Math 5467 – Homework 1

Instructions:

- Complete the problems below, and submit your solutions and Python code by uploading them to the Google form for HW1.
- If you use LaTeX to write up your solutions, upload them as a pdf file. Students who use LaTeX to write up their solutions will receive bonus points on the homework assignment (equivalent to 1/3 of a letter grade bump).
- If you choose to write your solutions and scan them, please either use a real scanner, or use a smartphone app that allows scanning with you smartphone camera. It is not acceptable to submit images of your solutions, as these can be hard to read.

Problems:

1. Register for the Piazza site `https://piazza.com/umn/spring2022/math5467` and make a post to the whole class. Tell us something about yourself (e.g., is there something new you learned to do during the pandemic), and let us know if there is something specific you would like to learn in the course.

2. Write a Python function that computes the square root of a positive number using the Babylonian method. The Babylonian method to compute $\sqrt{S}$ for $S > 0$ constructs the sequence x_n by setting $x_0 = S$ and iterating

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{S}{x_n}\right).$$

   Your code can take as input a tolerance parameter $\varepsilon > 0$, and should iterate until $|x_n^2 - S| \leq \varepsilon$, and then return x_n. Test your square root function to make sure it works.

   Your function should use only basic Python programming. In particular, do not use any packages, like Numpy, Scipy, etc.

3. Prove that the iteration in the Babylonian method above converges quadratically to the square root of S. In particular, show that the error $\varepsilon_n = \frac{x_n}{\sqrt{S}} - 1$ satisfies

$$\varepsilon_{n+1} = \frac{\varepsilon_n^2}{2(\varepsilon_n + 1)}.$$

   From this, we get that $\varepsilon_n \geq 0$ for $n \geq 1$, and so

$$\varepsilon_{n+1} \leq \frac{1}{2}\min\{\varepsilon_n^2, \varepsilon_n\}.$$

   Show that this implies that $\varepsilon_n < 1$ and $\varepsilon_{n+1} \leq \varepsilon_n^2$ for n sufficiently large, which is exactly quadratic convergence.

4. Write a Python function that computes the largest magnitude eigenvalue of a square matrix with the power iteration. The power iteration is

$$x_{n+1} = \frac{Ax_n}{\|Ax_n\|}.$$

For a diagonalizable matrix, the iteration converges to the eigenvector of A with largest magnitude eigenvalue. The eigenvalue is

$$\lambda = \lim_{n\to\infty} x_n^T A x_n.$$

Compare your function to the true eigenvector and eigenvalue for small matrices where you can compute it by hand, to check that your function works. You can either run your power iteration for a fixed (and large) number of iterations, or you can compute the residual vector

$$r_n = Ax_n - (x_n^T A x_n)x_n$$

and run the iterations until $\|r_n\| \leq \varepsilon$, where $\varepsilon > 0$ is a given tolerance parameter (which can be an argument to your function).

In this exercise you may use Numpy. Try to write your code with only one loop, over the iterations in the power method.

5. Let A be a symmetric matrix. Show that any minimizer x of

$$\min\{x^T A x \,:\, \|x\| = 1\} \tag{1}$$

is an eigenvector of A with eigenvalue $\lambda = x^T A x$, *without* using that A is diagonalizable. [Hint: Any minimizer of (1) is also a minimizer of the Rayleigh quotient

$$f(x) = \frac{x^T A x}{x^T x}$$

over $x \in \mathbb{R}^n$. Compute $\nabla f(x)$, set $\nabla f(x) = 0$, and use that $\|x\| = 1$.]

6. Suppose that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, and let $a_1, \ldots, a_n$ satisfy $0 \leq a_i \leq 1$ for all i, and

$$\sum_{i=1}^{n} a_i = k,$$

where k is an integer and $1 \leq k \leq n$. Show that

$$\sum_{i=1}^{n} \lambda_i a_i \leq \sum_{i=1}^{k} \lambda_i.$$