Assignment #1

Due Monday 14 September, 2015 at the start of class

Please read Lectures 1, 2, 3 in the required textbook *Numerical Linear Algebra* by Trefethen and Bau.

A major purpose of this assignment is to familiarize you with MATLAB/OCTAVE. You will need to find, download, or purchase a copy of MATLAB/OCTAVE. (Or PYLAB=PYTHON+SCIPY+IPYTHON; see the "Comparison" handout.) Make sure you can open and close the command window, use it as a calculator, and plot simple things. Make sure you can create a new program (text file), save it, edit it, and run it at the command line by typing its name.

Do these exercises:

P1. Consider the 3×3 real matrix

$$A = \begin{bmatrix} 2 & 1 & 1 \\ -2 & 0 & 1 \\ 4 & 4 & 6 \end{bmatrix}$$

(a) By hand on paper:

- compute the rank and determinant of A,
- compute the eigenvalues of A,
- compute the inverse of A (if possible),
- compute the inverse of $B = A(2:3, 2:3)$ (if possible), and
- solve the linear system $Ax = b$ where $b = \begin{bmatrix} 1 & -5 & -6 \end{bmatrix}^*$.

(b) Now check your work at the MATLAB/OCTAVE command line. (*You'll use these* MATLAB/OCTAVE *commands:* `rank,det,eig,inv,\`. *Type* "`help det`" *etc. if needed.*)

P2. (*This problem is about the most basic algorithms of linear algebra, and about their elementary implementations. Nothing fancy. Be careful with indices!*)

(a) On paper, write the algorithm which computes the product of a rectangular matrix $A \in \mathbb{C}^{m\times n}$ and a column vector $v \in \mathbb{C}^{n\times 1}$. Use the textbook's notation for vector and matrix entries. Count the number of floating point operations exactly (i.e. as an exact expression in m and n).

(b) Now implement this algorithm in a MATLAB/OCTAVE program `matvec.m` which is a function; the first line will say

```
function z = matvec(A,v)
```

The code should start by extracting the sizes of the input objects using the MATLAB/OCTAVE command `size`. Use `error` if the user-supplied inputs `A` and `v` are incompatible sizes. Use `for` loops.[1] Check your code against some example $A \in \mathbb{C}^{4\times 3}$ and $v \in \mathbb{C}^3$, where you compute Av by hand. Check your code against "`A * v`" in MATLAB/OCTAVE in the same case.

(c) Write a function "`matmat`" for the product $C = AB$ of matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times k}$. Check your code against the MATLAB/OCTAVE result "`A * B`" on a randomly generated example where $m = 5$, $n = 4$, and $k = 3$.

P3. (*This problem shows some additional* MATLAB/OCTAVE *commands*: `randn,norm, abs,mean,plot,loglog,semilogy`. *Because I am proposing you generate 90 matrices, I absolutely do not want you to turn in a giant table of the values of their determinants and so on, much less the matrices themselves! Instead, use sentences to state conclusions, and use plots to communicate data and patterns. Computes averages over the 10 tries as needed, but note that sometimes raw data can appear nicely in a plot.*)

Write a MATLAB/OCTAVE script (i.e. a program which is not a function) to generate 10 random matrices of size $m \times m$ for each of these powers of two: $m = 2, 4, 8, \dots, 512$. The matrix entries should be normally-distributed random real numbers with mean zero and standard deviation one. For each of these matrices compute the rank, the 2-norm, the 2-norm of the inverse, and the absolute value of the determinant. Communicate these data using plots in reasonable ways.

Exercise 1.1 in Lecture 1.

Exercise 1.3 in Lecture 1.

[1] Yes, I know it can be done by colon notation. Use `for` loops anyway, so you see the underlying implementation.