Name:________________________________

Non-evaluation 6b

Introductory Programming
Fall 2006

The Lotka-Volterra model describes the interactions between two species in an ecosystem, a predator and its prey. A common example is rabbits and foxes.

The model is governed by the following system of differential equations:

$$dR/dt = aR - bRF$$
$$dF/dt = ebRF - cF$$

where

- a is the natural growth rate of rabbits in the absence of predation,
- c is the natural death rate of foxes in the absence of prey,
- b is the death rate per encounter of rabbits due to predation,
- e is the efficiency of turning predated rabbits into foxes.

We would like to write a MATLAB script that uses `ode45` to solve this system of equations over the range of time from $t = 0$ to $t = 200$ with the parameters $a = 0.1$, $b = 0.01$, $c = 0.1$ and $e = 0.2$.

The first step is to write a function that evaluates the derivatives dR/dt and dF/dt as a function of t and the current values R and F. But `ode45` only takes two input variables and returns one output variable, so in order to make the interface work, we have to pack R and F into a single input vector, and pack dR/dt and dF/dt into a single output vector.

```
function dVdt = lotka(t, V)
    % unpack the elements of V
    R = V(1);
    F = V(2);

    % set the parameters
    a = 0.1;
    b = 0.01;
    c = 0.1;
    e = 0.2;

    % compute the derivatives
    dRdt = a*R - b*R*F;
    dFdt = e*b*R*F - c*F;

    % pack the derivatives into a vector
    dVdt = [dRdt; dFdt];
end
```

Note that the output value has to be a *column* vector, which is why there is a semi-colon between the elements.

Now to use `ode45` we would write something like

```
ode45(@lotka, [0, 200], [100, 10])
```

In this case the initial condition is $R = 100$ and $F = 10$. How does the behavior of this system vary as the initial rabbit population changes?

To get the results in a form you can use later, you can assign the return values from `ode45` to `T` and `Y`.

```
[T, Y] = ode45(@lotka, [0, 200], [100, 10])
```

In this case `Y` is a matrix with two columns: the first column contains the values of `R`; the second contains values of `F`.

Lorenz attractor

According to my friend the Wikipedia, "The Lorenz attractor, introduced by Edward Lorenz in 1963, is a non-linear three-dimensional deterministic dynamical system derived from the simplified equations of convection rolls arising in the dynamical equations of the atmosphere. For a certain set of parameters the system exhibits chaotic behavior and displays what is today called a strange attractor..."

The system is described by these three differential equations:

$$\frac{dx}{dt} = \sigma(y - x) \quad (1)$$

$$\frac{dy}{dt} = x(r - z) - y \quad (2)$$

$$\frac{dz}{dt} = xy - bz \quad (3)$$

Common values for the parameters are $\sigma = 10$, $b = 8/3$ and $r = 28$.

We would like to use `ode45` to estimate a solution to this system of equations.

1. The first step is to write a function named `lorenz` that takes `t` and `V` as input variables, where the components of `V` are understood to be the current values of `x`, `y` and `z`. It should compute the corresponding derivatives and return them in a single column vector.

2. The next step is to test your function by calling it from the command line with values like $t = 0$, $x = 1$, $y = 2$ and $z = 3$? Once you get your function working, you should make it a silent function before calling `ode45`.

3. Assuming that Step 2 works, you can use `ode45` to estimate the solution for the time interval $t_0 = 0$, $t_e = 30$ with the initial condition $x = 1$, $y = 2$ and $z = 3$.

4. Now, how can we use `plot3` to plot the trajectory of x, y and z?