PROBLEM SET 7 (SUPPLEMENT)

Section 1: Lorenz Model

The Lorenz model is given by

$$\dot{X} = P(Y - X) \quad (1)$$
$$\dot{Y} = -XZ + rX - Y \quad (2)$$
$$\dot{Z} = XY - bZ \quad (3)$$

We are using ode45 again in this problem set! This Matlab routine solves system of ODEs using the 4th order Runge-Kutta method. **lorenz.m** is provided.

Lets take a look at **lorenz.m**. The function is called `lorenz`. The next line declares that dy be a vector of length 3 and initializes to zeros. The next few lines declare the various parameters (P, r, and b) in the Lorenz model. Finally, the gut of the function, the specification of the 3 first order ODEs:

```
dy(1) = P*(y(2) - y(1));
dy(2) = -y(1)*y(3) + r*y(1) - y(2);
dy(3) = y(1)*y(2) - b*y(3);
```

In this problem set, $P = 10, b = 8/3$ and r is the primary parameter of interest to play around with. In the past, you have done this for other systems. (eg. μ in $x_{n+1} = 4\mu x_n(1 - x_n)$, h in the driven pendulum, and α in the first problem of your midterm!) This diversity should give you some appreciation of the universality of these phenomena.

As given, r is set to 0.5 in **lorenz.m**. You should use an editor to change this if you want to solve the model for other values of r and REMEMBER to save your changes before running ode45. To start Matlab, `add matlab` and execute `matlab &`.
An example run may look like

```
>> options = odeset('RelTol',1e-4,'AbsTol',[1e-6 1e-6 1e-8]);
>> tspan=0:0.01:100;
>> [t,y] = ode45('lorenz', tspan, [0.2 0.2 0.3], options);
```

The first line sets various options controlling the numerical tolerances. The second line sets the time interval and increment size. Initial conditions are:

$$X(0) = 0.2$$
$$Y(0) = 0.2$$
$$Z(0) = 0.3$$

Note that y stores all the time series for X, Y, Z, in consistent with the specification in **lorenz.m**.

$$\begin{array}{lll} \mathtt{y}(:,1) & \mathtt{is} & X \\ \mathtt{y}(:,2) & \mathtt{is} & Y \\ \mathtt{y}(:,3) & \mathtt{is} & Z \end{array}$$

To plot various time series

```
>>plot(t,y(:,1),'-');
>>plot(t,y(:,2),'-');
>>plot(t,y(:,3),'-');
```

To plot XY, XZ, and YZ projections as your Poincar'e section

```
>>plot(y(:,1), y(:,2), '.');
>>plot(y(:,1), y(:,3), '.');
>>plot(y(:,2), y(:,3), '.');
```

To plot trajectory in 3-D.

```
>>plot3(y(:,1), y(:,2), y(:,3));
```

To see the time evolution (animation),

```
>>comet3(y(:,1), y(:,2), y(:,3));
```

To investigate exponential divergence of $small$ differences in initial conditions. You should run another session of ode45 but saving your XYZ in a different variable name than y. eg.

```
>> [t,g] = ode45('lorenz', tspan, [0.200000001 0.2 0.3], options);
```

Note the almost zero deviation from the previous run. To compute the "distance" between points in two time series

```
>> distance = sqrt((y(:,1) - g(:,1)).^2 + (y(:,2) - g(:,2)).^2 +
   (y(:,3) - g(:,3)).^2);
>>
>>
>> plot(t,log(distance));
```

An exponential divergence will correspond to a straight line with positive slope on a semilogy plot. If your plot looks irregular, you can try average over several runs of nearby ICs.

OK... hope I have given enough information on this problem set. Please let us know immediately if you experience any problems.