CSE 6643 Homework 5

Karl Hiner, Spring 2023

1 Power Method [30 pts]

We consider a matrix $\boldsymbol{A}$ such that

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q} = \operatorname{diag}(\lambda_1, \dots, \lambda_m), \tag{1}$$

where $\boldsymbol{Q}$ is orthogonal. We denote by $\boldsymbol{q}_i$ the ith column of $\boldsymbol{Q}$.

We now consider the power method and the sequence $\boldsymbol{\nu}^{(k)}$ defined as

$$\begin{aligned} \boldsymbol{z}^{(k)} &= \boldsymbol{A}\boldsymbol{\nu}^{(k-1)} && (2)\\ \boldsymbol{\nu}^{(k)} &= \boldsymbol{z}^{(k)}/\|\boldsymbol{z}^{(k)}\|_2, && (3) \end{aligned}$$

where we assume that $\|\boldsymbol{\nu}^{(0)}\|_2 = 1$. Assume that we have θ_k such that

$$\cos(\theta_k) = \boldsymbol{q}_1^T\boldsymbol{\nu}^{(k)} \tag{4}$$

with $\cos(\theta_0) \neq 0$. Prove that

$$1 - \cos(\theta_k)^2 \leq \frac{1}{a_1^2}\sum_{i=2}^{m} a_i^2\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}, \quad a_i = \boldsymbol{q}_i^T\boldsymbol{\nu}^{(0)} \tag{5}$$

$$
\begin{aligned}
\cos(\theta_k) &= \boldsymbol{q}_1^T \boldsymbol{\nu}^{(k)} && \text{(given)} \\
&= \boldsymbol{q}_1^T \frac{\boldsymbol{A}^k \boldsymbol{\nu}^{(0)}}{\|\boldsymbol{A}^k \boldsymbol{\nu}^{(0)}\|_2} && \text{(def. of power method)} \\
&= \frac{\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{Q}^T \boldsymbol{\nu}^{(0)}}{\|\boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{Q}^T \boldsymbol{\nu}^{(0)}\|_2} && \text{(def. of } \boldsymbol{A}\text{)} \\
&= \frac{\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}}{\|\boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\|_2} && (\boldsymbol{a} \equiv \boldsymbol{Q}^T \boldsymbol{\nu}^{(0)} \to a_i = \boldsymbol{q}_i^T \boldsymbol{\nu}^{(0)}) \\
\cos(\theta_k)^2 &= \frac{\left(\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^2}{\|\boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\|_2^2} && \text{(square both sides)} \\
&= \frac{\left(\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^2}{\left(\boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^T \left(\boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)} && \text{(def. of 2-norm)} \\
&= \frac{\left(\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^2}{\boldsymbol{a}^T \boldsymbol{\Lambda}^k \boldsymbol{Q}^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}} && \text{(apply transpose in den.)} \\
&= \frac{\left(\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^2}{\boldsymbol{\Lambda}^{2k} \boldsymbol{a}^T \boldsymbol{a}} && \text{(simplify den.)} \\
&= \frac{\left(\boldsymbol{q}_1^T \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^2}{\sum\limits_{i=1}^{m} \lambda_i^{2k} a_i^2} && \text{(expand den.)} \\
&= \frac{\left(\boldsymbol{e}_1 \boldsymbol{\Lambda}^k \boldsymbol{a}\right)^2}{\sum\limits_{i=1}^{m} \lambda_i^{2k} a_i^2} && (\boldsymbol{q}_1^T \cdot \boldsymbol{q}_1 = 1,\ \boldsymbol{q}_1^T \cdot \boldsymbol{q}_{i \neq 1} = 0) \\
&= \frac{\lambda_1^{2k} a_1^2}{\sum\limits_{i=1}^{m} \lambda_i^{2k} a_i^2} && \text{(evaluate num.)} \\
1 - \cos(\theta_k)^2 &= 1 - \frac{\lambda_1^{2k} a_1^2}{\sum\limits_{i=1}^{m} \lambda_i^{2k} a_i^2} && \text{(negate \& add 1 to both sides)} \\
&= \frac{\sum\limits_{i=1}^{m} \lambda_i^{2k} a_i^2 - \lambda_1^{2k} a_1^2}{\sum\limits_{i=1}^{m} \lambda_i^{2k} a_i^2} && \text{(common den.)} \\
&= \frac{\sum\limits_{i=2}^{m} \lambda_i^{2k} a_i^2}{\lambda_1^{2k} a_1^2 + \sum\limits_{i=2}^{m} \lambda_i^{2k} a_i^2} && \text{(rearrange summation terms)} \\
&\leq \frac{\sum\limits_{i=2}^{m} \lambda_i^{2k} a_i^2}{\lambda_1^{2k} a_1^2} = \frac{1}{a_1^2} \sum_{i=2}^{m} a_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2k} && \text{(QED)}
\end{aligned}
$$

