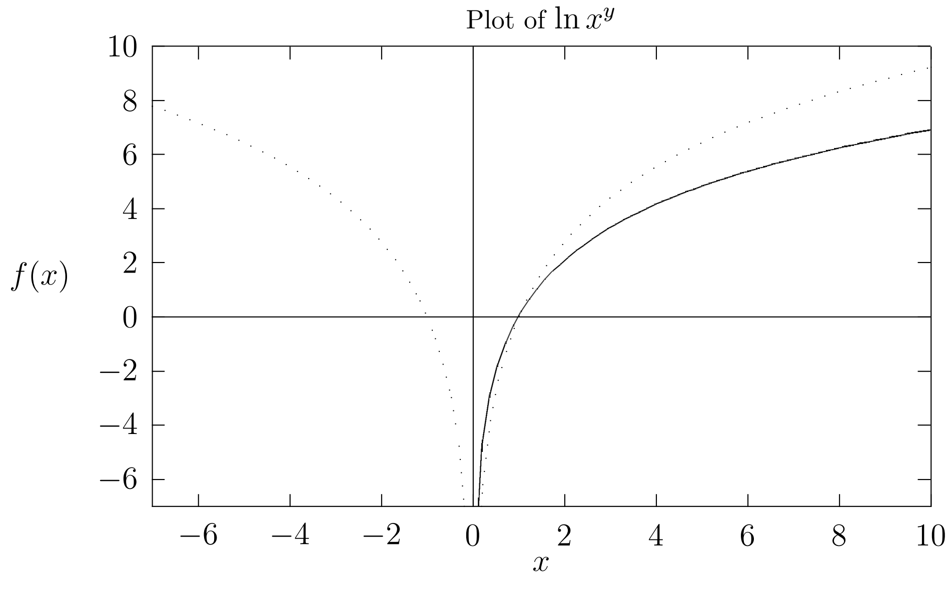
```
BEGIN LOOP from 1 to N
  x = power(x,y)
  x = log(x)
END LOOP
```

Note: You may want to look at the appendix for an example of the code in C.

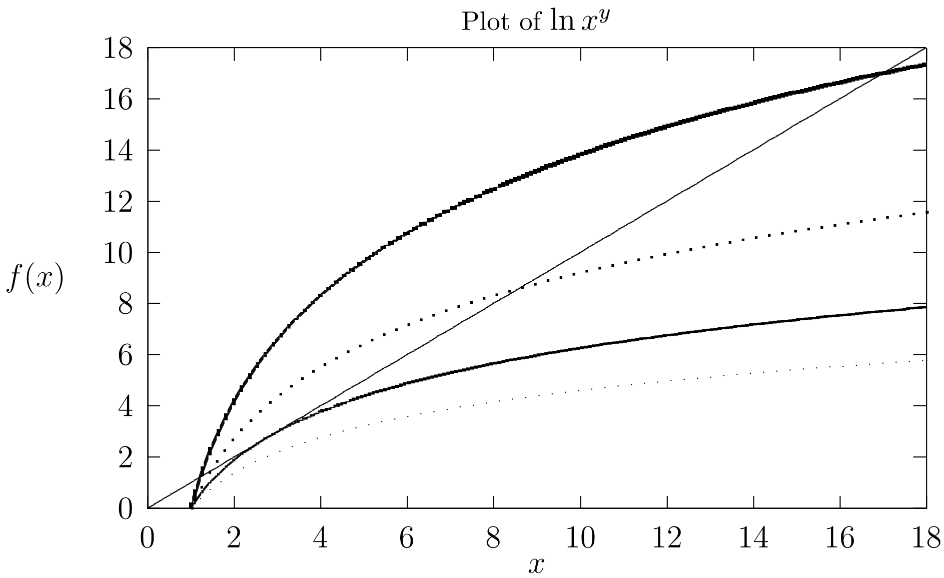
After you have seen a number of patterns in the values it is instructive to draw a graph from some of the inital values of x and y. But first let us take a look at the function $\ln(x^y)$ first for a few positive values of y. For a value of y and for $x > 1$ it will look something like this:



