SI231 - Matrix Computations, Fall 2020-21

Homework Set #4

Prof. Yue Qiu and Prof. Ziping Zhao

Name: Tao Huang **Major:** Undergraduate in CS

Student No.: 2018533172 **E-mail:** huangtao1@shanghaitech.edu.cn

---

Acknowledgements:

1) Deadline: **2020-11-20 23:59:00**
2) Submit your homework at **Gradescope**. Homework #4 contains two parts, the theoretical part the and the programming part.
3) About the the theoretical part:
   (a) Submit your homework in **Homework 4** in gradescope. Make sure that you have correctly select pages for each problem. If not, you probably will get 0 point.
   (b) Your homework should be uploaded in the **PDF** format, and the naming format of the file is not specified.
   (c) No handwritten homework is accepted. You need to use LaTeX in principle.
   (d) Use the given template and give your solution in English. Solution in Chinese is not allowed.
4) About the programming part:
   (a) Submit your codes in **Homework 4 Programming part** in gradescope.
   (b) When handing in your homework in gradescope, package all your codes into your_student_id+hw4_code.zip and upload. In the package, you also need to include a file named README.txt/md to clearly identify the function of each file.
   (c) Make sure that your codes can run and are consistent with your solutions.
5) **Late Policy details can be found in the bulletin board of Blackboard.**

STUDY GUIDE

This homework concerns the following topics:

- Eigenvalues, eigenvectors & eigenspaces
- Algebraic multiplicity & geometric multiplicity
- Eigendecomposition (Eigvenvalue decomposition) & Eigendecomposition for Hermitian matrices
- Similar transformation, Schur decomposition & Diagonolization
- Variational characterizations of eigenvalues
- Power iteration & Inverse iteration
- QR iteration & Hessenberg QR iteration
- Givens QR & Householder QR (from previous lectures)

I. UNDERSTANDING EIGENVALUES AND EIGENVECTORS

Problem 1. (6 points + 4 points)

Consider the 2×2 matrix

$$\mathbf{A} = \begin{bmatrix} -4 & -3 \\ 6 & 5 \end{bmatrix}.$$

1) Determine whether $\mathbf{A}$ can be diagonalized or not. Diagonalize $\mathbf{A}$ by $\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$ if the answer is "yes" or give the reason if the answer is "no".
2) Give the eigenspace of $\mathbf{A}$. And then consider: is there a matrix being similar to $\mathbf{A}$ but have different eigenspaces with it. If the answer is "yes", show an example (here you are supposed to give the specific matrix and its eigenspaces), or else explain why the answer is "no" .

Remarks:

- In 1), if $\mathbf{A}$ can be diagonalized, you are supposed to present not only the specific diagonalized matrix but also how do you get the similarity transformation. If not, you should give the necessary derivations of the specific reason.
- In 2), if your answer is "yes", you are supposed to give the specific matrix and its eigenspaces. If "no", you should give the necessary derivations of the specific reason.

Solution.

1) Yes. We give the derivations in the below.

$$\det(\lambda I - A) = \lambda^2 - \lambda - 2 = (\lambda - 2)(\lambda + 1) \Rightarrow \lambda_1 = 2, \lambda_2 = -1$$

For $\lambda_1 = 2$

$$\begin{bmatrix} 6 & 3 \\ -6 & -3 \end{bmatrix} v = 0 \Rightarrow v_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \text{(we choose this specific vector)}$$

For $\lambda_2 = -1$

$$\begin{bmatrix} 3 & 3 \\ -6 & -6 \end{bmatrix} v = 0 \Rightarrow v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \text{(we choose this specific vector)}$$

So we get

$$V = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix}, A = V\Lambda V^{-1} = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix}^{-1}$$

2) Yes. For matrix B which satisfies

$$B = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -7 & 4.5 \\ -12 & 8 \end{bmatrix}$$

The eigenspaces are

$$\mathcal{S}_1 = span \left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}, \mathcal{S}_2 = span \left\{ \begin{bmatrix} 3 \\ 4 \end{bmatrix} \right\}$$

So there exist a matrix B such that B and A are similar but have different eigenspaces.

Problem 2. (6 points $\times$ 5)

For a matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, $\lambda_1, \lambda_2, \ldots, \lambda_n$ are its n eigenvalues (though some of them may be the same). Prove that:

1) The matrix $\mathbf{A}$ is singular if only if 0 is an eigenvalue of it.
2) $\mathsf{rank}(\mathbf{A}) \geq$ number of nonzero eigenvalues of $\mathbf{A}$.
3) If $\mathbf{A}$ admits an eigendecomposition (eigenvalue decomposition), $\mathsf{rank}(\mathbf{A}) =$ number of nonzero eigenvalues of $\mathbf{A}$.
4) If $\mathbf{A}$ is Hermitian, then all of eigenvalues of $\mathbf{A}$ are real.
5) If $\mathbf{A}$ is Hermitian, then eigenvectors corresponding to different eigenvalues are orthogonal.

