

Chaos Musings



By Walid Maalouli

October 2010

For the TI 99/4A computer

Introduction

Ever since Edward Lorenz started dabbling in climate modeling in the 1960's and stumbled upon the apparent anarchy of weather, the field of chaos has been growing by leaps and bounds. But with the advent of computers, it became possible to take endless strings of numerical results and turn them into intriguing and frequently fascinating images.

This collection of three programs will attempt to demonstrate the capabilities of the TI 99/4A computer at modeling chaos. Be forewarned however that on a stock machine, these explorations will frequently require hours and even days of calculations before a final image is produced on the screen. While I have run all of them on a real TI, I have found that using an emulated environment on a fast modern computer works better from a time perspective.

System Requirements

All the programs were written in 100% assembly language in order to maximize performance as well as access the high resolution bitmap screen. Therefore, you will need the following:

- Editor/Assembler cartridge
- 32K RAM expansion
- One disk drive
- Lots of patience

Use

There are no restrictions on the use of these programs. Feel free to play with them and modify them at will. I would appreciate it however if you could send me any enhancements you make.

Enjoy and may the chaos be with you...

Walid Maalouli
wmaalouli@comcast.net

The Logistic Equation

How to run:

1. Choose Option #3 LOAD AND RUN from the E/A menu
2. At the FILE NAME? prompt, enter DSKx.LOGISTIC (DSKx can be any storage device you want)
3. At the PROGRAM NAME? prompt, enter START

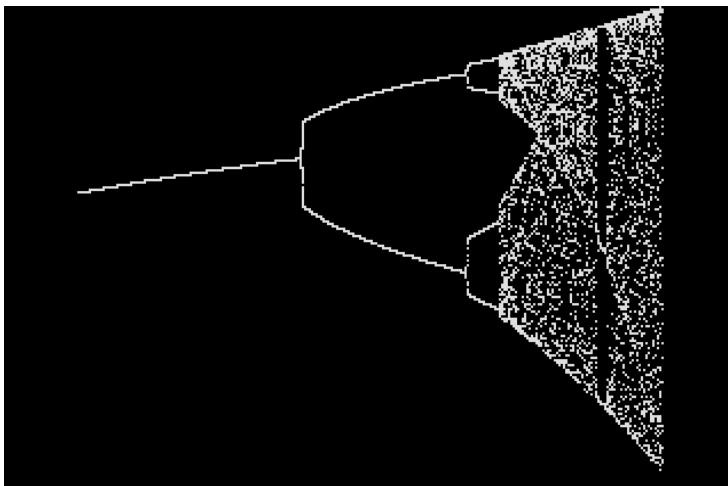
How to use:

The logistic equation represents a very simplified model of population growth and is written as

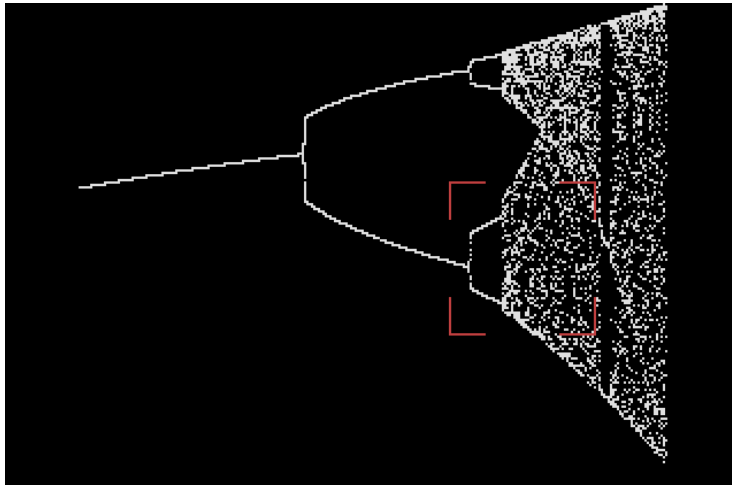
$$P = PR(1-P)$$

where P represents population number and R is the fecundity coefficient which varies from 0 to 4. This relatively benign looking equation start off simply enough and seems very linear, but then starts exhibiting an increasing number of bifurcations with 2 or more stable values for P as R increases, until a threshold is reached where chaos suddenly appears and it becomes impossible to predict where P will fall. Furthermore, the resulting image is actually a fractal, where increasing magnifications will reveal mirror images of the whole.

It takes about 6 minutes for a stock TI to draw the first image wirh R on the x-axis and P on the y-axis.



If you would like to magnify a section of the image, press **W** and a red window will appear which you can move around using the arrow key (**E/S/D/X**).



Once you are satisfied with the location of the window, press **ENTER** and the computer will start the calculations again, displaying its progress until it reaches the area of interest, at which point that area is displayed. You can continue to zoom in in that manner as much as you want, but please keep in mind that *each zoom level increases the calculation time by a factor of 10*. Therefore, on zoom level 1, it will take the computer 60 minutes to calculate and draw the image, 10 hours of zoom level 2, and so on and so forth.

