Matlab in Math 340, the first steps

Finding a computer with Matlab

You may buy a student version of Matlab to use on your own computer, check out the bookstore. If you are familiar with ssh or telnet you can log in to a university computer from your home computer and run Matlab that way. But most of you will probably use a public computer. Go to an owl lab and find a computer, there is an owl lab in room 0203 of the Math building. Others may be found in the list in `http://www.oit.umd.edu/wheretogo`. Log in. Start Matlab, for example on a Unix machine, type in `matlab` at the prompt in a terminal window. You may have to first type `tap matlab`. On PCs there will be some icon to click. Matlab will now give you a prompt:

\>>

Using Matlab

You can enter a matrix into Matlab as in the following example of a 3×3 matrix:

```
>> A =[ 1 2 3; 4 5 6; 7 8 9]
```

In other words, enclose the matrix in square brackets and indicate the end of a row with a semicolon. Commas between the row entries are optional. Instead of the semicolon you could also start a new line. For example the following 2×2 complex matrix:

```
>> A = [ 1+i, 7
   3, 8-sqrt(-4)]
```

In this assignment you will generate random matrices. Before generating any random matrices, type in the following to start the random numbers at a random value:

```
>> rand('state',sum(100*clock));
```

Now if you wanted to set A to be a random 5×7 matrix, you could type:

```
>> A=rand(5,7)
```

and A will be a random matrix with all entries between 0 and 1. To generate a random 3×4 matrix with integer entries from 2 to 13, you would type:

```
>> A = randint(3,4,[2,13])
```

Matlab allows you to do various matrix operations easily. For example:

- `rref(A)` is the reduced row echelon form of A
- `A'` is the conjugate transpose of A. (So for a real matrix it is just the transpose.)
- `inv(A)` is the inverse of A
- `A*B` is the matrix product
- `X = A\K` finds a solution to $AX = K$. Note: if A is not square then AX may not equal K, but will instead be as close to K as possible, This is a least squares solution which we will be looking at later in the course.
- `det(A)` finds the determinent of a square matrix.

When you generate a large matrix, you may not wish to have matlab print it out. You can suppress Matlab printout of a result by ending the command with a semicolon. For example, `A = rand(7,9);` will generate a random 7×9 matrix but not print it out. I encourage you to do this when appropriate.

My web site has links to a few Matlab tutorials with more information if you need it. See the links given in `www.math.umd.edu/users/hck/340.html`

Seeing whether two large matrices are equal

A good way to see whether two large matrices A and B are equal is to look at their difference $A - B$ and see whether it is zero or very small in comparison to A and B. This sure beats comparing each of the 100 entries in two 10×10 matrices. Note that because of roundoff error two matrices might not be quite equal even though theoretically they should be equal and in fact would be equal if the computer could do exact arithmetic. For example, `D-inv(inv(D))` will generally not evaluate to 0 even though it should.

If D and E are large, printing out $D - E$ can take up a great deal of paper. So here is a good way to check whether two matrices D and E are very nearly equal. The command

```
>> max(max(abs(D-E)))
```

will print out the entry of $D - E$ with largest absolute value. If this is quite small compared with the entries of D then D and E are equal for all practical purposes. (The reason two max are needed above is that the inner max gives you a vector with the maximum entry in each column, and the outer max finds the maximum of those numbers.)

Saving and printing output

One way to save your output is to cut and paste from the Matlab window to your favorite word processor, but be careful to include all relevant output if you do this. Another is to use Matlab to save your session for you. If you are not on a unix machine you will probably need to give the command

```
>> cd H:\
```

to make sure Matlab saves to your home directory on wam. Or to save output to a floppy on a PC, type

```
>> cd A:\
```

Now you can type in the command:

```
>> diary prob1
```

and Matlab will save all following output to a file named prob1. When you wish to stop saving, type

```
>> diary off
```

You may then print the file prob1 as is and write your commentary by hand, or edit in your commentary with a word processor. Note, to print off a public machine you will need to set up an account to pay the printing charges.

Matlab problems due Sept. 27

I encourage you to work in groups of two or three people, but you may work alone if you wish. Each group will hand in just one copy of the assignment. It is understood that all people in the group will contribute significantly to the assignment. Your completed project should include a printout of relevant computer output as well as additional analysis of the problem. If you edit the computer output– be sure to so indicate in an unambiguous fashion.

Problem 1: Generate a random 5×7 matrix by setting `A = rand(5,7)`. Put it in row reduced echelon form by using the command `rref(A)`. Are the columns of A linearly independent? What is the span of the columns of A?

Problem 2: Generate a random U in $\mathbb{R}_{7\times 1}$ by setting `U = rand(7,1)`. Calculate the product $K = AU$, using your matrix A in problem 1. (Recall matrix multiplication is denoted by $*$ so you would set `K = A*U`). Now use Matlab to solve $AX = K$ by setting `X = A\K`. You know that $X = U$ is a solution of $AX = K$. Did Matlab give you the solution $X = U$? What is going on here? Find all solutions to $AX = K$. (Challenge: find all solutions without doing any more calculations, just using the calculations you did already in problems 1 and 2.)

Problem 3: Generate random 4×4 matrices A and B. Are the columns of A linearly independent? What about the columns of B? What if you did this 100 times, would you expect the same result? What is the span of the columns of A?

Problem 4: Using the matrices A and B of problem 3 find $\det(A)$, $\det(B)$, $\det(AB)$, and $\det(A^{-1}B^T)$. Check that the expected relations hold, $\det(AB) = \det(A)\det(B)$, $\det(A^{-1}B^T) =?$.

Problem 5: Use the matrices A and B of problem 3. For each part below, calculate the matrices $A1, A2, A3$, and $A4$ and determine which are equal without printing them out. If Matlab does not calculate them as exactly equal, but they are extremely close you should say they are essentially equal, meaning that they would have been equal if Matlab had used exact arithmetic.

a) $A1 = (A^T B^T)^T, A2 = (B^T A^T)^T, A3 = BA, A4 = AB$.

b) $A1 = A^{-1}B^{-1}, A2 = B^{-1}A^{-1}, A3 = (BA)^{-1}, A4 = (AB)^{-1}$.