

SI231 - Matrix Computations, Fall 2020-21

Homework Set #4

Prof. Yue Qiu and Prof. Ziping Zhao

Name: **Major:** IE

Student No.: **E-mail:**

Acknowledgements:

- 1) Deadline: **2020-11-20 23:59:00**
- 2) Submit your homework at **Gradescope**. Homework #4 contains two parts, the theoretical part and the programming part.
- 3) About 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 (Eigenvalue decomposition) & Eigendecomposition for Hermitian matrices
- Similar transformation, Schur decomposition & Diagonalization
- 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 $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ 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.

$$1) |\lambda E - A| = \begin{vmatrix} \lambda + 4 & 3 \\ -6 & \lambda - 5 \end{vmatrix} = (\lambda - 2)(\lambda + 1)$$

$$\therefore \lambda_1 = 2, \lambda_2 = -1 \therefore \mathbf{\Lambda} = \begin{bmatrix} 2 & \\ & -1 \end{bmatrix}$$

$\therefore A$ can be diagonalized

$$2E - A = \begin{bmatrix} 6 & 3 \\ -6 & -3 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} 1 & -2 \end{bmatrix}^\top$$

$$-E - A = \begin{bmatrix} 3 & 3 \\ -6 & -6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \Rightarrow v_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}^\top$$

$$\therefore V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix}$$

then $A = V\mathbf{\Lambda}V^{-1}$

- 2) the eigenspace of A is $\mathcal{A} = \text{span}\{v_1, v_2\}$

if $A \sim B$, then exist invertible matrix P ,

$$s, t. \quad b = P^{-1}AP$$

$$\therefore Ax = \lambda x$$

$$\therefore Ax = PBP^{-1}x = \lambda x$$

$$\therefore B(P^{-1}x) = \lambda(P^{-1}x)$$

$$\therefore \text{the eigenspace of } B \text{ is } B = \text{span}\{P^{-1}v, P^{-1}v_2\}$$

$$\therefore \text{ if } P \neq I$$

$$\text{then } \lambda \neq B$$

Problem 2. (6 points × 5)

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

- 1) The matrix \mathbf{A} is singular if and only if 0 is an eigenvalue of it.
- 2) $\text{rank}(\mathbf{A}) \geq$ number of nonzero eigenvalues of \mathbf{A} .
- 3) If \mathbf{A} admits an eigendecomposition (eigenvalue decomposition), $\text{rank}(\mathbf{A}) =$ 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)

$$\text{necessity: } \because |A| = \prod_{i=1}^n \lambda_i$$

$$\text{if } 0 \text{ is an eigenvalue of } A \text{ then } \prod_{i=1}^n \lambda_i = 0$$

$$\therefore |A| = 0 \quad \text{and } A \text{ is singular}$$

$$\text{sufficiency: } \because |A| = \prod_{i=1}^n \lambda_i$$

$$\therefore \text{ if } |A| = 0$$

$$\Rightarrow \text{ there must be a } 0 \text{ eigenvalue}$$

2) suppose λ_i is the eigen values of A

μ_i is the algebraic multiplicity of the eigen $\lambda_i = 0$

γ_i is the geometric multiplicity of the eigen $\lambda_i = 0$

we can get

$$\text{rank}(A) = n - \dim N(A)$$

$$\text{number of nonzero eigenvalues of } A = n - \mu_i$$

$$\because \mu_i \geq \gamma_i$$

$$\therefore \text{rank}(A) \geq \text{number of nonzero eigenvalues of } A$$

3) If A admits an eigendecom position

then there exists an invertible matrix that satisfies $P^{-1}AP = \Lambda$

$$\therefore \text{rank}(A) = r(P^{-1}AP) = r(\Lambda) = \text{number of nonzero eigenvalues of } \mathbf{A}.$$

4) Let λ be an arbitrary eigenvalue of A and \mathbf{x} is the eigenvector of A associated with λ . Then, $A\mathbf{x} = \lambda\mathbf{x}$

Then we can get,

$$\begin{aligned} \mathbf{x}^H(A\mathbf{x}) &= \mathbf{x}^H(\lambda\mathbf{x}) \\ &= \lambda\mathbf{x}^H\mathbf{x} \\ &= \lambda\|\mathbf{x}\|^2 \end{aligned}$$