The computer will store in memory the first 3 images produced, namely zoom levels 0, 1 and 2, and you can page back and forth between them by pressing the **B** and **F** keys respectively and you can select different zoom areas in each if desired, which allows for a great deal of exploration. *If you go beyond zoom level 3 however, you will not be able to go back and view the earlier images.*

The Hénon Equations

How to run:

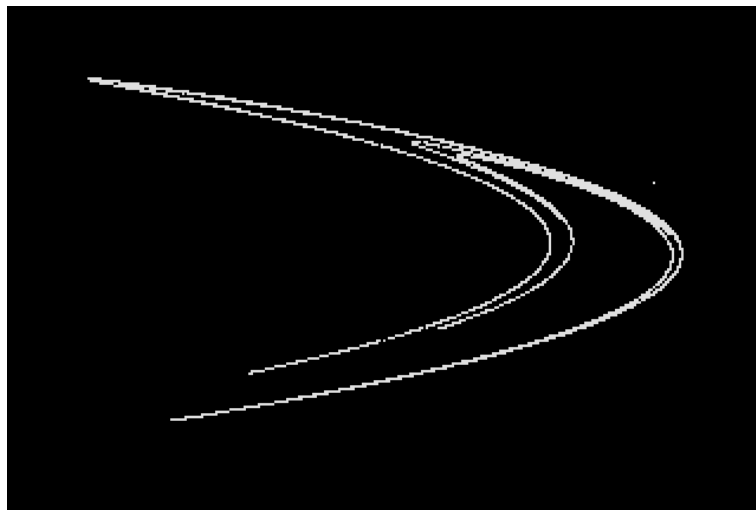
4. Choose Option #3 LOAD AND RUN from the E/A menu
5. At the FILE NAME? prompt, enter DSKx.HENON (DSKx can be any storage device you want)
6. At the PROGRAM NAME? prompt, enter START

How to use:

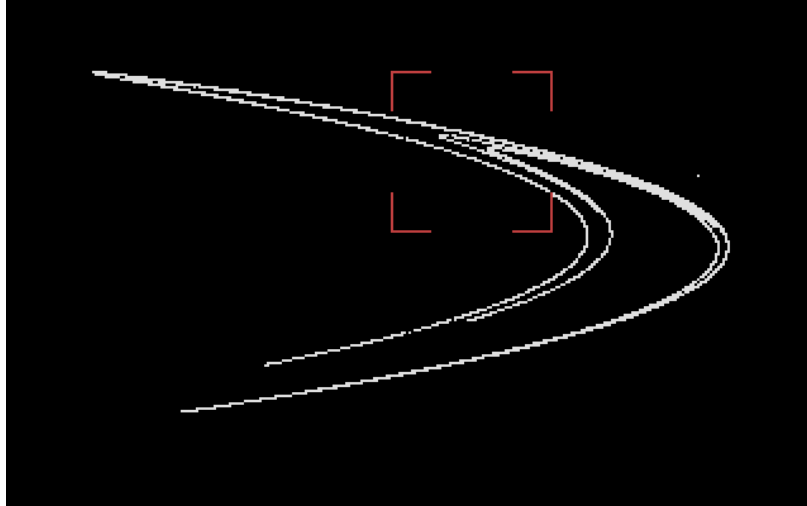
The Hénon equations represent a simple combination of folding and stretching of mathematical space to produce an image (attractor) that is easy to compute but still poorly understood by mathematicians. The equations are written as

$$\begin{aligned}x_{\text{new}} &= y + 1 - 1.4x^2 \\ y_{\text{new}} &= 0.3x\end{aligned}$$

In contrast to the logistic equation, these equations produce chaotic behavior from the get-go, and it is impossible to predict where the next point will appear on the screen. Slowly, a ghostly image will start to materialize and will progressively sharpen and become more defined the longer the calculations are allowed to go.



Therefore, there is no defined end-point to the calculations, which will stop only when you press the **SPACEBAR**. At that point, you may again open a selection window by pressing **W** and move it around with the arrow keys (**E/S/D/X**).



Press **ENTER** when you are satisfied with the location of the window, and the calculations will restart, but this time only showing the zoomed area. Here again, the deeper the zoom level, the longer it will take to compute the image, and the first 3 images are stored in memory and can be accessed back and forth by pressing the **B** and **F** keys respectively. If you wish, you can select another zoom area and explore again. *You will not be able to go back and view earlier images if you go beyond zoom level 3.*

What do you notice as you increase magnification? Yes, it is a fractal!

Lyapunov Space

How to run:

7. Choose Option #3 LOAD AND RUN from the E/A menu
8. At the FILE NAME? prompt, enter DSKx.LYAPUNOV (DSKx can be any storage device you want)
9. At the PROGRAM NAME? prompt, enter START

How to use:

Lyapunov spaces are based on the logistic equation mentioned previously [$P=PR(1-P)$], except that for each pixel on the screen the calculations are repeated a large number of times while alternating R between 2 fixed values in a specific pattern. Each calculated P is used to create a running total for each repetition cycle using the formula below

$$\text{Total} = \text{Total} + ((\log|R-2RP|)/\log 2)$$

and then the resulting total is divided by the number of repetitions. This gives you what is called the Lyapunov exponent. If the exponent is negative, then chaos is *not* present, otherwise it is. Here, chaos is represented as black, and everything else as white.



This particular image is produced using the pattern 222222111111, with 1 representing the value of R along the x-axis and 2 the one

along the y-axis and 200 calculation cycles for each pixel. Other patterns will produce widely different images and it will be up to you to experiment. In order to change the repetition pattern, you will need to edit the program source code by loading it into a text editor and modifying the data at the label PATTRN then saving and recompiling using the E/A assembler.

*Each image will take about **36 days** on a stock TI to calculate! This is a good opportunity to use an emulator instead...* The computer will display a progress bar as it toils on, and the image will be displayed once the calculations are done.

Final Words

If these programs have piked your interest in chaos, the references below will help you advance and explore further. I personally take great pleasure in pushing the TI 99/4A to the limit and prove that it is capable of more than just playing games and that it can be a serious scientific tool.

1. *Chaos. Making a New Science.* James Gleick 1987
2. *The Tinkertoy Computer.* A.K. Dewdney 1993