Spring 2022: Numerical Analysis
Assignment 5 (due April 18, 2022 at 11:59pm ET)

1. **[Gerschgorin, 2+1+1pt]** Gerschgorin's second theorem states that if the union of k Gerschgorin discs is disjoint from the other $n-k$ discs, it must contain exactly k eigenvalues. Now let

$$A = \begin{bmatrix} 1 & 0 & 1 \\ \text{-}1-i & \text{-}3 & 0 \\ 0 & 2i & z \end{bmatrix}$$

for some $z \in \mathbb{C}$. Here i is the imaginary unit.

   (a) Sketch the first two Gerschgorin discs for A.

   (b) Suppose we know that at least two of the three eigenvalues are equal. Using Gerschgorin's theorems, what can we conclude about the value of z? (Find the largest subset of $\mathbb{C}$ that you know z cannot be in)

   (c) Suppose we know that all eigenvalues are equal. What can we conclude about z?

2. **[Power method and inverse iteration, 2+2+2+2pt]** Power Method and Inverse Iteration.

   (a) Implement the Power Method for an arbitrary symmetric matrix $A \in \mathbb{R}^{n\times n}$ and an initial vector $\boldsymbol{x}_0 \in \mathbb{R}^n$.

   (b) Use your code to find an eigenvector of

$$A = \begin{bmatrix} \text{-}2 & 1 & 4 \\ 1 & 1 & 1 \\ 4 & 1 & \text{-}2 \end{bmatrix},$$

   starting with $x_0 = (1, 2, -1)^T$ and $x_0 = (1, 2, 1)^T$. Report the first 5 iterates for each of the two initial vectors. Then use the build-in function in MATLAB/Python/Julia (e.g., `eig(A)` in MATLAB) to examine the eigenvalues and eigenvectors of A. Where do the sequences converge to? Why do the limits not seem to be the same?

   (c) Implement the Inverse Power Method for an arbitrary matrix $A \in \mathbb{R}^{n\times n}$, an initial vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ and an initial eigenvalue guess $\theta \in \mathbb{R}$.

   (d) Use your code from (c) to calculate *all* eigenvectors of A. You may pick appropriate values for θ and the initial vector as you wish (obviously not the eigenvectors themselves). Always report the first 5 iterates and explain where the sequence converges to and why.

   Please also hand in your code.

3. **[Tridiagonalization with Householder, 4pt]** Use Householder matrices to transform the matrix A into tridiagonal form, i.e., find an orthogonal matrix Q such that $T = QAQ^T$ is tridiagonal.

$$A = \begin{bmatrix} 2 & 1 & 2 & 2 \\ 1 & \text{-}7 & 6 & 5 \\ 2 & 6 & 2 & \text{-}5 \\ 2 & 5 & \text{-}5 & 1 \end{bmatrix}.$$

4. **[QR-algorithm, 1+1+2+2pts]** Let A be a symmetric, tridiagonal matrix. You learned that the matrices A_k defined by the QR-algorithm converge to a diagonal matrix that is similar to (and thus has the same eigenvalues as) A. The convergence speed depends on the absolute value of the ratio of consecutive eigenvalues. Let $r \in (0,1)$ and

$$A = \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}$$

   (a) Calculate the eigenvalues of A as a function of r (by hand).

   (b) Implement the QR-algorithm using MATLAB's (or Python's) implementation of the QR-factorization, `qr()`. Your code should run for a quadratic matrix of any size.

   (c) Now define a tolerance, e.g., $\tau = 10^{-10}$. Introduce a stopping criterion in your code, causing it to stop when the maximal difference between the true eigenvalues of A and the diagonal entries of A_k is smaller than τ.[1]

   (d) Use your code with the matrix given for at least five values of $r \in (0,1)$ and make a plot with r versus the number of iterations needed to achieve the given tolerance. Explain your findings by examining the ratio between the eigenvalues of A using (a).

   Please also hand in your code.

5. **[Google and eigenvectors, extra credit, 2+2+2pts]** The Google page rank algorithm, which is responsible for providing ordering search results, has a lot to do with the eigenvector corresponding to the largest eigenvalue of a so-called stochastic matrix, which describes the links between websites. Before working on this question, read the SIAM Review paper on the Linear Algebra behind Google.[2] Stochastic matrices have non-negative entries and each column sums to 1, and one can show (under a few technical assumptions) that it has the eigenvalues $\lambda_1 = 1 > |\lambda_2| \geq \ldots \geq |\lambda_n|$. Thus, we can use the power method[3] to find the eigenvector $\boldsymbol{v}$ corresponding to λ_1, which can be shown to have either all negative or all positive entries. These entries can be interpreted as the importance of individual websites.

   Let us construct a large stochastic matrices (pick a size $n \geq 100$, the size of our "toy internet") in MATLAB as follows:

```
I = eye(n);
A = 0.5*I(randperm(n),:)  + (max(2,randn(n,n))-2);
A = A - diag(diag(A));
L = A*diag(1./(max(1e-10,sum(A,1))));
```

   (a) Plot the sparsity structure of L (i.e., the nonzero entries in the matrix) using the command `spy`. Each non-zero entry corresponds to a link between two websites.

   (b) Plot the (complex) eigenvalues of L by plotting the real part of the eigenvalues on the x-axis, and the imaginary part on the y-axis.[4] Additionally, plot the unit circle and check that all eigenvalues are inside the unit circle, but $\lambda_1 = 1$.

   (c) The matrix L contains many zeros. One of the technical assumptions for proving theorems is that all entries in L are positive. As a remedy, one considers the matrices $S = \kappa L + (1 -$

[1] You might want to sort the true and numerically computed eigenvalues before comparing them using `sort`.

[2] *The 25,000,000,000 eigenvector. The linear algebra behind Google* by Kurt Bryan and Tanya Leise. It's easy to find—just google it!

[3] We have discussed the power method for symmetric matrices, but it also works for non-symmetric matrices.

[4] Please make sure that the plotted eigenvalues are not connected by lines–that's confusing.

$\kappa)E$, where E is a matrix with entries $1/n$ in every component[5]. Study the influence of κ numerically by visualizing the eigenvalues of S for different values of κ. Why will $\kappa < 1$ improve the speed of convergence of the power method?

Please also hand in your code and output.

[5] In the original Brin/Page Google paper, the authors use $\kappa = 0.85$. The introduction of the matrix E makes each matrix entry positive and also helps dealing with web pages without outgoing links, which lead to zero columns in L.