We also have

$$\mathbf{x}^H(A\mathbf{x}) = (A\mathbf{x})^T \mathbf{x}^* = \mathbf{x}^T A^T \mathbf{x}^*$$

so

$$\mathbf{x}^T A^T \mathbf{x}^* = \lambda \|\mathbf{x}\|$$

Taking the complex conjugate of this equality, we have

$$\mathbf{x}^H A^H \mathbf{x} = \lambda^* \|\mathbf{x}\|$$

because $A^H = A$ so,

$$\begin{aligned} \lambda^* \|\mathbf{x}\| &= \mathbf{x}^H A \mathbf{x} \\ &= \mathbf{x}^H \lambda \mathbf{x} \\ &= \lambda \|\mathbf{x}\| \end{aligned}$$

$\therefore \mathbf{x}$ is an eigenvector, $\therefore \mathbf{x} \neq 0$ and $\|\mathbf{x}\| \neq 0$. Therefore,

$$\lambda = \lambda^*$$

5) x is the eigenvector of A associated with λ ,

y is the eigen vector of A associated with λ_2

so, $Ax = \lambda_1 x$, $Ay = \lambda_2 y$

we can get

$$\begin{aligned} \langle x, Ay \rangle &= (Ay)^H x = y^H A^H x = y^H Ax = y^H \lambda_1 x \\ &= \lambda_1 y^H x = \lambda_1 \langle x, y \rangle \end{aligned}$$

and $\langle x, Ay \rangle$ also equal to

$$\langle x, Ay \rangle = \langle x, \lambda_2 y \rangle = \lambda_2^H y^H x = \lambda_2 \langle x, y \rangle.$$

since $\lambda_2^H = \lambda_2$ because A is Hermitian so we get

$$\lambda_1 \langle x, y \rangle = \lambda_2 \langle x, y \rangle \Rightarrow (\lambda_1 - \lambda_2) \langle x, y \rangle = 0$$

Therefore $\lambda_1 \neq \lambda_2 \Rightarrow \langle x, y \rangle = 0$

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 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ represent the eigenvalues of \mathbf{A} . For any $k \in \{1, 2, 3, \dots, n\}$, 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.

$A \in \mathbb{R}^{n \times n}$ is a symmetric matrix.

$\therefore A$ have n orthogonal eigenvectors.

suppose u_1, u_2, \dots, u_n are n orthogonal eigenvectors,

and u_i represents the eigenvectors of A associated to λ_i

Then we will have $u_i^T u_j = 0$ for $i \neq j$

$$\text{suppose } x = \sum_{i=1}^k \alpha_i u_i, \quad U = \text{span}\{u_1, u_2, \dots, u_k\}$$

$$\begin{aligned} \text{then } x^T A x &= \left(\sum_{i=1}^k \alpha_i u_i^T \right) \left(\sum_{i=1}^k \alpha_i A u_i \right) \\ &= \left(\sum_{i=1}^k \alpha_i u_i^T \right) \left(\sum_{i=1}^k \alpha_i \lambda_i u_i \right) \\ &= \sum_{i=1}^k \lambda_i \alpha_i^2 u_i^T u_i \end{aligned}$$

when $\|x\|_2 = 1$, then $\sum_{i=1}^k \alpha_i^2 u_i^T u_i = 1$.

$$\therefore x^T A x \geq \lambda_k \sum_{i=1}^k \alpha_i^2 u_i^T u_i = \lambda_k$$

and $\therefore \dim u + \dim \mathcal{S}_{n-k+1} = n + 1 > n$

and \therefore there must exist a $x \in \mathcal{S}_{n-k+1}$

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

and because the equation was set up of any \mathcal{S}_{n-k+1}

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

On the other hand, if we suppose $x = \sum_{i=k}^n \alpha_i u_i$ it's easy to find $x \in V$ $V = \text{span}\{u_k, u_{k+1}, \dots, u_n\}$ and

$$x^T A x = \sum_{i=k}^n \lambda_i \alpha_i^2 u_i^T u_i \leq \lambda_k \sum_{i=k}^n \alpha_i^2 u_i^T u_i = \lambda_k$$

$$\text{so, } \max_{\mathbf{x} \in V, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x} \leq \lambda_k$$

And V is obviously one of the subspaces of \mathcal{S}_{n-k+1} , so

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

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

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 $1, 2, \dots, n$ respectively. The adjacency matrix of G is a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ given by

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

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 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ denote the eigenvalues of \mathbf{L} . Please prove the following propositions:

1) For any vector $\mathbf{x} \in \mathbb{R}^n$, it follows that

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{d} \sum_{(i,j) \in E} (\mathbf{x}_i - \mathbf{x}_j)^2, \quad (2)$$

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) I find it is more general to represent $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ than $L = I - \frac{1}{d}A$, and $L = I - \frac{1}{d}A$ is a specific case that all d_i are the same. so in the course of my proof, I use $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ and then simplify it. The degree matrix $D \in \mathbb{R}^{n \times n}$ is defined as the diagonal matrix with diagonal entries (d_1, \dots, d_n)