2 The LU iteration algorithm [30 pts]

We consider the following iteration, starting with some full rank $\boldsymbol{G}_0 \in \mathbb{C}^{m \times m}$:

$$\boldsymbol{Z}_k = \boldsymbol{A}\boldsymbol{G}_{k-1}, \tag{6}$$

$$\boldsymbol{G}_k\boldsymbol{R}_k = \boldsymbol{Z}_k, \quad \text{LU factorization with no pivoting,} \tag{7}$$

where $\boldsymbol{G}_k$ is lower-triangular and $\boldsymbol{R}_k$ is upper-triangular. We define

$$\boldsymbol{T}_k = \boldsymbol{G}_k^{-1}\boldsymbol{A}\boldsymbol{G}_k. \tag{8}$$

Find an algorithm that computes the sequence $\boldsymbol{T}_k$ in a manner similar to the QR iteration. This algorithm is a variant of the QR iteration. The eigenvalues of $\boldsymbol{A}$ appear on the diagonal of $\boldsymbol{T}_k$.

From Piazza:

You just need to show your algorithm and are not required to prove the algorithm reveals the eigenvalues on the diagonal of $\boldsymbol{T}_k$. It is sufficient to use the given equations to derive a sequence of $\boldsymbol{T}_k$. You need to find a manner similar to the QR iteration to compute $\boldsymbol{T}_k$. That is, the computation of $\boldsymbol{T}_k$ should be similar to $\boldsymbol{R}_k\boldsymbol{G}_k$. Note that here $\boldsymbol{T}_k \neq \boldsymbol{R}_k\boldsymbol{G}_k$, but you will find that when changing the formula a little bit, e.g, inserting a matrix between $\boldsymbol{R}_k$ and $\boldsymbol{G}_k$, it will make the equality hold. This modification should not bring much additional computational cost.

This algorithm consists of iteratively doing matrices multiplication, applying LU factorization, and then doing two different matrices multiplications to compute $\boldsymbol{T}_k$. Its relationship to QR factorization only concerns the manner of computation.

Goal: Find a matrix $\boldsymbol{B}$ s.t. $\boldsymbol{T}_k = \boldsymbol{G}_k^{-1}\boldsymbol{A}\boldsymbol{G}_k = \boldsymbol{R}_k\boldsymbol{B}\boldsymbol{G}_k$.

The QR Iteration algorithm:

- $\boldsymbol{T}_{k-1} = \boldsymbol{U}_k\boldsymbol{R}_k$
- $\boldsymbol{T}_k = \boldsymbol{R}_k\boldsymbol{U}_k$

Answer:

We insert the matrix $\boldsymbol{G}_k^{-1}$ between $\boldsymbol{R}_k$ and $\boldsymbol{G}_k$. The algorithm involves matrix multiplication, LU factorization, and then computing $\boldsymbol{T}_k$ using the inverse of the lower triangular matrix $\boldsymbol{G}_k$.

To avoid the direct computation of the inverse, we can solve a linear system, as solving a linear system with a triangular matrix is more efficient:

1. Start with the initial full rank $\boldsymbol{G}_0 \in \mathbb{C}^{m \times m}$.
2. For each iteration k:
   (a) Compute $\boldsymbol{Z}_k = \boldsymbol{A}\boldsymbol{G}_{k-1}$.
   (b) Perform LU factorization with no pivoting: $\boldsymbol{G}_k\boldsymbol{R}_k = \boldsymbol{Z}_k$.
   (c) Solve the linear system $\boldsymbol{G}_k\boldsymbol{T}_k = \boldsymbol{Z}_k$ for $\boldsymbol{T}_k$ using forward or backward substitution.

3 Convergence of Orthogonal Iteration [40 pts]

In this problem, you will explore the convergence behavior of the orthogonal iteration algorithm. In the problems below, only the asymptotic convergence will match the theoretical estimates. For small k, you may see deviations. To simplify grading, please use the provided (empty) file `HW4_your_code.jl` for your code. Please also attach all results to your report, both plots and print statements. The TAs should be able to grade your homework without running your code.

(a) [10 pts]

Write Julia code to create a matrix in $\mathbb{R}^{m \times m}$ of size $m = 8$ with eigenvalues $1, 3, 9, \ldots, 3^{m-1}$. Explain your code (as comments *and* in your report) and describe what it does.

