MAS212 Scientific Computing and Simulation

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

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/`
- `http://matplotlib.org/examples/animation/index.html`

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
%matplotlib notebook
```

1. Testing the animations.

Download the Python scripts for animation from the course website. Paste the code into Jupyter Notebook. Add the line `%matplotlib notebook`. Run the code and watch the animation.

You can also run the animations in Spyder. First, you need to make a change of setting. From the menu: `Tools -> Preferences -> IPython Console -> Graphics -> Backend` then set `Backend` to `Automatic`.

---

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 $0 \le x \le 10$ and $0 \le y \le 1$. What does the animation show?

---

3. Animating ODEs

Make an **animated** phase plot of the modified predator-prey (Lotka-Volterra) equations, describing a population of rabbits (x) and foxes (y),

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

with $f = 8/3$, $g = 3/50$, $h = 3/20$. (It may help to adapt the code `vanderpol.py`).
Try initial conditions $x_0 = r$, $y_0 = 1$, for various $r \in (0, 3]$. What do you see?

4. Animating a fractal

Try the code `fractal_anim.py`. This plots (in purple) the region of the complex plane z_0 that leads to a converging sequence $z_0, z_1, z_2, \ldots$ under the map $z_{i+1} = z_i^2 + c$. Try changing the values of c.

---

5. Animating the Barnsley Fern.

The Barnsley fern is made from points in the (x, y) 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.