Assignment #5

Due Wednesday 15 October, at the start of class

Please read Lectures 7, 8, 10, and 11 in the textbook *Numerical Linear Algebra*, SIAM Press 1997, by Trefethen and Bau. The experiments in Lecture 9 are interesting, but not needed for any Homework or Exams.

DO THE FOLLOWING EXERCISES FROM THE TEXTBOOK:

- **Exercise 7.1**
- **Exercise 7.3** Start by explaining why $|\det(Q)| = 1$ if Q is unitary.
- **Exercise 8.2** Use your preferred language. Implicit in this, and similar questions, is to show that your code works! After the code, please show a brief command line session where you generate a generic/random matrix, run the code on it, and then verify that the outputs have the required properties. Use `norm()` to avoid spewing numbers at me. In other words, act like a professional. See my previous Homework solutions for examples.
- **Exercise 10.2** Same advice.
- **Exercise 10.3**

DO THE FOLLOWING ADDITIONAL PROBLEMS.

P12. Suppose $A \in \mathbb{C}^{m\times n}$, for $m \ge n$, is a matrix with ortho*gonal*, but not necessarily ortho*normal*, columns. Describe its reduced QR decomposition.

P13. *While we have used QR to solve linear systems, here we see that QR factorization has a completely different application. For more, see Lectures 24–29.*

(a) By googling for "unsolvable quintic polynomials" or similar, confirm that there is a theorem which shows that fifth and higher-degree polynomials cannot be solved using finitely-many operations, including nth roots ("radicals"). In other words, there is no finite formula for the solutions ("roots") of such polynomial equations. Who proved this theorem? When? Show a quintic polynomial for which it is known that there is no finite formula. (*You do* not *need to prove that it is "unsolvable"!*)

(b) At the Matlab command line, do the following:

```
>> A = randn(5,5);  A = A' * A      % create a random 5x5 symmetric matrix
...
>> A0 = A;                          % save a copy of the original A
>> [Q, R] = qr(A);  A = R * Q
```

```
...                                  % repeat about 10 times
>> [Q, R] = qr(A);  A = R * Q
```

We start with a random, symmetric 5×5 matrix A_0 and then generate a sequence of new matrices A_i. Specifically, each matrix is factored and then the next matrix is generated by multiplying-back in reversed order:

$$A_i = Q_i R_i \qquad \longrightarrow \qquad A_{i+1} = R_i Q_i.$$

What happens to the matrix entries when you iterate at least 10 times? What do you observe about this sequence of A_i? Now compare, both visually and using `norm()`, the vectors/lists `sort(diag(A))` to `sort(eig(A0))`.

Without turning in anything for this stuff, also do:

- *Use a* `for` *loop to see results from 100 iterations.*
- *Repeat the experiment for a* 13×13 *matrix. That is, there is nothing special about* 5×5.

(c) To see a bit more of what is going on in part **(b)**, show that

$$A_{i+1} = Q_i^* A_i Q_i.$$

So, at least this shows A_{i+1} has exactly the same eigenvalues as A_i; explain why.

(d) Apparently we have discovered an eigenvalue solver for symmetric matrices. Write a few sentences which relate the context from part **(a)** to the results in **(b)** and **(c)**. (*Hint. Think and speculate. Then read "A Fundamental Difficulty" in Lecture 25 to confirm your understanding.*)