Lotka-Volterra in Python

September 30, 2016

In this example document, we want to solve the Lotka-Volterra ODE system given by:

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{1}$$

$$\frac{dy}{dt} = \delta xy - \gamma y. \tag{2}$$

We will solve this problem numerically in Python.
Since we have a system of equations, we want to define it in *vector form*:

$$\begin{matrix} \frac{dx}{dt} = \alpha x - \beta xy \\ \frac{dy}{dt} = \delta xy - \gamma y \end{matrix} \equiv \frac{d}{dt} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{X} = F(X), \tag{3}$$

where

$$F(X) = \begin{bmatrix} \alpha x - \beta xy \\ \delta xy - \gamma y \end{bmatrix}. \tag{4}$$

This now is the derivative function we will define for Python. First we set our parameters and load the necessary libraries:

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        from scipy.integrate import odeint

        α = 2/3;
        β  = 4/3;
        γ = 1;
        δ = 1;
```

Next, we define the function handle. Notice that since $X = \begin{bmatrix} x & y \end{bmatrix}^T$, we have `x=X(1)` and `y=X(2)`.

```
In [2]: def F(X,t):
            x = X[0]
            y = X[1]
            dx = α*x-β*x*y
            dy = δ*x*y-γ*y
            return [dx, dy]
```

We can now pick our initial conditions and the time frame we want to integrate over:

```
In [3]: tspan = np.arange(0., 25.0, 0.1)    # arange takes args: tstart, tend, tincr
        X0 = [0.8, 0.4]
```

Now we're ready to solve the ODE:

```
In [4]: X = odeint(F, X0, tspan)
```

Now we can plot the solutions.

```
In [5]: %matplotlib inline
        plt.plot(tspan, X[:,0], label="x(t)")
        plt.plot(tspan, X[:,1], label="y(t)")
        plt.title("Lotka-Volterra")
        plt.xlabel("Population")
        plt.ylabel("Time")
        plt.legend(loc='upper right', shadow=True)

Out[5]: <matplotlib.legend.Legend at 0x7f0ea8aca1d0>
```