Solution

1) By the relation between singularity and eigenvalues, we have

$$A \text{ is singular} \Leftrightarrow \det(A) = 0 \Leftrightarrow \Pi_{i=1}^{n}\lambda_i = 0 \Leftrightarrow \exists\lambda_i = 0.$$

This gives

$$A \text{ is singular} \Leftrightarrow \exists\lambda_i = 0.$$

which proves that the matrix A is singular if only if 0 is an eigenvalue of it.

2) The eigenvectors v corresponds to eigenvalue $\lambda = 0$:

$$(0 \cdot I - A)v = 0 \Leftrightarrow Ax = 0$$

So the geometric multiplicity γ_0 of $\lambda_i = 0$ is equal to $n - rank(A)$. Besides, we know that algebraic multiplicity μ_0 of $\lambda_i = 0$ is larger or equal to γ_0, this gives

$$\mu_0 \geq \gamma_0 = n - rank(A).$$

which implies

$$rank(A) \geq n - \mu_0$$

Also, since the sum of algebraic multiplicity is equal to n (see from characteristic polynomial), we have

$$\sum \mu_i = n \Leftrightarrow \mu_0 + \sum_{\lambda\neq 0} \mu_i = n \Leftrightarrow n - \mu_0 = \sum_{\lambda\neq 0} \mu_i$$

So we have

$$rank(a) \geq n - \mu_0 = \sum_{\lambda\neq 0} \mu_i$$

which means $rank(A) \geq$ number of nonzero eigenvalues of A

3) If A admits an eigendecomposition, the geometric multiplicity γ_0 and algebraic multiplicity μ_0 of eigenvalue $\lambda = 0$ are equal. Inherit from subproblem 2), we have

$$\mu_0 = \gamma_0 = n - rank(A)$$

which implies that

$$rank(A) = n - \mu_0$$

Since $n - \mu_0 = \sum_{\lambda \neq 0} \mu_i$, $rank(A) = \sum_{\lambda \neq 0} \mu_i =$ number of nonzero eigenvalues of A.

4) By the definition of eigenvalues and eigenvectors:

$$\begin{gathered} Ax = \lambda x \\ \Rightarrow x^H A x = x^H \lambda x = \lambda x^H x \end{gathered}$$

And

$$x^H A^H x = (Ax)^H x = \lambda^\star x^H x$$

Since $A = A^H$, these give

$$\lambda x^H x = \lambda^\star x^H x, x \neq 0$$

which implies $\lambda = \lambda^\star, \forall \lambda$ (otherwise the above equality will not hold). Thus, all of eigenvalues of A are real.

5) $\forall \lambda_i \neq \lambda_j$

$$\begin{aligned} v_j^H A v_i &= v_j^H (A v_i) \\ &= \lambda_i v_j^H v_i \\ v_j^H A v_i &= v_j^H A^H v_i \\ &= (A v_j)^H v_i \\ &= \lambda_j^\star v_j^H v_i \\ &= \lambda_j v_j^H v_i \quad \text{(conclusion from subproblem 4))} \end{aligned}$$

which implies

$$(\lambda_i - \lambda_j) v_j^H v_i = 0 \Rightarrow v_j^H v_i = 0$$

Thus, If A is Hermitian, then eigenvectors corresponding to different eigenvalues are orthogonal.

II. Understanding The Eigenvalues of Real Symmetric Matrices

Problem 3. (12 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a symmetric matrix, $\mathcal{S}_k$ denote a subspace of $\mathbb{R}^n$ of dimension k, and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ represent the eigenvalues of $\mathbf{A}$. For any $k \in \{1, 2, 3, ..., n\}$, prove that

$$\lambda_k = \min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{\mathbf{x} \in \mathcal{S}_{n-k+1}, ||\mathbf{x}||_2 = 1} \mathbf{x}^T \mathbf{A} \mathbf{x}.$$

Solution.

Recall that every real symmetric matrix admits an eigendecomposition. We decompose A as

$$A = UDU^T$$

where U is an orthogonal matrix and D is a diagonal matrix. We denote the columns of U as $u_1, u_2, ..., u_n$ such that $Au_i = \lambda_i u_i$.

- First we show $\min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{x \in \mathcal{S}_{n-k+1}, ||x||_2 = 1} x^T A x \leq \lambda_k$.
  Considering the subspace $U := span\{u_k, u_{k+1}, ..., u_n\}$, we have

  $$\min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{x \in \mathcal{S}_{n-k+1}, ||x||_2 = 1} x^T A x \leq \max_{x \in U, ||x||_2 = 1} x^T A x \leq \lambda_k \quad \text{(Rayleigh-Ritz)}$$

  where the last inequality follows from Rayleigh-Ritz theorem.
- Secondly, we show that $\min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{x \in \mathcal{S}_{n-k+1}, ||x||_2 = 1} x^T A x \geq \lambda_k$.
  Considering the subspace $W := span\{u_1, u_2, ..., u_k\}$ of dimension k, we have, $\forall \mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n$

  $$\dim(\mathcal{S}_{n-k+1} \cap W) = \dim(\mathcal{S}_{n-k+1}) + \dim(W) - \dim(\mathcal{S}_{n-k+1} \cup W) \geq n - k + 1 + k - n = 1$$

