MAS212 Scientific Computing and Simulation

#5: Animations with `matplotlib.animation.FuncAnimation`

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

Key resources:

- Lec 9: `http://sam-dolan.staff.shef.ac.uk/mas212/docs/l9.pdf`
- Code examples: `http://sam-dolan.staff.shef.ac.uk/mas212/code/`
- `https://jakevdp.github.io/blog/2012/08/18/matplotlib-animation-tutorial/`
- `http://matplotlib.org/examples/animation/index.html`

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from matplotlib.animation import FuncAnimation
# %matplotlib nbagg   # Uncomment this if you want the animation to appear in the notebook,
                      # rather than in a separate window.
```

1. Testing the animations.

Download the Python scripts for animation from the course website. Paste the code into IPython Notebook. Run the code and watch the animation. Note that `particle_box.py` and `lorenz.py`, described on the blog of Jake Vanderplas (link above), are particularly interesting examples.

---

2. Simple example.

Start with the code in `sinwave.py`. Change the `animate` function so that it uses the following function instead:

$$y(x) = \exp\left(-\frac{10(x-x_0)^2}{1.5+\cos(x_1)}\right), \qquad x_0 = 5(1+\sin(0.01i)), \quad x_1 = 0.05i,$$

where i is the integer that is incremented by one each time `animate` is called. Change the plot range to $0 \leq x \leq 10$ and $0 \leq y \leq 1$. What does the animation show?

---

3. Animating ODEs

Make an **animated** phase plot of the modified predator-prey (Lotka-Volterra) equations from part 3(b) of assignment 2,

$$\begin{aligned} \frac{dx}{dt} &= x(1-gx) - \frac{xy}{1+hx}, \\ \frac{dy}{dt} &= -fy + \frac{xy}{1+hx}, \end{aligned}$$

with $f = 8/3$, $g = 3/50$, $h = 3/20$. Try initial conditions $x_0 = r, y_0 = 1$, where $r \in (0, 3]$. What can you infer from an animated phase plot that you can't infer from a static phase plot?

4. Animating the Newton-Raphson fractal

Recall the image of the basins of attraction in Assignment 1, part 2(c). Now consider the set S of all points in the complex plane that map onto 0 under iteration of $z \to z - \frac{z^3-1}{3z^2}$. If w is in the set S, then the three complex roots of $2z^3 - 3wz^2 + 1 = 0$ are also in the set S (as they map onto w and so eventually on to 0). Write code to generate a list of arrays `l`, where the array `l[k]` has 3^k elements. Now make an animation whose first frame shows only the point at 0 in the complex plane (where $x = \mathrm{Re}(z)$, $y = \mathrm{Im}(z)$). The next frame should show 0 and the points which map onto it after one step, i.e., $\{1, e^{2i\pi/3}, e^{-2i\pi/3}\}$. The next frame should show a further 9 points, which map onto zero after two steps. The next frame, 27 new points, and so on. Your animation should reveals the structure of the Julia set.

If these instructions are not clear, please ask the tutorial helpers.

---

5. Animating the Barnsley Fern.

The Barnsley fern is made from points in the xy plane generated by a stochastic algorithm. Starting with the notebook at `http://sam-dolan.staff.shef.ac.uk/mas212/notebooks/Fern.ipynb`, make an animation of the fern being created.