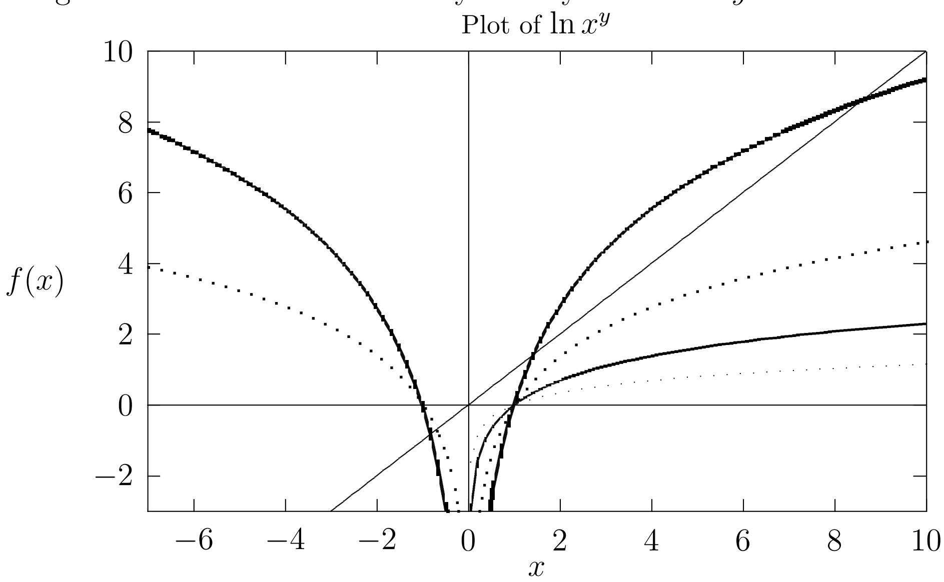
If we include negative values of x we find that it behaves differently depending on whether y is even or odd obviously, though the curves are similar in the positive domain.



If we now include various values of y and compare with the line $y = x$ we see that for $f(x) = \ln(x^y)$ with $y < e$ the curve does not touh the line, it touches at one point for $y = e$ and the curve $f(x)$ crosses the line twice for values of $y > e$



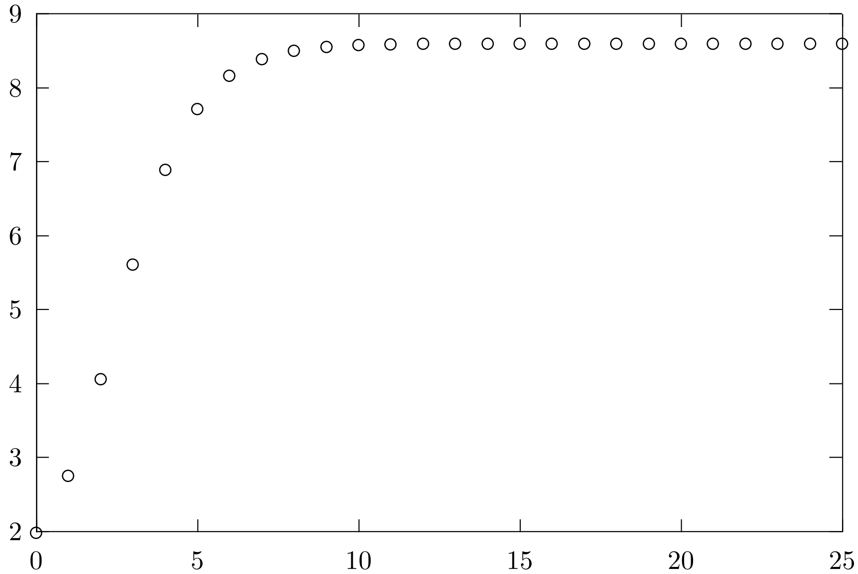
If we again "zoom back", we see that the even curves also cross the line at a negative value of x due to the symmetry about the y axis.