  So we can pick $v_0 \in \mathcal{S}_{n-k+1} \cup W$ with $\|x\|_2 = 1$ for any $\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n$. This gives

  $$\max_{x \in \mathcal{S}_{n-k+1}, ||x||_2 = 1} x^T A x \geq v_0^T A v_0 \geq \lambda_k$$

  where the last equality still follows from Rayleigh-Ritz theorem (see that $v_0^T A v_0 \geq \min\{u_1, u_2, ..., u_k\} = u_k$), which further implies that

  $$\min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{x \in \mathcal{S}_{n-k+1}, ||x||_2 = 1} x^T A x \geq \min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \lambda_k = \lambda_k$$
- Combining the results from 1) and 2), we prove that $\min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{x \in \mathcal{S}_{n-k+1}, ||x||_2 = 1} x^T A x = \lambda_k$ for any $k \in \{1, 2, 3, ..., n\}$.

Problem 4. (5 points+8 points+10 points) To assist the understanding of this problem, we first provide some **basic concepts of graph theory:**

① A *simple graph* G is a pair (V, E), such that

- V is the set of vertices;
- E is the set of edges and every edge is denoted by an *unordered* pair of its two *distinct* vertices.

② If i, j are two distinct vertices and (i, j) is an edge, we then say that i and j are *adjacent*. A graph is called d-regular graph if every vertex in the graph is adjacent to d vertices, where d is a positive integer.

③ Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, if $V_1 \subset V_2$ and $E_1 \subset E_2$, we call G_1 the *subgraph* of G_2. Furthermore, we call G_1 the *connected component* of G_2 if

- any vertex in G_1 is only connected to vertices in G_1.
- any two vertices in G_1 are connected either directly or via some other vertices in G_1;

Suppose $G = (V, E)$ is a simple graph with n vertices indexed by $1, 2, ..., n$ respectively. The adjacency matrix of G is a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ given by

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if vertex } i \text{ and vertex } j \text{ are adjacent;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Besides, if G is a d-regular graph, its *normalized Laplacian matrix* $\mathbf{L}$ is defined as $\mathbf{L} \triangleq \mathbf{I} - \frac{1}{d}\mathbf{A}$, where $\mathbf{I}$ is the identity matrix. Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ denote the eigenvalues of $\mathbf{L}$. Please prove the following propositions:

1) For any vector $\mathbf{x} \in \mathbb{R}^n$, it follows that
$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{d}\sum_{(i,j)\in E}(\mathbf{x}_i - \mathbf{x}_j)^2, \tag{2}$$
where i, j represent two distinct vertices and $(i, j) \in E$ represents an edge between i and j in the graph G.
2) $\lambda_n = 0$ and $\lambda_1 \leq 2$.
3) **(Bonus Problem)** the graph G has at least $(n - k + 1)$ connected components if and only if $\lambda_k = 0$.

Hint: The matrix $\mathbf{L}$ is real and symmetric. You can directly utilize Courant-Fischer Theorem without proof. Particularly, you may need to utilize the min-max form of the Courant-Fischer Theorem for the Bonus Problem.

Solution

1) We show the derivations as follows

$$
\begin{aligned}
x^T L x &= x^T x - \frac{1}{d} x^T A x \\
&= \sum_{i=1}^{n} x_i^2 - \frac{1}{d} \sum_{i=1} x_i \sum_{j=1}^{n} A_{ij} x_j \\
&= \sum_{i=1}^{n} x_i (x_i - \frac{1}{d} \sum_{j=1}^{n} A_{ij} x_j) \\
&\text{(By the definition of d-regular: every vertex is adjacent to } d \text{ vertices)} \\
&= \sum_{i=1}^{n} x_i (\frac{1}{d} \sum_{A_{ij} \neq 0} x_i - \frac{1}{d} \sum_{A_{ij} \neq 0} x_j) \\
&= \frac{1}{d} \sum_{i=1}^{n} x_i (\sum_{A_{ij} \neq 0} (x_i - x_j)) \\
&= \frac{1}{d} \sum_{i,j,A_{ij} \neq 0} (x_i^2 - x_i x_j) \\
&\text{(By the symmetry of adjacency matrix } A\text{)} \\
&= \frac{1}{d} \sum_{i>j,A_{ij} \neq 0} (x_i^2 - x_i x_j + x_j^2 - x_j x_i) + \frac{1}{d} \sum_{i=j,A_{ij} \neq 0} (x_i^2 - x_i^2) \\
&= \frac{1}{d} \sum_{i>j,A_{ij} \neq 0} (x_i - x_j)^2 \\
&\text{(By the definition of adjacency matrix } A\text{)} \\
&= \frac{1}{d} \sum_{(i,j) \in E} (x_i - x_j)^2
\end{aligned}
$$

Q.E.D.

