Lab 5/6 for Math 398 Section 952: Differential Equations and Linear Algebra

Thomas Shores
Department of Math/Stat
University of Nebraska
Fall 2002

This document constitutes your fourth lab assignment, which will be largely worked in the lab session. It will draw on material from Lecture 5, so it would be a good idea to go to my Home Page (http://www.math.unl.edu/~tshores/) and thence to my Teaching Page, thence to the Math398 Section 952 Home Page. Really, it would be a good idea to bookmark the class home page.

Reminder: Here is how to do these lab assignments. You will need to open a Matlab session and have this pdf file open for reading as well. To keep a recording of your work, you issue the following command to Matlab

```
> diary 'myfile'
```

Matlab will then send a copy of all your typed input and the output to a file called 'myfile.' For clarity, use descriptive names for your files, such as 'jsmithasgn1' so that when I save the files that you will email me, I can tell what it's about by the title. If at any point you want to stop the diary feature, issue the command

```
> diary off
```

To resume the diary feature, simply type

```
> diary on
```

This will cause input to be appended to myfile. You can make comments in your homework file by typing % at the command line and this too will be recorded. For example

```
> % This is a comment.
```

Be sure to start your file with the comments

```
> % Name:  yourname
> % Email:  your email addresss
```

When you end your session, the file will be closed and you can view it and even edit it with a text editor. As a matter of fact, you can even edit and view it with the Matlab Editor. Just type

```
> edit myfile
```

For problems where you are asked to write a small script or function file, you should use the Matlab editor to do so, and once you are satisfied that it works correctly, save this copy for emailing to me as another attachment. Be sure to put your name in the script or function file by way of a comment line:

```
> % programmer:  J. Smith, for exercise x of assignment xx.
```

When you have finished the assignment, email the files to me as attachments. My email address is `tshores@math.unl.edu`. Here is the assignment. As usual, turn the diary command on to record your session. Unless otherwise indicated, it is not necessary to save the graphics you create for the assignment. Here is the assignment.

Problems

The first two problems are the basis for a game that we will play. It goes like this: a rabbit runs a predetermined path in the xy-plane given parametrically as a function of time t and starting at time $t = 0$ at the point $(1, 0)$. A fox which is located at the point $(3, 0)$ spots the rabbit at $t = 0$ and gives chase. The fox always runs in the direction pointed straight at the rabbit and at a speed that is a positive constant factor k times the current speed of the rabbit. We want to simulate this scenario and generate suitable graphics.

1. Write a function rabbitpos.m which takes a time t and returns a column 2-vector that represents the position of the rabbit at time t. Then write a function rabbitvel.m that takes a time t and returns the velocity vector for the rabbit at time t. Of course, you should make sure they match. For starters, let's have the rabbit move at increasing speed in a path that spirals about and away from the origin, say

   $$r(t) = (1 + t/5)[\cos(t), \sin(t)]$$

   For convenience, make sure rabbitpos can handle a row vector t.

2. Next, write a function foxvel.m that takes a time t and position vector x and calculates the fox's velocity at time t and position x. Use the speed factor k as a global and also use ctime as a global (initialized to 0) for the "caught time." Here, we understand that if the fox's position is closer than $0.01$to the rabbit's position then the rabbit is caught and the fox just continues along the same path as the rabbit, so that foxvel(t,x) = rabbitvel(t). So the function should do the following:
   (a) Calculate the distance from fox to rabbit (use norm(rabbit(t)-x)).
   (b) If smaller than 0.01, return rabbitvel(t) and set if ctime = 0, set ctime=t, else return a vector in the direction rabbit(t)-x of magnitude k*rabbitvel(t).
   Now plot the rabbit path, set hold on and grid. Find by experiment the smallest speed factor t such that the rabbit gets caught before time $t = 10$ and plot the result. Remember to reset ctime and k for each run.

3. This is an experiment in matrix stability. Reproduce the steps in the "Linear equations" section of Lecture 7, except you should experiment until you find a size for n such that the error is about 4 in magnitude. Now calculate the condition number of the matrix. How many digits of accuracy does this suggest that you will lose in solving a system with this matrix as coefficient matrix?

4. This experiment illustrates the eigenvalue features of Matlab. Create a function fcnC(alpha) that creates a companion matrix $C(p(x))$ for the polynomial

   $$p(x) = x^5 + \alpha x^4 + 3x^2 - x + 1$$

and then computes the largest eigenvalue of this matrix in absolute value (this is called the *spectral radius* of the matrix). You will need to use the compan command, as well as max and abs. Now use this function to plot the spectral radius as α ranges from -2 to 1 and calculate the eigenvalues for the value of α in this range with (roughly) the smallest spectral radius.