Laplacian matrix \mathbf{L} is defined as $\mathbf{L} = D^{-1/2}(D - A)D^{-1/2} = \mathbf{I} - D^{-1/2}\mathbf{A}D^{-1/2}$,

$$\begin{aligned}
 x^T L x &= x^T x - x^T D^{-1/2} A D^{-1/2} x \\
 &= \sum_i x_i^2 - \sum_{i,j} \frac{x_i}{\sqrt{d_i}} A_{ij} \frac{x_j}{\sqrt{d_j}} \\
 &= \sum_i d_i \left(\frac{x_i}{\sqrt{d_i}} \right)^2 - \sum_{(i,j) \in E} \frac{x_i}{\sqrt{d_i}} \cdot \frac{x_j}{\sqrt{d_j}} \\
 &= \sum_{\{i,j\} \in E} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 \\
 &= \frac{1}{d} \sum_{(i,j) \in E} (\mathbf{x}_i - \mathbf{x}_j)^2
 \end{aligned}$$

2) First, according to the definition of L and degree matrix D , we know that $(D - A)e = 0$, in which $e = (1, 1, \dots, 1)^T$, so e is a eigenvector of $D - A$ corresponding to eigenvalue 0.

Then we can find

$$L(D^{1/2}e) = D^{-1/2}(D - A)D^{-1/2}D^{1/2}e = D^{-1/2}(D - A)e = 0$$

$\therefore D^{1/2}e$ is an eigenvector of L of eigenvalue 0.

for any $x \in \mathbb{R}^n$, we can get

$$\begin{aligned}
 x^T L x &= x^T (I - D^{-1/2} \mathbf{A} D^{-1/2}) x \\
 &= \sum_{i \in V} x(i)^2 - \sum_{(i,j) \in E} \frac{2x(i)x(j)}{\sqrt{d(i)d(j)}} \\
 &= \sum_{(i,j) \in E} \left(\frac{x(i)}{\sqrt{d(i)}} - \frac{x(j)}{\sqrt{d(j)}} \right)^2 \\
 &\geq 0
 \end{aligned}$$

$\therefore L$ is positive semidefinite

$\therefore \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$

$\therefore L$ has nonnegative eigenvalues, so $\lambda_1 = 0$.

Suppose $\alpha_1 \geq \dots \geq \alpha_n$ are the eigenvalues of $D^{-1/2} \mathbf{A} D^{-1/2}$, and it's easy to know $\lambda_i = 1 - \alpha_{n+1-i}$

We can use the same way to prove 2 is positive semidefinite

$$x^T (I + D^{-1/2} \mathbf{A} D^{-1/2}) x = \sum_{i \in V} x(i)^2 + \sum_{(i,j) \in E} \frac{2x(i)x(j)}{\sqrt{d(i)d(j)}} = \sum_{(i,j) \in E} \left(\frac{x(i)}{\sqrt{d(i)}} + \frac{x(j)}{\sqrt{d(j)}} \right)^2 \geq 0$$

$$\text{so, } x^T x + x^T A x \geq 0 \implies \frac{x^T (D^{-1/2} \mathbf{A} D^{-1/2}) x}{x^T x} \geq -1$$

use Rayleigh-Ritz Theorem, $\implies \alpha_n \geq -1$

and $\therefore L = I - D^{-1/2} \mathbf{A} D^{-1/2}$

$$\therefore \lambda_1 = 1 - \alpha_n$$

$$\therefore \lambda_1 \leq 2$$

3) because $L \in R^{n \times n}$ is a symmetric matrix

$$\text{We have } \lambda_k = \min_{S_{n-k+1} \subseteq \mathbb{R}^n} \max_{\mathbf{x} \in S_{n-k+1}, \|\mathbf{x}\|_2=1} \mathbf{x}^T L \mathbf{x} = \min_{S_{n-k+1} \subseteq \mathbb{R}^n} \max_{\mathbf{x} \in S_{n-k+1}, \|\mathbf{x}\|_2=1} \frac{1}{d} \sum_{(i,j) \in E} (\mathbf{x}_i - \mathbf{x}_j)^2$$

Necessity:

If $\lambda_k = 0$, there must exist a $n - k + 1$ dimensional subspace S_{n-k+1} such that for every $\mathbf{x} \in S_{n-k+1}$ and every $\{i, j\} \in E$, we have $x_i = x_j$,

This means that each $\mathbf{x} \in S_{n-k+1}$ must be constant within each connected component of G , and so the dimension of S_{n-k+1} can be at most the number of connected components of G , meaning that G has at least $n - k + 1$ connected components.