2) • We first show that $\lambda = 0$ is an eigenvalue of L. Pick $v_1 = k\mathbf{1}, k \in R$, then

$$
\begin{aligned}
L v_1 &= v_1 - \frac{1}{d} A v_1 \\
&\text{(Through the fact that } \sum_{j=1}^{n} A_{ij} = d\text{)} \\
&= v_1 - \frac{1}{d} \cdot dk\mathbf{1} \\
&= v_1 - v_1 \\
&= 0 \\
\Rightarrow L v_1 &= 0 v_1
\end{aligned}
$$

Since $x^T L x = \frac{1}{d} \sum_{(i,j) \in E} (x_i - x_j)^2 \geq 0$, we know that L is semi-positive definite, which implies that $\lambda_i \geq 0$. Thus $\lambda_n = 0$

- Secondly, we show that $\lambda_1 \leq 2$.

$$\begin{aligned}
\lambda_1 &= \max_{||x||_2=1} x^T L x \\
&= \max_{||x||_2=1} \frac{1}{d} \sum_{(i,j)\in E} (x_i - x_j)^2 \\
&\text{(From the derivation in 1))} \\
&= \max_{||x||_2=1} \frac{1}{d} \sum_{i,j,A_{ij}\neq 0} (x_i^2 - x_i x_j) \\
&= \max_{||x||_2=1} \frac{1}{d} (\sum_{i,j,A_{ij}\neq 0} x_i^2 - \sum_{i,j,A_{ij}\neq 0} x_i x_j) \\
&\text{(By } \sum_{i=1}^{n} x_i^2 = 1, \sum_{i,j,A_{ij}\neq 0} x_i^2 = d\text{)} \\
&= \max_{||x||_2=1} (1 - \frac{1}{d} \sum_{i,j,A_{ij}\neq 0} x_i x_j)) \\
&\text{(Notice that the sign of } x_i \text{ are uncertain thus holding the following inequality)} \\
&\leq \max_{||x||_2=1} (1 + \frac{1}{d} \sum_{i,j,A_{ij}\neq 0} \frac{x_i^2 + x_j^2}{2}) \\
&= \max_{||x||_2=1} (1 + \frac{1}{d} \cdot \frac{2d}{2}) \\
&= 2
\end{aligned}$$

Q.E.D.

- In conclusion, $\lambda_n = 0, \lambda_1 \leq 2$.

3) - Firstly, we show the graph G has at least $(n-k+1)$ connected components $\Leftarrow \lambda_k = 0$.

$$\begin{aligned}
\lambda_k &= \min_{\mathcal{S}_{n-k+1}\subseteq \mathbb{R}^n} \max_{x\in \mathcal{S}_{n-k+1}, ||x||_2=1} x^T L x \\
&= \min_{\mathcal{S}_{n-k+1}\subseteq \mathbb{R}^n} \max_{x\in \mathcal{S}_{n-k+1}, ||x||_2=1} \frac{1}{d} \sum_{(i,j)\in E} (x_i - x_j)^2 \\
&= 0
\end{aligned}$$

This implies that, there exist a subspace W of dimension $n-k+1$ such that

$$\max_{x\in W, ||x||_2=1} \frac{1}{d} \sum_{(i,j)\in E} (x_i - x_j)^2 = 0.$$

which further indicates that

$$\forall x \in W, x_i = x_j, (i,j) \in E \tag{3}$$

Now, let's dive into the connected component in G under such conclusion. Since $\forall x \in W, x_i = x_j, (i,j) \in E$ and any two vertices in a connected component is connected by some paths, we have

$$x_i = x_j \text{ if they belong to the same connected component.}$$

This basically means that all vertices in a same connected components only admit one dimension. Based on that, the condition $x \in W$, which requires $\dim(W) = n - k + 1$ dimensions, could be satisfied only when there are at least $(n - k + 1)$ connected components.

Therefore, at least $n-k+1$ connected components are needed to meet the existence of W so that $\lambda_k = 0$ could be satisfied. Q.E.D.

- Secondly, we show the graph G has at least $(n - k + 1)$ connected components $\Rightarrow \lambda_k = 0$.

  Suppose G has exactly $(n-k+1)$ connected components. Denote the connected components as $C^1, C^2, ..., C^{n-k+1}$. Construct the basis $B = \{X^1, X^2, ..., X^{n-k+1}\}$ where

$$X^i_j = \begin{cases} 1 \text{ if } j \in C^i \\ 0 \text{ otherwise} \end{cases}$$

  Since any vertex in C^i will not connect to the vertex in C^j, $i \neq j$ (which can be verified by the definition of connected component), X^i and X^j contains no same vertices thus $X^i_m X^j_m = 0, \forall i \neq j, \forall m$. Therefore $\{X^i\}$ form a orthogonal basis such that

$$< X^i, X^j >= 0, \quad \forall i, j$$

  Based on that, we hereby define the subspace $W = span\{B\}$ with dimension $(n-k+1)$ and orthogonal basis $\{X^i\}$.

