Fall 2022: Numerical Analysis
Assignment 5 (due Nov 29, 2022 at 11:59pm ET)

2 extra credit points will again be given for generally cleanly written and reasonably well-organized homework. This includes cleanly plotted and labeled figures (see also rules on the first assignment).

1. **[Gerschgorin warmup, 4pts]** Consider the matrix

$$A := \begin{pmatrix} -6 & 2 & 0.3 & 0 & -0.7 \\ 2 & -4 & 0.1 & 0.05 & 0 \\ 0.3 & 0.1 & 2 & 0.1 & 0.1 \\ 0 & 0.05 & 0.1 & 4 & 0 \\ -0.7 & 0 & 0.1 & 0 & 6 \end{pmatrix}$$

   a) Argue that all eigenvalues of A are real.
   b) What are the Gerschgorin disks for A? Use them to give a set, $D \subset \mathbb{R}$, that contains all eigenvalues of A.
   c) Based on b), can you conclude that the eigenvalue with the largest absolute value is simple?
   d) Using the result in (b), argue that A is invertible.

2. **[More 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

$$A := \begin{bmatrix} 1 & 0 & 1 \\ -1-i & -3 & 0 \\ 0 & 2i & z \end{bmatrix} \in \mathbb{C}^{3\times 3}$$

   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?

3. **[Power method and inverse iteration, 2+2+2+2pt]**

   (a) Implement the Power Method for an arbitrary 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

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

   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 a build-in eigenvalue solver (e.g., MATLAB's `eig(A)`) 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 and output.

4. **[Power method meets Gerschgorin, 2+2+2pt]** Let $A \in \mathbb{R}^{n\times n}$ be symmetric with eigenvalues

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|,$$

and denote the corresponding eigenvectors by $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_n$. We consider the power method for finding the eigenvector corresponding to the dominant (i.e., largest in absolute value) eigenvalue. Given an initialization $\boldsymbol{x}^0$ not orthogonal to $\boldsymbol{\eta}_1$, we compute $\boldsymbol{x}^{k+1} := A\boldsymbol{x}^k$ for $k = 0, 1, \ldots$. The Rayleigh quotient, an approximation to the corresponding eigenvalue, is given by

$$r_k := \frac{(\boldsymbol{x}^k)^T A \boldsymbol{x}^k}{(\boldsymbol{x}^k)^T \boldsymbol{x}^k} \tag{1}$$

(a) Show that

$$r_k = \lambda_1[1 + a_k],$$

where a_k satisfies $a_k(\lambda_2/\lambda_1)^{-2k} \to C$ as $k \to \infty$, where C is a constant that does not depend on k. Hint: Simply use the form of the $\boldsymbol{x}^k$ we found in the proof of the theorem in class.

(b) Consider a symmetric matrix $A \in \mathbb{R}^{5\times 5}$

$$A = A^T = \begin{bmatrix} -9 & * & * & * & * \\ * & 0 & * & * & * \\ * & * & 1 & * & * \\ * & * & * & 4 & * \\ * & * & * & * & 21 \end{bmatrix},$$

where $*$ represents elements of absolute values $\leq 1/4$. Suppose the power method is applied with A and the initial vector $\boldsymbol{x}^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$. Show that $\boldsymbol{x}^0$ is an "appropriate" initial vector, i.e., that the sequence $\boldsymbol{y}^k = \boldsymbol{x}^k/\|\boldsymbol{x}^k\|$ does converge toward the eigenvector belonging to the dominant eigenvalue of A.[1]

(c) Estimate how many correct digits r_{k+5} gains compared to r_k, with r_k as defined in (1).

5. **[Root finding by solving an eigenvalue problem, 3+3pts]** An efficient way to find individual roots of a polynomial is to use Newton's method. However, as we have seen, Newton's method requires an initialization close to the root one wants to find, and it can be difficult to find *all* roots of a polynomial. Luckily, one can use the relation between eigenvalues and polynomial roots to find all roots of a given polynomial. Let us consider a polynomial of degree n with leading coefficient 1:

$$p(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} + x^n \quad \text{with } a_i \in \mathbb{R}.$$

[1]This and the next problem require basic estimation of the eigenvalues of A, for which you can use Gerschgorin theorems.

(a) Show that $p(x)$ is the characteristic polynomial of the matrix (sometimes called a companion matrix for p)

$$A_p := \begin{bmatrix} 0 & & & -a_0 \\ 1 & & & -a_1 \\ & \ddots & & \vdots \\ & & 1 & -a_{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Thus, the roots of $p(x)$ can be computed as the eigenvalues of A_p using, e.g., MATLAB's `eig` function.