Sufficiency:

Conversely, if G has at least $n - k + 1$ connected components, we can let S_{n-k+1} be the space of vectors that are constant within each component, and S_{n-k+1} is a space of dimension at least $n - k + 1$ such that for every element \mathbf{x} of S_{n-k+1} we have

$$\sum_{\{i,j\} \in E} (x_i - x_j)^2 = 0$$

meaning that S_{n-k+1} is a witness of the fact that $\lambda_k = 0$.

III. EIGENVALUE COMPUTATIONS

A. Power Iteration

Problem 5. (20 points)

Consider the 2×2 matrix \mathbf{A}

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

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

1 **Initialization:** random choose $\mathbf{q}^{(0)}$.

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

3 $\mathbf{z}^{(k)} = \mathbf{A}\mathbf{q}^{(k-1)}$

4 $\mathbf{q}^{(k)} = \mathbf{z}^{(k)} / \|\mathbf{z}^{(k)}\|_2$

5 $\lambda^{(k)} = (\mathbf{q}^{(k)})^H \mathbf{A} \mathbf{q}^{(k)}$

6 **end**

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

Algorithm 2: Inverse iteration

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

1 **Initialization:** random choose $\mathbf{q}^{(0)}$.

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

3 $\mathbf{z}^{(k)} = (\mathbf{A} - \mu\mathbf{I})^{-1} \mathbf{q}^{(k-1)}$

4 $\mathbf{q}^{(k)} = \mathbf{z}^{(k)} / \|\mathbf{z}^{(k)}\|_2$

5 $\lambda^{(k)} = (\mathbf{q}^{(k)})^H \mathbf{A} \mathbf{q}^{(k)}$

6 **end**

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

Solution

$$\begin{aligned}
1) \quad |\lambda E - A| &= \begin{vmatrix} \lambda & -\alpha \\ -\beta & \lambda \end{vmatrix} = \lambda^2 - \alpha\beta = (\lambda + \sqrt{\alpha\beta})(\lambda - \sqrt{\alpha\beta}) \\
\therefore \quad \lambda_1 &= -\sqrt{\alpha\beta} \quad \lambda_1 E - A = \begin{bmatrix} -\sqrt{\alpha\beta} & -\alpha \\ -\beta & -\sqrt{\alpha\beta} \end{bmatrix} \rightarrow \begin{bmatrix} -\sqrt{\alpha\beta} & -\alpha \\ 0 & 0 \end{bmatrix} \\
\Rightarrow \quad v_1 &= \begin{bmatrix} 1 & -\sqrt{\frac{\beta}{\alpha}} \end{bmatrix}^\top \\
\text{and } \lambda_2 &= \sqrt{\alpha\beta} \quad \lambda_2 E - A = \begin{bmatrix} \sqrt{\alpha\beta} & -\alpha \\ -\beta & \sqrt{\alpha\beta} \end{bmatrix} \rightarrow \begin{bmatrix} \sqrt{\alpha\beta} - \alpha & \\ 0 & 0 \end{bmatrix} \\
\Rightarrow \quad v_2 &= \begin{bmatrix} 1 & \sqrt{\frac{\beta}{\alpha}} \end{bmatrix}^\top \\
\text{so the eigenvalues are } \lambda_1 &= -\sqrt{\alpha\beta} \text{ and } \lambda_2 = \sqrt{\alpha\beta} \\
v_1 &= \begin{bmatrix} 1 & -\sqrt{\frac{\beta}{\alpha}} \end{bmatrix}^\top \text{ is the eigenvector of } A \text{ associated with } \lambda_1
\end{aligned}$$

$$v_2 = \begin{bmatrix} 1 & \sqrt{\frac{\beta}{\alpha}} \end{bmatrix}^\top \text{ is the eigenvector of } A \text{ associated with } \lambda_2$$

- 2) In algorithms 1, λ can't converge, λ will oscillate between two different values, for example, if I choose $\alpha = 2$ and $\beta = 8$, $q = [2 \quad 9]$, I will get $\lambda_{2k+1} = 4.9655$ and $\lambda_{2k} = 2.1176$, $k = 0, 1, \dots$, so it's not convergent
In algorithms 2, if $\mu > 0$, λ will converge to 4; if $\mu < 0$, λ will converge to -4 ; if $\mu = 0$, λ will not converge, λ will oscillate between two different values.