  – We show that $LX^i = 0$ for any X^i, which means X^i is an eigenvector of L with eigenvalue $= 0$.

$$\begin{aligned}
(LX^i)_j &= X^i_j - \frac{1}{d}(AX^i)_j \\
&= X^i_j - \frac{1}{d}\sum_{k=1}^{n} A_{jk} X^i_k \\
&= \begin{cases} X^i_j - 1 \text{ if } j \in C^i & \text{(since } A_j \text{ has total } d \text{ nonzero entries)} \\ X^i_j - 0 \text{ otherwise} & \text{(since vectex } j \text{ is disconnected with } C^i) \end{cases} \\
&= \begin{cases} 1 - 1 = 0 \text{ if } j \in C^i \\ 0 - 0 = 0 \text{ otherwise} \end{cases} \\
&= 0
\end{aligned}$$

  – Next, we show that $Lv = 0$ for any $v \in W$. Recall that the subspace $W = span\{B\}$ so that any $v \in W$ can be represented as linear combination of B: $v = \sum_{i=1}^{n-k+1} \alpha_i X^i$. Then we have

$$Lv = \sum_{i=1}^{n-k+1} \alpha_i L X^i = \vec{0}$$

This indicates that

$$\begin{aligned}\lambda_k &= \min_{\mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \max_{v \in \mathcal{S}_{n-k+1}, ||v||_2=1} v^T L v \\ &\leq \max_{v \in W, ||v||_2=1} v^T L v \\ &= \max_{v \in W, ||v||_2=1} v^T \vec{0} \\ &= 0\end{aligned}$$

Since $\lambda_k \geq \lambda_n = 0$, we have $\lambda_k = 0$.

– For the case that G has more than $(n-k+1)$ connected components, we similarly show that

$$\lambda_{k'} = 0$$

where $k' > k$. This implies that

$$0 = \lambda_n \leq \lambda_k \leq \lambda_{k'} = 0 \quad \Rightarrow \lambda_k = 0$$

in such cases. So we have proved that the graph G has at least $(n-k+1)$ connected components $\Rightarrow \lambda_k = 0$.

In conclusion, the graph G has at least $(n-k+1)$ connected components if and only if $\lambda_k = 0$. Q.E.D.

III. Eigenvalue Computations

A. Power Iteration

Problem 5. (20 points)

Consider the 2×2 matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 0 & \alpha \\ \beta & 0 \end{bmatrix}, \qquad \text{with } \alpha, \beta > 0\,.$$

1) Find the eigenvalues and eigenvectors of $\mathbf{A}$ by hand. (5 points)
2) Program **the power iteration** (See Algorithm 1) and **the inverse iteration** (See Algroithm 2) respectively and report the output of two algorithms for $\mathbf{A}$ (you can determine α, β by yourself), do the two algorithms converge or not? Report what you have found (you can use plots to support your analysis). (10 points: programming takes 5 points and the analysis takes 5 points) After a few iterations, the sequence given by the power iteration fails to converge, explain why. (5 points) (**After-class exercise:** If you want, you can study the case for other randomly generated matrices.)

Remarks: Programming languages are not restricted. In `Matlab`, you are free to use `[v,D] = eig(A)` to generate the eigenvalues and eigenvectors of $\mathbf{A}$ as a reference to study the convergence.

Algorithm 1: Power iteration

Input : $\mathbf{A} \in \mathbb{C}^{n \times n}$

Initilization: random choose $\mathbf{q}^{(0)}$.

for $k = 1, \dots,$ **do**

  $\mathbf{z}^{(k)} = \mathbf{A}\mathbf{q}^{(k-1)}$

  $\mathbf{q}^{(k)} = \mathbf{z}^{(k)} / \|\mathbf{z}^{(k)}\|_2$

  $\lambda^{(k)} = (\mathbf{q}^{(k)})^H \mathbf{A}\mathbf{q}^{(k)}$

end

Output: $\lambda^{(k)}$

Algorithm 2: Inverse iteration

Input : $\mathbf{A} \in \mathbb{C}^{n \times n}$, μ

Initilization: random choose $\mathbf{q}^{(0)}$.

for $k = 1, \dots,$ **do**

  $\mathbf{z}^{(k)} = (\mathbf{A} - \mu\mathbf{I})^{-1}\mathbf{q}^{(k-1)}$

  $\mathbf{q}^{(k)} = \mathbf{z}^{(k)} / \|\mathbf{z}^{(k)}\|_2$

  $\lambda^{(k)} = (\mathbf{q}^{(k)})^H \mathbf{A}\mathbf{q}^{(k)}$

end

Output: $\lambda^{(k)}$

Solution

1) We give the derivations in the below.

$$\det(\lambda I - A) = \lambda^2 - \alpha\beta = (\lambda - \sqrt{\alpha\beta})(\lambda + \sqrt{\alpha\beta}) \Rightarrow \lambda_1 = \sqrt{\alpha\beta}, \lambda_2 = -\sqrt{\alpha\beta}$$