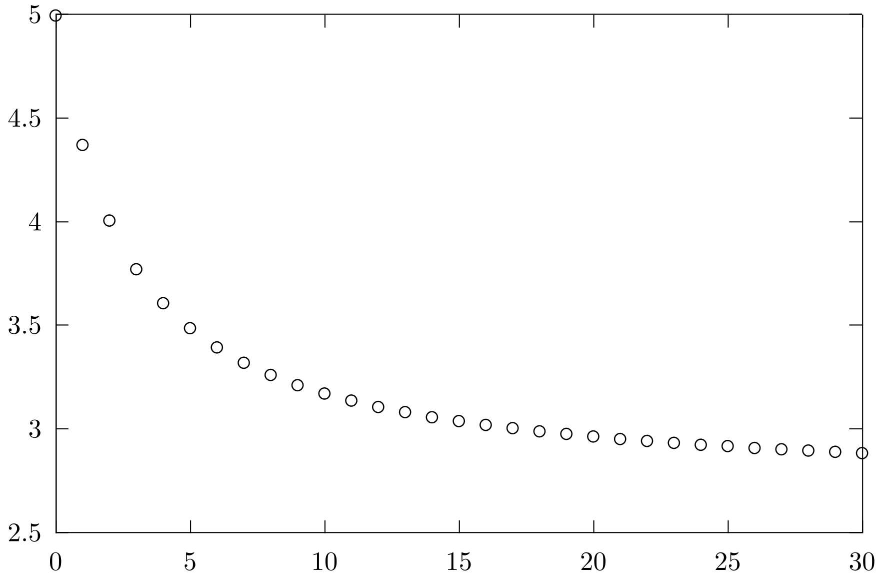
It does not take long to realize that the initial value $x = 1$ (as well as $x = -1$) for $y = 2 \cdot n$ leads to error after a finite number of iterations (it "catches up" with the value of $x = 0$ which is undefined). For $y = e$ (i.e. $f(x) = \ln(x^y) = \ln(x^e)$) there is one fixed value: $x = e$, and for all values of $y > e$ there are two fixed values, the lower one is unstable, and tends to 1 for values of $y \to \infty$, while the higher one is a stable fixed point which tends to infinitity as $y \to \infty$, and all inital values tend to the same fixed value for a given (fixed) y.

For $y < e$ we get a very interesting pattern. Let us look at the iterations for various values of y (and initial x)

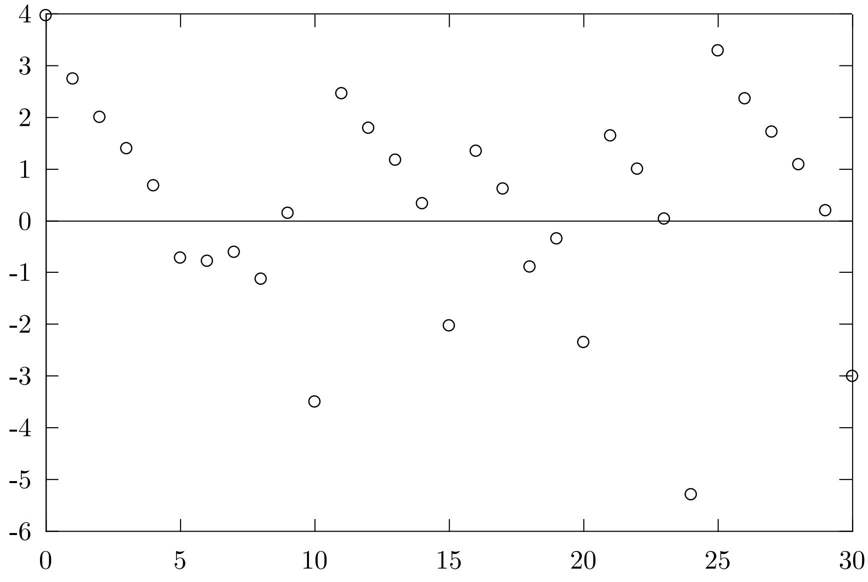
This first example $\langle \ln(x^4) \rangle$ with initial value $x = 2$. This is a typical solution for $x_{t+1} = \ln(x_t^y)$ with even values of y. For negative initial values, the solution set "catches up" with the values of the repsective positive value.



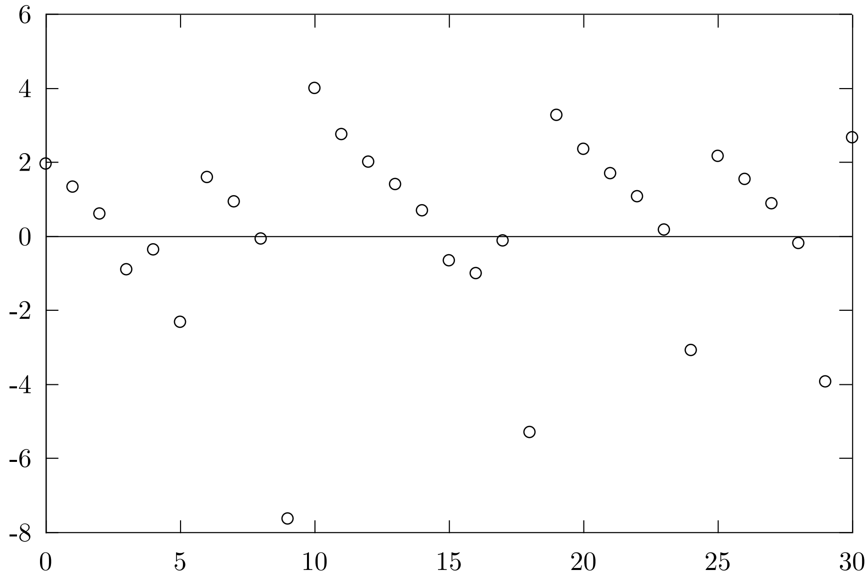
Below is an example $\langle \ln(x^e) \rangle$, using 5 as an initial value. Trying different initial values for $x > e$ you will note that they all have a similar structure, the stable fixed point is always e, and, in fact, if the initial value is $x = e$ we then have a constant value.



