Assignment #2

Due Monday 15 September, at the start of class

Please read Lectures 2, 3, and 4 in the textbook *Numerical Linear Algebra*, SIAM Press 1997, by Trefethen and Bau. An ongoing purpose in this Assignment is to familiarize you with MATLAB,[1] but emphasis is increasingly on ideas and algorithms from the textbook.

DO THE FOLLOWING EXERCISES from Lecture 2:

- **Exercise 2.1**
- **Exercise 2.3**

DO THE FOLLOWING ADDITIONAL PROBLEMS.

P5. Write a MATLAB function which uses `for` loops to compute the product $C = AB$ of an $m \times n$ matrix A and a $n \times k$ matrix B. The first line will be:

```
function C = mymatmat(A,B)
```

As for `mymatvec()` from Assignment #1, please make this a well-implemented function. Specifically, use `size()` to determine m, n, k, do error-checking to ensure compatible sizes, and add a few short comments to explain what is happening. Demonstrate that it is correct by running it on a couple of small cases where you have computed the correct answer by hand; the largest case you should do by hand might be $m = 3, n = 4, k = 3$, for example. (I recommend computing norms of differences to demonstrate correctness.) Then count the exact number of floating-point arithmetic operations, as a function of m, n, k.

P6. Use MATLAB to reproduce Figure 3.1 in the textbook. Start with plotting the unit balls in $\mathbb{R}^2$ just by plotting many points with unit norms. Then apply A to those points and generate the right-hand figures. `subplot` may be useful. Then add the annotation/decorations to match the appearance in a reasonable manner. For mathematical annotations, try `text(3,0,"$\|A\|_1=4$")`, for example.

P7. It is likely that you have already learned a recursive algorithm for computing determinants called "expansion in minors." If you do not know it, please look it up.

(a) Compute the following determinant by hand, showing your work to demonstrate that you can apply expansion in minors:

$$\det\left(\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}\right).$$

[1] You can use any language you want, but I am calling it "MATLAB" for brevity.

(b) For any matrix $A \in \mathbb{C}^{m\times m}$, find an exact recursion for the number of multiplication operations needed to compute $\det(A)$ by expansion in minors. (*Hint*: How much more work is the $m \times m$ case than the $(m-1)\times(m-1)$ case?) Describe the significance of this recursion in at least one sentence, and give a clear estimate which shows that the algorithm is appallingly expensive.

Comment: *If* $\det(A) = 0$ exactly *then* A *is not invertible. However, for matrices with generic real or complex entries, rounding error makes an exact zero value extremely unlikely on such matrices. On the other hand, the magnitude of* $\det(A)$ *does not measure or correlate with invertibility of* A *anyway! The next part addresses this idea.*

(c) Consider square, diagonal matrices A. Give a formula for $\det(A)$. Give a formula for A^{-1}, if it exists. Show by example that $\det(A)$ is often very large or very small even for obviously and easily invertible diagonal matrices with boring-magnitude entries.

Comment: *Based upon such ideas and results, I claim these generalities about determinants.*

1. *Numerical determinants should not be used to measure invertibility of matrices. (*Use the condition number instead; Lectures 4, 5.*)*
2. *If the value of a numerical determinant is actually needed, it should not be computed by expansion in minors. (*Use an $O(m^3)$ LU decomposition instead, for example; Lectures 20, 21.*)*
3. *Never use Cramer's rule. (*Do not learn it if you don't know it.*)*
4. *Determinants* are *indeed needed for changing variables in integrals. Typically the matrices are small, and this is numerically safe.*
5. *Computing determinants is a low priority, rarely-used algorithm within practical linear algebra.*