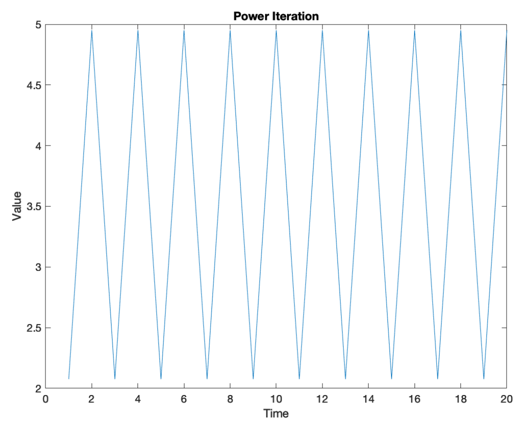
For $\lambda_1 = \sqrt{\alpha\beta}$

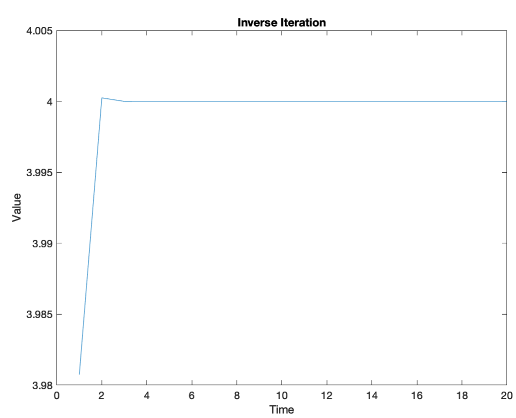
$$\begin{bmatrix} \sqrt{\alpha\beta} & -\beta \\ -\alpha & \sqrt{\alpha\beta} \end{bmatrix} v = 0 \Rightarrow v_1 = k \begin{bmatrix} \sqrt{\beta} \\ \sqrt{\alpha} \end{bmatrix}, \quad k \in R$$

For $\lambda_2 = -\sqrt{\alpha\beta}$

$$\begin{bmatrix} -\sqrt{\alpha\beta} & -\beta \\ -\alpha & -\sqrt{\alpha\beta} \end{bmatrix} v = 0 \Rightarrow v_2 = k \begin{bmatrix} \sqrt{\beta} \\ -\sqrt{\alpha} \end{bmatrix}, \quad k \in R$$

2)
- Settings: $\alpha = 2, \beta = 8$. Parameters $\mu = 3.9$ in Inverse iteration. Theoretical eigenvalues: $\lambda_1 = 4, \lambda_2 = -4$.
- Analysis: The power iteration method did not converge, while the inverse iteration method converged within ten steps with μ set as 3.9, which gives the correct result.
- Explanation: Recall that $|\lambda^{(k)} - \lambda_1| = \mathcal{O}(|\frac{\lambda_2}{\lambda_1}|^k)$. In this problem settings, the ratio of the absolute value of two eigenvalues is exact one, *i.e.* $\frac{|\lambda_1|}{|\lambda_2|} = 1$. Such an undesired ration results in the non-convergence of the residual in power iteration, thus leading to the non-convergence of the whole algorithm.





B. QR itertiaon and Hessenberg QR iteration

Recap. For $\mathbf{A} \in \mathbb{C}^{n\times n}$, consider the QR iteration (See Algorithm 3) for finding all the eigenvalues and eigenvectors of $\mathbf{A}$. In each iteration, $\mathbf{A}^{(k)}$ is similar to $\mathbf{A}$ in that

---

Algorithm 3: QR iteration

Input : $\mathbf{A} \in \mathbb{C}^{n\times n}$

Initilization: $\mathbf{A}^{(0)} = \mathbf{A}$.

for $k = 1, \ldots,$ **do**

  $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$ % Perform QR for $\mathbf{A}^{(k-1)}$

  $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$

end

Output: $\mathbf{A}^{(k)}$

$$\begin{aligned}\mathbf{A}^{(k)} &= \mathbf{R}^{(k)}\mathbf{Q}^{(k)} = (\mathbf{Q}^{(k)})^H\mathbf{Q}^{(k)}\mathbf{R}^{(k)}\mathbf{Q}^{(k)} = (\mathbf{Q}^{(k)})^H\mathbf{A}^{(k-1)}\mathbf{Q}^{(k)} = \cdots \\ &= (\mathbf{Q}^{(1)}\mathbf{Q}^{(2)}\cdots\mathbf{Q}^{(k)})^H\mathbf{A}(\mathbf{Q}^{(1)}\mathbf{Q}^{(2)}\cdots\mathbf{Q}^{(k)}) \Rightarrow \mathbf{A}^{(k)} \text{ is similar to } \mathbf{A}\,.\end{aligned}$$

Suppose the Schur decomposition of $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^H$, then under some mild assumptions, $\mathbf{A}^{(k)}$ converges to $\mathbf{T}$. Therefore, we can compute all the eigenvalues of $\mathbf{A}$ by taking the diagonal elements of $\mathbf{A}^{(k)}$ for sufficiently large k. However, each iteration requires $\mathcal{O}(n^3)$ flops to compute QR factorization which is computationally expensive. One possible solution is: first perform similarity transform $\mathbf{A}$ to an upper Hessenberg form (Step 1 in Algorithm 4),

