

MAS212 Scientific Computing and Simulation

#3: Plotting with matplotlib

<http://sam-dolan.staff.shef.ac.uk/mas212>

Key resources:

- pyplot tutorial: http://matplotlib.org/users/pyplot_tutorial.html
- Lec 3: <http://sam-dolan.staff.shef.ac.uk/mas212/docs/l3.pdf>

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline # include plots in ipython notebook
```

1. Simple plots:

- Plot $\sin^2(x)$ and $\sin(x^2)$ using the `plt.plot()` function, in the range $0 \leq x \leq 4$;
- Label the x and y axes, and add a title;
- Change the line colours to green and magenta, and make them thicker;
- Add a legend;
- Save the plot to file with `plt.savefig("test.png")`. Check the file.

2. Multiple plots. Create a figure with four subplots in a 2×2 array. Using the lectures notes, plot four **Lissajous curves** with $b/a = 1, 2, 3$ and π in four separate subplots.

3. Elliptic curves (http://en.wikipedia.org/wiki/Elliptic_curve). Consider the equation

$$y^2 = f(x),$$

where $f(x)$ is a cubic (a polynomial of degree 3). Plot in the xy plane for the case $f(x) = x^3 + ax + 1$ with $a = 1, 0, -1$ and -2 .

4. Coin flipping. Imagine flipping a coin $N = 10$ times and recording the number of 'heads'.

```
def heads(N=10):
    return sum(np.random.randint(0,2,N))
```

Plot a **histogram** of the frequency distribution of the 'number of heads', when this trial is repeated $n = 1024$ times. How is this distribution related to the n th row of Pascal's triangle?

5. Monte Carlo integration (http://en.wikipedia.org/wiki/Monte_Carlo_integration).

Using the method of Monte Carlo integration, described in Lec 3, estimate (a) the area of a unit circle and (b) the volume of a unit sphere. Illustrate (a) with a scatter plot (`plot.scatter()`). Use (b) to find an estimate for π . Explain how the error in your estimate scales with n , the number of points that are randomly generated.

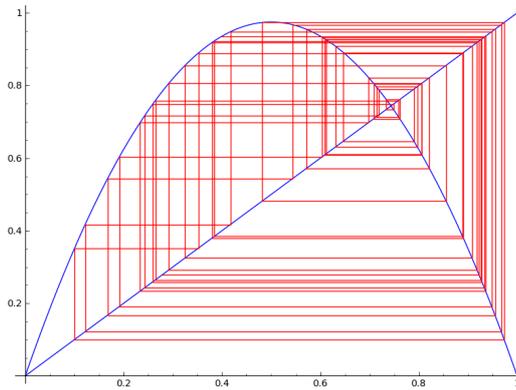
6. The logistic map (http://en.wikipedia.org/wiki/Logistic_map)

(a) Investigate the logistic map

$$x_{n+1} = rx_n(1 - x_n)$$

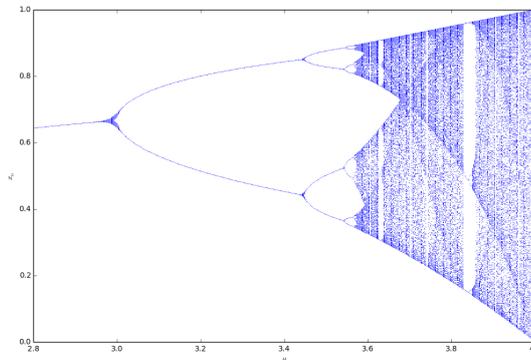
by plotting sequences of values x_k for $r = 2.5, 3.5, 3.7, 3.82843$ and 4.0 .

(b) Use `matplotlib` to create a **cobweb diagram**, similar to that shown below:



Credit: <http://www.ocf.berkeley.edu/~morawski/blog/category/dynamical-systems/>

(c) Use `matplotlib` to create a **bifurcation diagram**, similar to that shown below:



7. Fractals. Referring to Lecture 3, use a 2D array and `plt.imshow()` to plot the region of the Argand diagram that does not diverge under N iterations of the map $z \rightarrow z^2 + c$, with $c = 0, -0.75, i$ or any value of your choosing.