Solving Ordinary Differential Equations (ODEs) using Python

Simple differential equations can be solved numerically using the Euler-Cromer method, but more complicated differential equations may require a more sophisticated method. The `scipy` library for Python contains numerous functions for scientific computing and data analysis. It includes the function `odeint` for numerically solving sets of first-order, ordinary differential equations (ODEs) using a sophisticated algorithm. Any set of differential equations can be written in the required form. The example below calculates the solution to the following second-order differential equation,

$$\frac{d^2y}{dt^2} = ay + b\frac{dy}{dt}.$$

It can be rewritten as the following two first-order differential equations,

$$\frac{dy}{dt} = y' \quad \text{and} \quad \frac{dy'}{dt} = ay + by'.$$

Notice that the first of these equations is really just a definition. In Python, the function y and its derivative y' will be part of elements of an array. The function y will be the first element $y[0]$ (remember that the lowest index of an array is zero, not one) and the derivative y' will be the second element $y[1]$. In this case, you can think of the index as how many derivatives are taken of the function. In this notation, the differential equiations are

$$\frac{dy[0]}{dt} = y[1] \quad \text{and} \quad \frac{dy[2]}{dt} = ay[0] + by[1].$$

The `odeint` function requires a function (called `deriv` in the example below) that will return the first derivative of each of element in the array. In other words, the first element returned is $dy[0]/dt$ and the second element is $dy[1]/dt$, which are both functions of $y[0]$ and $y[1]$. You must also provide initial values for $y[0]$ and $y[1]$ which are placed in the array `yinit` in the example below. Finally, the values of the times at which solutions are desired are provided in the array `time`.

```
from scipy import odeint
from pylab import *  # for plotting commands

def deriv(y,t):  # return derivatives of the array y
   a = -2.0
   b = -0.1
   return array([ y[1], a*y[0]+b*y[1] ])

time = linspace(0.0,10.0,1000)
yinit = array([0.0005,0.2])  # initial values
y = odeint(deriv,yinit,time)

figure()
plot(time,y[:,0])  # y[:,0] is the first column of y
xlabel('t')
ylabel('y')
show()
```

Note that `odeint` returns the values of both the function $y[0]=y$ and its derivative $y[1]=y'$ at each time. In the example above, the function is plotted versus the time.

For a second example, suppose that you want to solve the following coupled, second-order differential equations,

$$\frac{d^2x}{dt^2}=ay \quad \text{and} \quad \frac{d^2y}{dt^2}=b+c\frac{dx}{dt}.$$

In order to rewrite these equations as a set of first-order differential equations, start by defining

$$\frac{dx}{dt}=x' \quad \text{and} \quad \frac{dy}{dt}=y'.$$

The original equations can be written as

$$\frac{dx'}{dt}=ay \quad \text{and} \quad \frac{dy'}{dt}=b+cx'.$$

To use `odeint`, the four first-order equations must be written as elements of an array. If we make the definitions,

$$z[0]=x, \quad z[1]=x', \quad z[2]=y, \quad \text{and} \quad z[3]=y',$$

then the four equations become

$$\frac{dz[0]}{dt}=z[1], \quad \frac{dz[1]}{dt}=az[2], \quad \frac{dz[2]}{dt}=z[3], \quad \text{and} \quad \frac{dz[3]}{dt}=b+cz[1].$$

These equations are now in a form necessary for the derivative function, which would be an array with four elements. Notice that the index of the array is not the number of derivatives of a single function in this case.

Exercise:
An example of a differential equation that exhibits chaotic behavior is

$$\frac{d^3x}{dt^3}=-2.017\frac{d^2x}{dt^2}+\left(\frac{dx}{dt}\right)^2-x.$$

(a) Write the differential equation as a set of *first-order* differential equations.
(b) Modify the example program to solve the equations with the initial conditions of $x=0$, $dx/dt=0$, and $d^2x/dt^2=1$.
(c) Plot the results for *t* from 0 to 100.

Additional documentation is available at:
http://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html
http://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html