---

Algorithm 4: Hessenberg QR iteration

Input : $\mathbf{A} \in \mathbb{C}^{n\times n}$

1 **Initilization:** $\mathbf{H} = \mathbf{Q}^H\mathbf{A}\mathbf{Q}\,, \mathbf{A}^{(0)} = \mathbf{H}$. % Hessenberg reduction for $\mathbf{A}$

2 **for** $k = 1, \ldots,$ **do**

3   $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$ % Perform QR for $\mathbf{A}^{(k-1)}$ using Givens QR

4   $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$ % Matrix computation

5 **end**

Output: $\mathbf{A}^{(k)}$

---

then perform QR iteration (Algorithm 3) over new $\mathbf{A}^{(0)} = \mathbf{H}$. By using Givens rotations, the QR step only takes $\mathcal{O}(n^2)$ flops.

Problem 6. (15 points +10 points)

1) Complete the Algorithm 5 (corresponding to the step 3-4 of Algrorithm 4) first (7 points), then show **the detailed derivation** of the computational complexity of in Algorithm 5 ($\mathcal{O}(n^2)$). (8 points) (Derivation is for the computaional complexity of the algorithm.)

   To be more specific, we can present the process of performing QR for $\mathbf{A}^{(k)}$ using Givens rotations as:

   (a) First, overwrite $\mathbf{A}^{(k)}$ with upper-triangular $\mathbf{R}^{(k)}$

   $$\mathbf{A}^{(k)} = (\mathbf{G}_m^H \mathbf{G}_{m-1}^H \cdots \mathbf{G}_1^H)\mathbf{A}^{(k)} = \mathbf{R}^{(k)} ,$$

   where $\mathbf{G}_1, \ldots, \mathbf{G}_m$ is a sequence of Givens rotations for some m (In your algorithm, you need to clearly specify what $\mathbf{G}_i$ is), and $\mathbf{R}^{(k)} = \mathbf{G}_1 \cdots \mathbf{G}_m$.

   (b) Perform matrix multiplication such that $\mathbf{A}^{(k)}$ is of Hessenberg form,

   $$\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} = \mathbf{A}^{(k)}\mathbf{G}_1 \cdots \mathbf{G}_m .$$

2) **(Bouns Problem) Implicit QR iteration**

   Another way to implement step 3-4 in Algorithm 4 is through *implicit QR iteration*. The idea is as follows, for $\mathbf{A}^{(0)} \in \mathbb{R}^{n\times n}$ which is of Hessenberg form,

   (a) First, compute a Givens rotation $\mathbf{G}_1$ such that $(\mathbf{G}_1^H \mathbf{A}^{(0)})_{2,1} = 0$ and update $\mathbf{A}^{(1)} = \mathbf{G}_1^H \mathbf{A}^{(0)} \mathbf{G}_1$. However, the entry $\mathbf{A}_{3,1}^{(1)}$ may be nonzero (known as "bulge").

   (b) Compute another Givens rotation $\mathbf{G}_2$ such that $(\mathbf{G}_2 \mathbf{A}^{(1)})_{3,1} = 0$ (i.e., nulling out the "bulge") and update $\mathbf{A}^{(2)} = \mathbf{G}_2^H \mathbf{A}^{(1)} \mathbf{G}_2$ which is analogous with step (a). Note that the entry $\mathbf{A}_{4,2}^{(2)}$ will now be nonzero.

   (c) Then, we try to find $\mathbf{G}_3$ such that $(\mathbf{G}_3 \mathbf{A}^{(2)})_{4,2} = 0$. The procedure of iterating nulling out the "bulges" to reset in a upper Hessenberg form is known as "bulge chasing".

   This algorithm *implicitly* computed QR factorization at the cost of $\mathcal{O}(n^2)$, and this is why the algorithm is called the *Implicit QR iteration*. Consider a 4×4 Hessenberg matrix

   $$\mathbf{A}^{(0)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 2 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix} .$$

   Carry out the implicit QR iteration (show the detailed derivation) (To simplify the computation, you can use `Matlab` to do the matrix multiplications. Specifically, explicitly show $\mathbf{G}_i$, $\mathbf{G}_i^H \mathbf{A}^{(i-1)}$ and $\mathbf{A}^{(i)}$ for each step but when computing the matrix multiplication such as $\mathbf{G}_i^H \mathbf{A}^{(i-1)}$, $\mathbf{G}_i^H \mathbf{A}^{(i-1)} \mathbf{G}_i$, you are free to use `Matlab` . But be careful with the precision issue during the process of computing.), and observe where does the so-called "bulge" appears. (5 points: including the detailed derivation of the implicit QR iteration and pointing out the "bulge") Based on your observations, explain why the implicit QR iteration is indeed equivalent to the Algorithm 5. (5 points)

