Assignment #6

Due Monday 18 October, 2021 at the start of class

Please read Lectures 9, 10, 11, and 12 in the textbook *Numerical Linear Algebra* by Trefethen and Bau. Then do the following exercises.

P14. Equation (10.1) on page 77 is a cartoon of how Householder triangularization works on a 5×3 matrix. Turn this cartoon into a specific calculation by showing the stages A, $Q_1 A$, $Q_2 Q_1 A$, and $Q_3 Q_2 Q_1 A = R$ on the following matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & -2 \\ 1 & 3 & 5 \\ 4 & 5 & 6 \\ 3 & -3 & 3 \end{bmatrix}$$

(*Hints on implementation.* Show the stages by rounding each number at the fourth digit, for example. There is no need to compute the matrices Q_i themselves, or to display them. A small modification of `house.m` in Exercise 10.2 below will generate the stages. This problem encourages you to think through Exercise 10.2 more carefully!)

P15. *The Matlab built-in* `qr()` *computes the QR factorization using Householder transformations as described in Lecture 10. This problem asks you to go ahead and use it! While Lecture 10, and my in-class lecture, has shown how to use QR to solve linear systems, the purpose of* this *problem is to show that QR has a completely different purpose. For more, see Lectures 24–29.*

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

(b) At the Matlab/Octave command line, try the following:

```
>> A = rand(5,5);  A = A + A';    % create a random 5x5 symmetric matrix
>> A0 = A;                        % save the original A
>> [Q, R] = qr(A);  A = R * Q
>> [Q, R] = qr(A);  A = R * Q
...
>> [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 A_i is factored

$$A_i = Q_i R_i$$

and then the next matrix A_{i+1} is generated by multiplying-back in the other order:

$$A_{i+1} = R_i Q_i.$$

Watch what happens to the matrix entries when you iterate at least 10 times. (*Use a* `for` *loop to see a strong effect from e.g. 100 iterations.*) What do you observe about this sequence of A_i? Now compare `sort(diag(A))` to `sort(eig(A0))`.

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

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

This shows A_{i+1} has exactly the same eigenvalues as A_i; explain.

(d) Write a few sentences which relate parts **(a)** and **(b)**. (*Hint. Try to relate the two parts by yourself first. Then read Lecture 25 to either confirm your understanding or, if needed, help you do this part.*)

P16. *Either* by using the built-in functions `polyfit()` and `polyval()`, *or* by setting-up linear systems and solving using Matlab's backslash command, reproduce Figures 11.1 and 11.2. Please make at least modest effort to duplicate the appearance of these Figures. (*Note that* `axis off` *will generate a clean picture without unnecessary ticks and axes labels, and such. But then you might want to put back the axes themselves using* `plot([-6 6],[0 0],'k')` *and similar commands.*)

P17. *This problem is a version of Exercise 11.2 (a), which was done in class.*

How closely, as measured in the L^2 norm on the interval $[1, 2]$, can the function $f(x) = x^{-1}$ be fitted by a linear combination of the functions e^x, $\cos(x)$, and $\sqrt{x}$? Write a program that determines the answer to at least two digits of relative accuracy using a discretization of $[1, 2]$ and a discrete least squares problem. Write down your estimate of the error and also of the coefficients of the optimal linear combination, and produce a plot showing both $f(x)$ and the optimal approximation.

Exercise 10.1.

Exercise 10.2.

Exercise 11.3.