Assignment 1: Report

Abhijit Chakraborty

January 28, 2019

Q1: Duffing Oscillator

The Duffing oscillator can be solved numerically using the `odeint.c` program available in the numerical recipes package. The program uses adaptive stage size Runge-Kutta method to solve the differential equation. The `xodeint.c` example can be modified to create a working code for solving the Duffing differential equation system.

$$\dot{x} = p, \tag{1}$$
$$\dot{p} = -\gamma p + 2ax - 4bx^3 + F_0 \cos(\omega t) \tag{2}$$

Using parameter values $\gamma = 0.1$, $a = 0.5$, $b = 0.25$, $F_0 = 0.35$, and $\omega = 1.4$ yields a strange attractor (Figure 1). The strange attractor results from the Poincare section of the phase space.

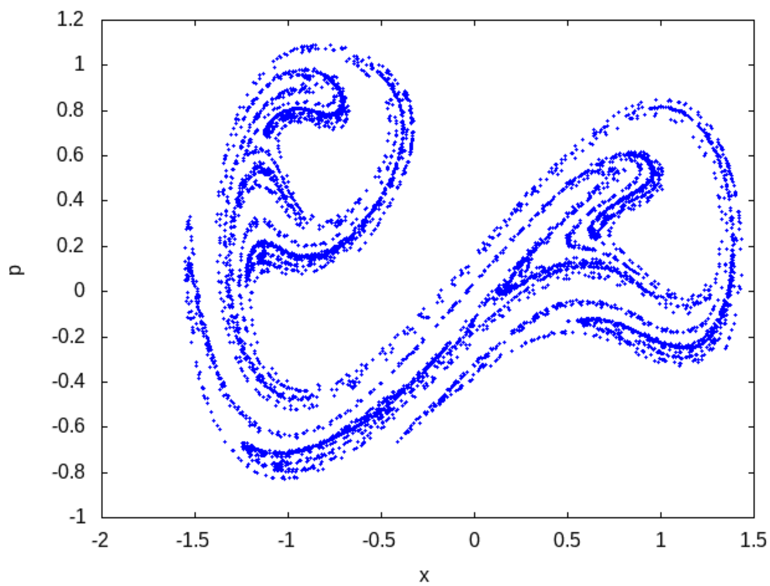

Figure 1: Poincare section of the Duffing oscillator phase space.

To obtain the points, the value of x and p after each integration of time period $2\pi/\omega$ is written to a data file. The initial conditions used for the simulation is taken to be $x_0 = 0.0$ and $p_0 = 0.0$.

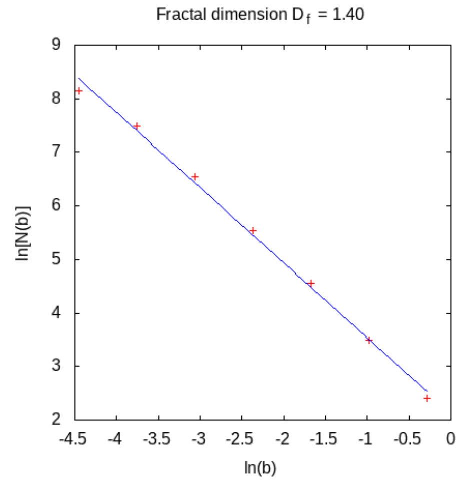

Figure 2: Linear fit of data to determine fractal dimension.

Fractal Dimension

The resulting strange attractor is actually a fractal diagram which has a dimension $1 < D_f < 2$. The dimension can be determined using the method described in the book $N(b) \propto b^{-D_f}$ where b is the square-grid size. Using this method we can determine the fractal dimension of a line and a square to be 1 and 2 by the following argument.

Let's divide the line segment into n equal parts, which yields $b = 1/n$. So, we need $N(b) = n$ number of segments to cover the whole segment. This yields $D_f = \frac{\log(n)}{\log(1/n)} = 1$. Thus, the fractal dimension is $D_f = 1$. For a square, if we we divide the side by 2, then we need $2^2 = 4$ squares to cover the whole square. Similarly, if the side is divided into n segments, then we need n^2 squares to cover the whole square. So the fractal dimension is $D_f = \frac{\log(n^2)}{\log(n)} = 2$.

