Math 230 — Hour Exam 1

March 2, 2015

General Directions. This is an open-book, open-notes, open-computer test. However, you may not communicate with any person, except me, during the test. You have the full class period (50 minutes) in which to do the test. Put your answer to each question in the space provided (use the backs of pages if you need more space). Be sure to **show your work**! I give partial credit for incorrect answers if you show correct steps leading up to them; conversely, I do not give full credit even for correct answers if it is not clear that you understand where those answers come from. Good luck.

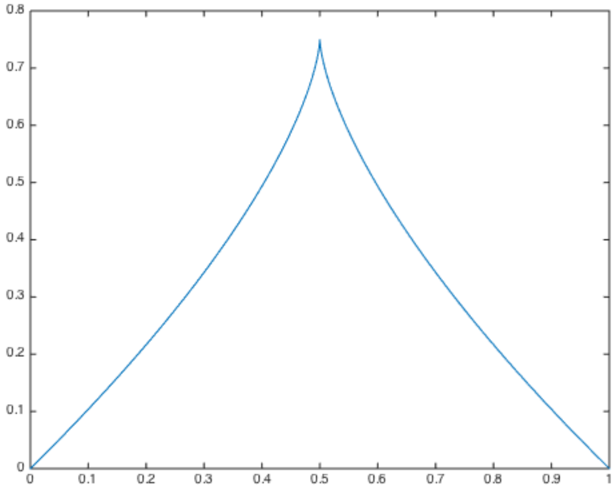
This test contains 5 questions on 5 pages.

Question 1 (5 Points). I recently took my car to the mechanic for a tune-up. The bill I got had a charge for labor and a charge for parts; the total I owed the mechanic was the labor charge, the parts charge, and an 8% tax on both. If I were going to write Matlab statements to calculate my total bill, they would start like this:

```
Parts = 44.95;
Labor = 30;
```

Write one or more additional statements that continue the ones above and calculate the tax I owe and the total bill. Your statements should place each of these quantities in a separate, appropriately named, variable. You may also introduce other variables if you wish.

Question 2 (10 Points). Here is a graph of a pointed curve:



This curve is defined by the parametric equations

$$x(t) = 4t^3 - 6t^2 + 3t$$
$$y(t) = 3t - 3t^2$$

as t ranges from 0 to 1.

Write one or more Matlab statements that plot this curve, assuming no variables are already defined (i.e., your answer needs to include statements that assign values to any variables you use elsewhere in that answer).

Question 3 (15 Points). If you make an *n*-by-*n* matrix in which every element is equal to *n*, and then multiply it (via matrix multiplication) by an *n*-by-*n* matrix in which every element equals $1/n$, you get the original matrix of *n*'s back. For example, if $n = 2$, the original matrix would be

$$\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

and the matrix of reciprocals would be

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

You can check that if you multiply these two matrices together, you get back the matrix

$$\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

I have shown below a Matlab statement that reads *n* from the user. Write a series of additional Matlab statements that use this *n* to demonstrate the above fact about matrices. Specifically, your statements should generate the *n*-by-*n* matrix of *n*'s, and the *n*-by-*n* matrix of reciprocals, and then display their product.

```
n = input( ‘Enter the size of the matrix ‘ );
```

Question 4 (15 Points). Here is a theoretically not bad, although somewhat outdated, way to encrypt information and send it to someone else. The information to be encrypted is a vector of numbers (which could, for instance, be text if letters are represented by numbers, A = 1, B = 2, etc., or a list of measurements, or anything else). Call this vector the plaintext. To encrypt the plaintext, multiply it by a vector of random numbers (the key) that is the same length as the plaintext. Finally, to guard against damage to the encrypted message while in transit to the other person, append the sum of the original numbers to the end (such a sum is called a checksum).[1]

For example, if the plaintext were [1, 2, 3], and the key were [0.1, 0.7, 0.3] then the encrypted message and checksum would be the vector [1×0.1, 2×0.7, 3×0.3, 1+2+3], or [0.1, 1.4, 0.9, 6].

Write a series of Matlab statements that carry out this encryption for a plaintext vector **P**. Do not assume any particular contents for **P**; you do not need to write a statement to initialize **P**. Other than **P**, however, your statements should calculate everything they need, including, in particular, the key. Your statements should leave the encrypted message, including the checksum, in another vector named **C**.

[1] If you are interested, although it is <u>not</u> part of what you need to do to answer this question, the recipient of such a message would divide all but the last received number by the key to get back the plaintext, and then add all of the plaintext numbers together and see if the sum is equal to the last received number—if not the message has been damaged somehow.

Question 5 (5 Points). Here is a script that prints the *a*th through *b*th letters of the first line of “Mary Had a Little Lamb.” Note that this script assumes that variables *a* and *b* are initialized elsewhere, e.g., through the command line.

```
lyrics = ‘Mary had a little lamb, little lamb, little lamb’;
fprintf( ‘%s\n’, lyrics(a:b) );
```

Add code around these statements to make a <u>function</u> that prints the *a*th through *b*th letters of the first line of “Mary Had a Little Lamb” and that takes *a* and *b* as its arguments. Your function needn’t have any result variables.