In the next two examples we see what may in fact be chaotic behavior, both are different inital values of x with $y = 2$. A negative initial value will equal the solutions of the positive one after a a finite set of iterations. So we can (ignoring the finite set) consider only the positive initial values. In this example the inital value is 4.

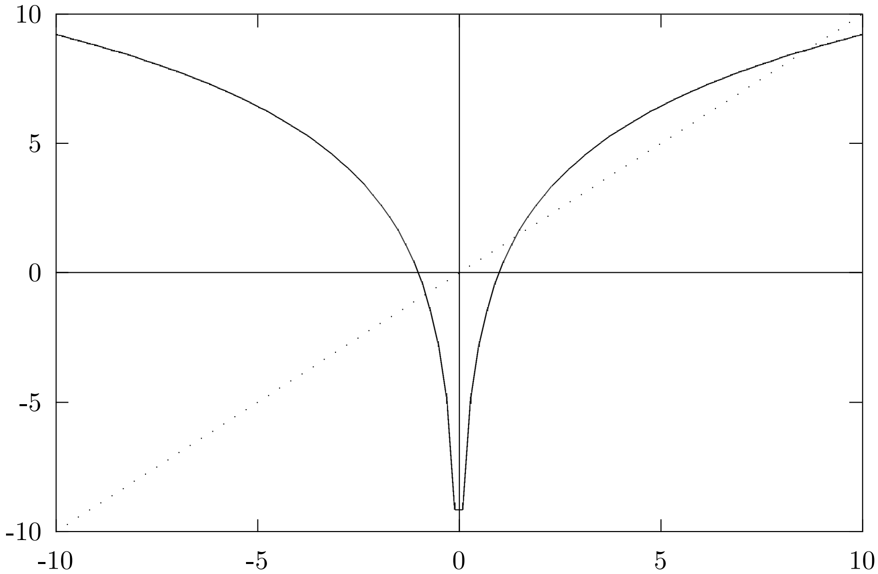


This is the same as the above graph, except that the initial value is $x = 2$.

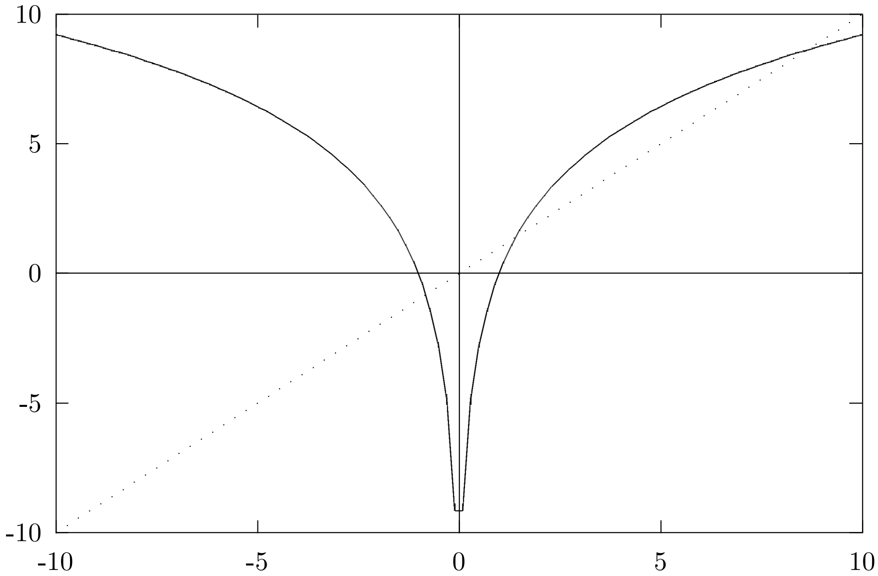


And now let's see what the results of the cobweb method gives us, starting from several values.