Using this same method, the fractal dimension of the strange attractor can also be determined. The program `dimension.c` calculates the dimension of the Poincare section Figure 1. The program runs a loop on l starting from 4 and going to 128. It divides the square of size 3 into grids of size $(3/l) \times (3/l)$ and count the number of grids that cover at least one point of the strange attractor. It also prints $\log(N(b))$ and $\log(b)$ to a file. The calculation yields,

$$D_f = 1.40 \pm 0.06. \tag{3}$$

The fitted line with the above slope along with the data points are shown in the Figure 2.

Q2: Diffusion Limited Aggregation

The program `dla_final.c` asks for lattice site points, number of seeds and seed for the random number generator as user input and generates two data files `data.dat`, `gyration.dat`. Plotting the `data.dat` for 9000 seeds and a 151×151 lattice generates Figure 3. The provided program was modified according to the description given in the book where the walker starts from the boundary and has three neighbors in the sites at the boundary, two neighbors at the corner and the walkers don't die. With this modification, after adding 9000 particles, the site positions in the lattice matrix is written in a file and plotted using gnuplot. The resulting figure is shown in Figure 3

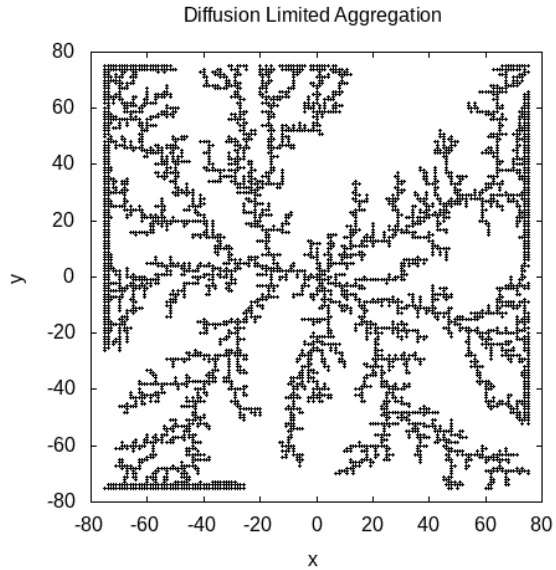

Figure 3: Diffusion limited aggregation for 9000 seeds and 151×151 lattice.

The program also calculates the radius of gyration R_g, after adding every 200 particles in the lattice and prints $\log(R_g)$ and $\log(N)$ to a file. The formula to calculate the radius of gyration is given by,

$$R_g^2 = \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{r_i}^2\right) - \boldsymbol{r_0}^2 \tag{4}$$

where $\boldsymbol{r_0}$ is the mean distance. Plotting $\log(N)$ vs $\log(R_g)$ yields the fractal dimension D_f, which in this particular case yields,

$$D_f = 1.86 \pm 0.01. \tag{5}$$

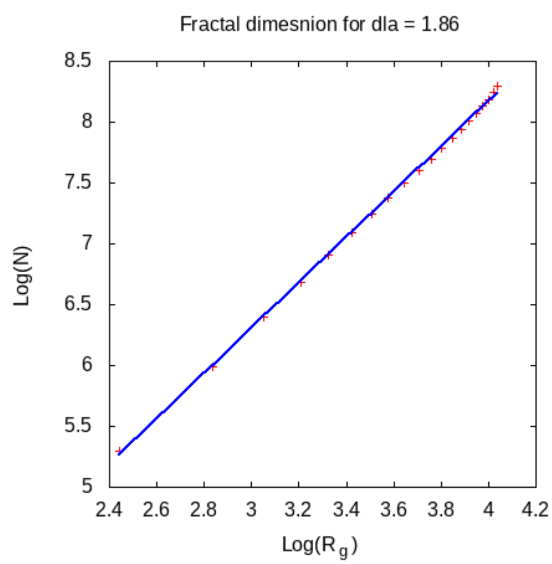

Figure 4: Linear fit of the $Log(N)$ vs $Log(R_g)$ data to find DLA fractal dimension.

Due to overlapping of sites, the slope of the curve changes after adding about 5000 particles and so last few of the data points are deleted from the data file to obtain a reasonable fit. The resulting linear fit is shown in the figure Figure 4.