Spring 2019: Numerical Analysis
Assignment 4 (due April. 18, 2019)

1. **[2+2+2+2pt]** Power Method and Inverse Iteration.

   (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 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.

2. **[2+2+2pts]** Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. The Rayleigh quotient is an important function in numerical linear algebra, defined as:

   $$r(\boldsymbol{x}) = \frac{\boldsymbol{x}^T A\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}.$$

   (a) Show that $\lambda_{\text{min}} \leq r(\boldsymbol{x}) \leq \lambda_{\text{max}} \quad \forall \boldsymbol{x} \in \mathbb{R}^n$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of A respectively.

   (b) We needed to use the gradient of $r(\boldsymbol{x})$ in the analysis of the power method. Compute the gradient of $r(\boldsymbol{x})$. Recall how we computed the gradient of a function for deriving the normal equations to help you with this.

   (c) The Rayleigh quotient iteration is a modification to the inverse power method described in problem 1. The shift is modified at every iteration, i.e. given $\boldsymbol{v}^{(0)}$ so that $\|\boldsymbol{v}^{(0)}\| = 1$, for $k = 1, 2, 3, \ldots$

   - set $\theta^{(k-1)} = r(\boldsymbol{v}^{(k-1)})$.
   - solve $(A - \theta^{(k-1)})\,\boldsymbol{w} = \boldsymbol{v}^{(k-1)}$.
   - normalize $\boldsymbol{v}^{(k)} = \boldsymbol{w}/\|\boldsymbol{w}\|$.

Modify the inverse power method code from problem 1 to implement the Rayleigh quotient iteration. Compare the convergence speed of this approach to the plain vanilla inverse power method.

Please also hand in your code.

3. **[2+2+2+2pts]** Issues with the power method and Givens rotations. For a given angle φ, consider the *Givens rotation* matrix defined as:

$$G(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$$

   (a) Show that $G(\varphi)G(\varphi)^T = G(\varphi)^T G(\varphi) = I$.

   (b) What does the matrix $G(\varphi)$ do geometrically? You can see this by applying the matrix to a standard unit vector.

   (c) Show that the eigenvalues of $G(\varphi)$ have magnitude 1. Consider an arbitrary eigenvalue/eigenvector pair $(\boldsymbol{x}, \lambda)$, with $\boldsymbol{x}$ normalized, and compute $|\lambda|^2 = \|\lambda\boldsymbol{x}\|_2^2 = \ldots$.

   (d) How does the power method iteration behave on the matrix $G(\varphi)$? Explain this behavior using your answer in part (c).

4. **[2+2+4pts]** Orthogonalization methods.

   (a) Given any two nonzero vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^n$, construct a Householder matrix H, such that $H\boldsymbol{x}$ is a scalar multiple of $\boldsymbol{y}$. Is the matrix H unique?

   (b) Use Givens rotations to transform the vector

$$\boldsymbol{x} = \begin{bmatrix} 4 \\ \text{-}3 \\ 1 \end{bmatrix}$$

   to a multiple of the first unit vector. Specify the Givens rotations you used.

   (c) Use Householder matrices to compute the QR-factorization of the matrix

$$\begin{bmatrix} 9 & -6 \\ 12 & -8 \\ 0 & 20 \end{bmatrix}.$$

   Now use Givens rotations to compute the QR-factorization of the above matrix. Based on your results from using Givens rotations or Householder matrices, make a guess about the uniqueness of the QR factorization.

5. **[2+1+1pts]** 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?

6. **[2+2pts]** 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:

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

(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 the QR algorithm (as implemented, e.g., in 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.[1]

Please also hand in your code.

7. **Bonus [3+3+4pts]** Before answering this question, read the Google page rank article on Piazza in the 'General Resources' section. The Google page rank algorithm 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.[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:

[1] 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.

[2] See the interesting 2006 SIREV paper *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.

```
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 $S = \kappa L + (1 - \kappa)E$, 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.

[4] Please make sure that the plotted eigenvalues are not connected.

[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.