Starting with initial value $a_1 < x < a_2$, (where $f(a_1)$ is the lower (unstable) positive fixed value, and $f(a_2)$ is the stable fixed value.



Starting with initial value $x < a_1$, (where $f(a_1)$ is the lower (unstable) positive fixed value) we have a few oscillations before reacing a convergence towards the stable fixed value.

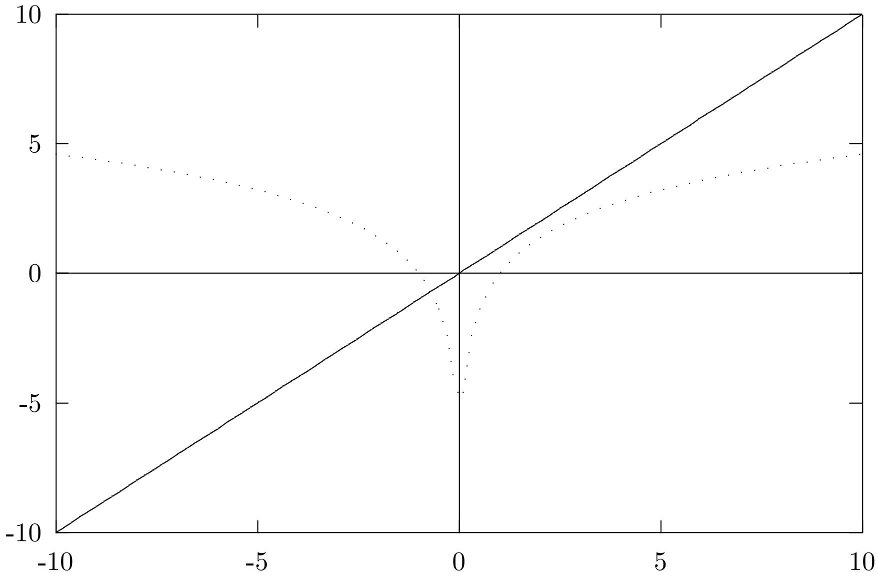


Note that for different values of y, we have as stable fixed values, for $x = 4, 6, 8$ and 10 respectively:

8.61316945644
16.9988873523
26.093485
35.771521

For $y = e$ we can actually calculate it rather quickly, $x = \ln(x^e) \Rightarrow x = e\ln(x)$ whose solution is obviously $x = e$, since $e = e \cdot \ln(e)$.

Below we shoe the cobweb procedure for $\ln(x^2)$.


10
5
0
-5
-10
-10
-5
0
5
10

In C it would look something like this (this uses hardcoded values of y and then loops for a hardcoded number of operations, and writes the results into an ASCII file.

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>

FILE    *f_out ;
long    i;
double  x;
char    rtf_line  [255];
char    f_out_name [60] ;

main ( )
{ sprintf ( f_out_name,"%.4s%.4s","RSLT",".DAT");
  /* ----- this is just error trapping for the file --*/
  if ( ( f_out = fopen ( f_out_name , "w" ) ) == NULL )
  {
    printf ("\n");
    printf ( " >>> Error: write %s FAILED.\n", f_out_name );
    printf ("\n");
    exit(0);
  }
  /* ----- this is a header string ----- */
  strcpy( rtf_line, " list of results of f(x)= ln(x^n) \n");
  fputs ( rtf_line , f_out );
 /* ---- this is the initial value --- */
 x = 5.8;
 /* ---- prints a line and initial value to file --- */
 sprintf(rtf_line, "\n");
 fputs ( rtf_line , f_out );
 sprintf(rtf_line, " %f \n", x);
 fputs ( rtf_line , f_out );
 /* --- loops 100 times, reiterating operation ---- */
 /* --- and creates a line for each value output -- */
```

```
 for(i=1;i<=100;i++)
 {
   x = pow(x,4);
   x = log(x);
   sprintf(rtf_line, " %f \n",x);
   fputs ( rtf_line , f_out );
 }
 /* --- just closes the file and sneds message to screen -- */
  fclose ( f_out );
  clrscr();
  printf ( "\n Data in file: %.12s ",f_out_name);
  return 0 ;
}
```