The method `random_matrix(m, Λ)` performs the following to produce a matrix in $\mathbb{R}^{m \times m}$ with eigenvalues Λ using the following algorithm:

1. Create a random matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ of normally distributed values with $\mu = 0$ and $\sigma = 1$.
2. Perform a QR decomposition of $\boldsymbol{A}$ to create an orthogonal matrix $\boldsymbol{Q}$.
3. Return $\boldsymbol{Q}$ $\operatorname{diag}(\boldsymbol{\Lambda})$ $\boldsymbol{Q}^T$.

(b) [10 pts]

Implement the orthogonal iteration algorithm. Print the values along the diagonal of $\boldsymbol{R}_k$ at each iteration k for $k = 1, \ldots, 5$. Print each number using at most four significant digits.

```
$ julia --project=. HW5_your_code.jl
[-152.1, -555.1, -140.0, -47.68, -215.6, -3.844, -10.86, 4.508]
[1888.0, 639.5, 274.7, 62.78, 40.42, 3.566, 6.945, 1.098]
[2177.0, 716.7, 247.8, -76.64, -28.58, 6.24, 4.286, 1.01]
[2186.0, 727.7, 243.5, 80.46, 27.18, -8.452, -3.191, 1.001]
[2187.0, 728.9, 243.1, 80.94, 27.02, 8.933, 3.022, 1.0]
```

(c) [10 pts]

Considering entry p along the diagonal, plot the convergence of the pth eigenvalue. Choose $p = 1$, 2, and 3. Compare with the theoretical rate of convergence at step k for entry p, which is given by

$$\max(|\lambda_{p+1}/\lambda_p|^k, |\lambda_p/\lambda_{p-1}|^k), \quad 1 < p < m, \tag{9}$$

$$|\lambda_2/\lambda_1|^k, \quad p = 1 \tag{10}$$

$$|\lambda_m/\lambda_{m-1}|^k, \quad p = m. \tag{11}$$

Use a semi-log plot (meaning that the y-axis of your plot should be logarithmic).

Figure 1 shows plots of the estimated and true eigenvalues, and the actual and theoretical convergence rates for $p = 1, 2, 3$ and $k \in 1, 2, \ldots, 10$. Notice that the actual convergence rates are somewhat faster than the theoretical convergence rates.

Note: Depending on the generated random matrix, the convergence rate of one or more of the eigenvalues may be closer to the theoretical convergence rate.

(d) [10 pts]

Consider the block $(p + 1 : m, 1 : p)$ in the matrix

$$\boldsymbol{A}_k = \boldsymbol{Q}_k^T \boldsymbol{A} \boldsymbol{Q}_k. \tag{12}$$

Plot the 2-norm of this block as a function of k for $p = 4$. Compare with the analytical estimate, which states that it should decay like $|\lambda_{p+1}/\lambda_p|^k$.

Figure 2 shows a plot of the true and analytical 2-norms of these subblocks.

The analytical norm estimates have exactly the same decay rate as the true norm estimates, though the magnitudes are lower.

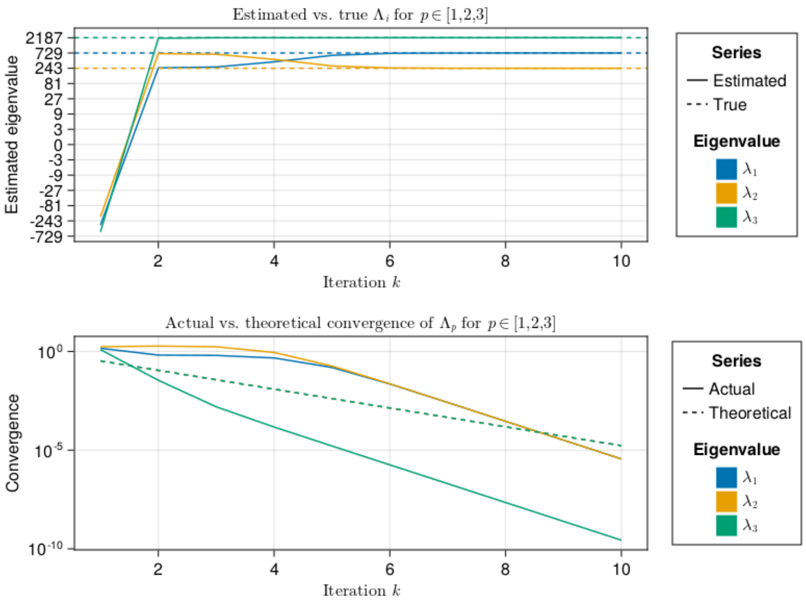

Figure 1 Eigenvalue convergence for $p = 1, 2, 3$

Note: The exact magnitude of the difference in analytic vs true norms is dependent on the generated random matrix.