(b) Let us consider Wilkinson's polynomial $p_w(x)$ of order 15, i.e., a polynomial with the roots $1, 2, \ldots, 15$:

$$p_w(x) = (x-1) \cdot (x-2) \cdot \ldots \cdot (x-15).$$

The corresponding coefficients can be found using the `poly()` function. Use these coefficients in the matrix A_p to find the original roots again, and compute their error. Compare with the build-in method (called `roots()`) for finding the roots of a polynomial.[2]

Please also hand in your code and output.

6. **[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.

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

7. **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.[3] 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[4] 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))));
```

[2] Note that for MATLAB functions that do not use external libraries, you can see how they are implemented by typing `edit name_of_function`. Doing that for the `roots` function will show you that MATLAB implements root finding exactly using a companion matrix as described above.

[3] *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!

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

(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.[5] 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 $S = \kappa L + (1 - \kappa)E$, where E is a matrix with entries $1/n$ in every component[6]. 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.

8. **[Data compression with SVD, extra credit, 2+3pts]** Recall that for a matrix $A \in \mathbb{R}^{m \times n}$, the singular value decomposition is $A = U \Sigma V^T$ with $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ orthogonal, and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n}$ diagonal with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \geq 0$.

   (a) Show that the Frobenius norm of A, i.e., $\|A\|_F := \left(\sum_{i,j} |a_{ij}|^2 \right)$ can be computed as
   $$\|A\|_F = \sqrt{\sigma_1^2 + \ldots + \sigma_p^2}.$$
   Hints: (1) Show that for any matrix A, the Frobenius norm satisfies $\|A\|_F^2 = \operatorname{tr}(A^T A)$, where tr denotes the trace of a matrix, i.e., the sum of its diagonal entries. (2) Use that the trace satisfies a cyclic property, i.e., $\operatorname{tr}(ABC) = \operatorname{tr}(BCA) = \operatorname{tr}(CAB)$ for all matrices A, B, C where the multiplication is well-defined. (You don't have to prove this!).

   (b) Take your favorite black and white photo, consider the gray values of the pixels as a matrix, and compute the SVD of that matrix. Then, compress the matrix by only retaining the singular vectors corresponding to the largest few singular values (i.e., by zeroing out small singular values)[7]. What percentage of data of the original image data is necessary to obtain a reasonable image reconstruction? How do you think does this depend on the image? Show images with different compression levels.

9. **[Space of polynomials P_n, 1+2+2pts]** Let P_n be the space of functions defined on $[-1, 1]$ that can be described by polynomials of degree less or equal to n with coefficients in $\mathbb{R}$. P_n is a linear space in the sense of linear algebra, in particular, for $p, q \in P_n$ and $a \in \mathbb{R}$, also $p + q$ and ap are in P_n. Since the monomials $\{1, x, x^2, \ldots, x^n\}$ are a basis for P_n, the dimension of that space is $n + 1$.

   (a) Show that for pairwise distinct points $x_0, x_1, \ldots, x_n \in [-1, 1]$, the Lagrange polynomials $L_k(x)$ are in P_n, and that they are linearly independent, that is, for a linear combination of the zero polynomial with Lagrange polynomials with coefficients α_k, i.e.,
   $$\sum_{k=0}^{n} \alpha_k L_k(x) = 0 \text{ (the zero polynomial)}$$

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

[6] 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.

[7] Many examples of this can be found on the web, see for instance http://math.arizona.edu/~brio/VIGRE/ThursdayTalk.pdf.

necessarily follows that $\alpha_0 = \alpha_1 = \ldots = \alpha_n = 0$. Note that this implies that the $(n+1)$ Lagrange polynomials also form a basis of P_n.

(b) Since both the monomials and the Lagrange polynomials are a basis of P_n, each $p \in P_n$ can be written as linear combination of monomials as well as Lagrange polynomials, i.e.,

$$p(x) = \sum_{k=0}^{n} \alpha_k L_k(x) = \sum_{k=0}^{n} \beta_k x^k, \tag{2}$$

with appropriate coefficients $\alpha_k, \beta_k \in \mathbb{R}$. As you know from basic matrix theory, there exists a basis transformation matrix that converts the coefficients $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_n)^T$ to the coefficients $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_n)^T$. Show that this basis transformation matrix is given by the so-called Vandermonde matrix $V \in \mathbb{R}^{n+1\times n+1}$ given by

$$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} & x_n^n \end{pmatrix},$$

i.e., the relation between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (2) is given by $\boldsymbol{\alpha} = V\boldsymbol{\beta}$. An easy way to see this is to choose appropriate x in (2).

(c) Note that since V transforms one basis into another basis, it must be an invertible matrix. Let us compute the condition number of V numerically.[8] Compute the 2-based condition number $\kappa_2(V)$ for $n = 5, 10, 20, 30$ with uniformly spaced nodes $x_i = -1 + (2i)/n$, $i = 0, \ldots, n$. Based on the condition numbers, can this basis transformation be performed accurately?

[8] MATLAB provides the function `vander`, which can be used to assemble V (actually, the transpose of V). Alternatively, one can use a simple loop to construct V.