Solution

Algorithm 5: Step 3-4 in Hessenberg QR iteration

```
Input : A^(k-1) ∈ C^{n×n} which is of upper Hessenberg form  % corresponding to A^(k-1) in
        step 3 of Algorithm 4
% Perform QR for A^(k) using Givens roations
A^(k) = A^(k-1)
for j = 1, ..., n-1 do
    θ_j = arctan(-A^k(j+1,j) / A^k(j,j))
    G_j = Givens_Rotation(j, j+1, θ_j)  % Here we specify what G_j is
    A^(k)(j:j+1, j:n) = [G_j(j,j) G_j(j,j+1); G_j(j+1,j) G_j(j+1,j+1)]^T A^(k)(j:j+1, j:n)
end
% Matrix computation
for j = 1, ..., n-1 do
    A^(k)(1:j+1, j:j+1) = A^(k)(1:j+1, j:j+1) [G_j(j,j) G_j(j,j+1); G_j(j+1,j) G_j(j+1,j+1)]
end
Output: A^(k)  % corresponding to A^(k) in step 4 of Algorithm 4
```

1) Complexity analysis:

- Givens rotation in first loop: $2 \times square + 1 \times addition + 1 \times root + 2 \times division = 6\, flops$.
- Matrix computation in first loop: $2 \times (n - j + 1) \times (2 + 1) = 6(n - j + 1)\, flops$.
- Matrix computation in second loop: $2 \times (j + 1) \times (2 + 1) = 6(j + 1)\, flops$.

Total complexity:

$$
\begin{aligned}
&\sum_{j=1}^{n-1} (6 + 6(n - j + 1) + 6(j + 1)) \\
=&\sum_{j=1}^{n-1} (6n + 18) \\
=&6n^2 + 12n - 18 \\
=&\mathcal{O}(6n^2)
\end{aligned}
$$

2) • Step1:

$$G_1 = \begin{bmatrix} 0.4472 & -0.8944 & 0 & 0 \\ 0.8944 & 0.4472 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G_1^H A^{(0)} = \begin{bmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & -0.4472 & -2.2361 & -2.6833 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

$$A^{(1)} = \begin{bmatrix} 4.2 & -0.4 & 2.2361 & 3.5777 \\ -0.4 & -0.2 & -2.2361 & -2.6833 \\ 0.8944 & 0.4472 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

Bulge:

$$A^{(1)}(3,1) = 0.8944$$

• Step2:

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.4082 & -0.9129 & 0 \\ 0 & 0.9129 & -0.4082 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G_2^H A^{(1)} = \begin{bmatrix} 4.2 & -0.4 & 2.2361 & 3.5777 \\ 0.9798 & 0.4899 & 3.6515 & 2.9212 \\ 0 & 0 & 0.8165 & 1.6330 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

$$A^{(2)} = \begin{bmatrix} 4.2 & 2.2045 & -0.5477 & 3.5777 \\ 0.9798 & 3.3133 & -1.9379 & 2.9212 \\ 0 & 0.7454 & -0.3333 & 1.6330 \\ 0 & 1.8257 & -0.8165 & 1 \end{bmatrix}$$

Bulge:

$$A^{(2)}(4,2) = 1.8257$$

- Step3:

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.3780 & -0.9258 \\ 0 & 0 & 0.9258 & 0.3780 \end{bmatrix}$$

$$G_3^H A^{(2)} = \begin{bmatrix} 4.2 & 2.2045 & -0.5477 & 3.5777 \\ 0.9798 & 3.1333 & -1.9379 & 2.9212 \\ 0 & 1.9720 & -0.8819 & 1.5430 \\ 0 & 0 & 0 & -1.1339 \end{bmatrix}$$

$$A^{(3)} = \begin{bmatrix} 4.2 & 2.2045 & 3.1053 & 1.8593 \\ 0.9798 & 3.1333 & 1.9720 & 2.8983 \\ 0 & 1.9720 & 1.0952 & 1.3997 \\ 0 & 0 & -1.0498 & -0.4286 \end{bmatrix}$$

- The algorithm 5 aim to keep the tridiagonal form of $A^{(k)}$, for the purpose of reducing the time complexity of QR factorization in each iteration. By such a way, similarity will also be hold so that eigenvalues could be found in the Schur form.

  The implicit QR iteration shares the same idea, but the difference compared to the aforementioned algorithm lies in the integration of the two steps in algorithm 5. By such a combination, the bulges should be eliminated in each iteration in order to maintain the tridiagonal form. However, we observe that this subtle design gives no essential difference with algorithm 5, since $A^{(k)} = G_m^H ... G_1^H A^{(k-1)} G_1 ... G_m = Q^H A^{(k-1)} Q$ holds still. On one hand, this ensures the similarity between $A^{(k)}$ and $A^{(k-1)}$. On the other, tridiagonal form of $A^{(k)}$ is always maintained when output.

  Therefore, there is no essential difference between Implicit QR Iteration and Algorithm 5.