4 LU and QR iteration [20 bonus pts]

We consider the LU iteration applied to a symmetric matrix. Assume that $\boldsymbol{A}_0 \in \mathbb{R}^{m \times m}$ is symmetric and positive-definite. We produce a sequence of matrices $\boldsymbol{A}_i$ using the following algorithm:

$$\boldsymbol{A}_i = \boldsymbol{G}_i^T \boldsymbol{G}_i, \quad \boldsymbol{A}_{i+1} = \boldsymbol{G}_i \boldsymbol{G}_i^T, \tag{13}$$

where $\boldsymbol{G}_i$ are upper-triangular matrices.

Consider now $\boldsymbol{A}'$, the matrix obtained after one step of the QR iteration, that is,

$$\boldsymbol{A}_0 = \boldsymbol{Q}\boldsymbol{R}, \quad \boldsymbol{A}' = \boldsymbol{R}\boldsymbol{Q}. \tag{14}$$

We assume that the diagonal of $\boldsymbol{R}$ is positive.

(a) [10 bonus pts]

Use $\boldsymbol{A}_0^2$ to show that

$$\boldsymbol{R} = \boldsymbol{G}_1 \boldsymbol{G}_0 \tag{15}$$

(b) [10 bonus pts]

Show that

$$\boldsymbol{A}' = \boldsymbol{A}_2 \tag{16}$$

5 Sensitivity of Eigenvalues [20 bonus pts]

We are interested in determining the sensitivity of eigenvalues with respect to perturbations in the matrix.

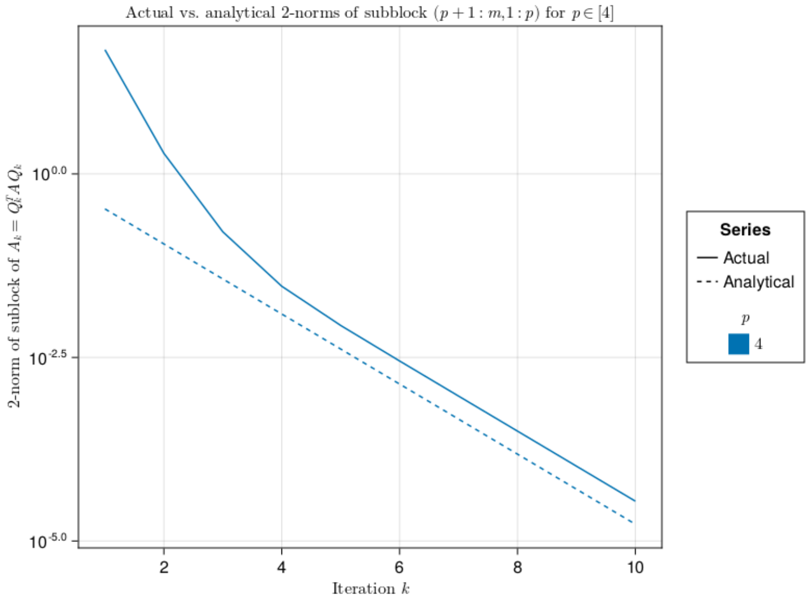

Figure 2 Analytical vs. actual 2-norm decay rates

Prove that if $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ is diagonalizable with $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}$, and if μ is an eigenvalue of $\boldsymbol{A}+\boldsymbol{E}$,

$$\min_{\lambda\in\lambda(\boldsymbol{A})} |\mu - \lambda| \leq \kappa_p(\boldsymbol{V})\|\boldsymbol{E}\|_p. \tag{17}$$

Here $\|\cdot\|_p$ denotes the p-norm and $\kappa_p(\boldsymbol{V}) := \|\boldsymbol{V}\|_p\|\boldsymbol{V}^{-1}\|_p$ is the condition number associated with the p-norm.

Hint: assume that $\mathbf{I}+\boldsymbol{F}$ is singular. Then, there is an $\boldsymbol{x} \neq \mathbf{0}$ such that $\boldsymbol{x} + \boldsymbol{F}\boldsymbol{x} = \mathbf{0}$. Therefore, $\boldsymbol{F}\boldsymbol{x} = -\boldsymbol{x}$ and $\|\boldsymbol{F}\|_p \geq 1$. For the proof, identify the proper singular matrix and use this result.