In the power iteration $\vec{q}_{n+1} = A\vec{q}_n$ and we normalize each vector $\vec{q}_{n+1} \leftarrow \vec{q}_{n+1} / \|\vec{q}_{n+1}\|$. Note that

$$A^2 = \alpha\beta \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \alpha\beta I$$

Thus, $\vec{q}_{2n} = A^{2n}\vec{q}_0 = (\alpha\beta)^n \vec{q}_0$ with $\vec{q}_0 = [1, 1]^\top$. Normalizing vector gives

$$\vec{q}_{2n} \leftarrow \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

While $\vec{q}_{2n+1} = A^{2n+1}\vec{q}_0 = AA^{2n}\vec{q}_0 = (\alpha\beta)^n A\vec{q}_0 = (\alpha\beta)^n [\alpha, \beta]^\top$. Normalizing gives

$$\vec{q}_{2n+1} \leftarrow \frac{1}{\sqrt{\alpha^2 + \beta^2}} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Iterations oscillate between these two vectors, thus the iterations do not converge.

B. QR iteration 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 : $\mathbf{A} \in \mathbb{C}^{n \times n}$

```

1 Initialization:  $\mathbf{A}^{(0)} = \mathbf{A}$ .
2 for  $k = 1, \dots$ , do
3    $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$  % Perform QR for  $\mathbf{A}^{(k-1)}$ 
4    $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$ 
5 end
```

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

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

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 Initialization:  $\mathbf{H} = \mathbf{Q}^H \mathbf{A} \mathbf{Q}$ ,  $\mathbf{A}^{(0)} = \mathbf{H}$ . % Hessenberg reduction for  $\mathbf{A}$ 
2 for  $k = 1, \dots$ , 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 Algorithm 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 computational 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 $\mathbf{G}_1, \dots, \mathbf{G}_m$ 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 : $\mathbf{A}^{(k-1)} \in \mathbb{C}^{n \times n}$ which is of upper Hessenberg form % corresponding to $\mathbf{A}^{(k-1)}$ in step 3 of Algorithm 4

1 % Perform QR for $\mathbf{A}^{(k)}$ using Givens rotations

2 **Initilization:** $\mathbf{R}^{(0)} = \mathbf{A}$, $\mathbf{A}^{(0)} = \mathbf{A}$, $\mathbf{Q}^{(0)} = \mathbf{I}$.

3 **for** $i = 1, \dots, n-1$ **do**

4 $\mathbf{G}_0 = \mathbf{I}$

5 **for** $j = n, n-1, \dots, i+1$ **do**

6 $\mathbf{J} = \mathbf{I}$

7 $c = \frac{\mathbf{A}(i,i)}{\mathbf{A}(i,i)^2 + \mathbf{A}(j,i)^2}$, $s = \frac{\mathbf{A}(j,i)}{\mathbf{A}(i,i)^2 + \mathbf{A}(j,i)^2}$

8 $\mathbf{J}(i,i) = c$; $\mathbf{J}(i,j) = s$; $\mathbf{J}(j,i) = -s$; $\mathbf{J}(j,j) = c$;

9 $\mathbf{A}^{(i-1)} = \mathbf{J} * \mathbf{A}^{(i-1)}$

10 $\mathbf{G}_i = \mathbf{J} * \mathbf{G}_{i-1}$

11 **end**

12 $\mathbf{Q}^{(i)} = \mathbf{Q}^{(i-1)} \mathbf{G}_i^H$

13 $\mathbf{R}^{(i)} = \mathbf{G}_i * \mathbf{R}^{(i-1)}$

14 $\mathbf{A}^{(i)} = \mathbf{R}^{(i)} \mathbf{Q}^{(i)}$

15 **end**

16 % Matrix computation

Output: $\mathbf{A}^{(k)}$ % corresponding to $\mathbf{A}^{(k)}$ in step 4 of Algorithm 4

1) In step 7 – 8, it has 8 flops.

In step 9 – 10, because \mathbf{J} is Givens rotation matrix, so it has $2^3 = 8$ flops in every Matrix multiplication. so $2 \times 8 = 16$ flops in step 9 – 10.

In step 12, it has $8 \times [n - (i + 1) + 1] = 8(n - i)$ flops.

In step 13, it also has $8 \times [n - (i + 1) + 1] = 8(n - i)$ flops.

so, it to tally has

$$\sum_{i=1}^{n-1} [(8 + 16)(n - i) + 2 \times 8(n - i)] = 20(n^2 - n) = \mathcal{O}(n^2)$$

$$2) \ G_1 = \begin{vmatrix} 0.4472 & 0.8944 & 0 & 0 \\ -0.8944 & 0.4472 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{vmatrix}, \ G_2 = \begin{vmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & -0.4082 & 0.9129 & 0 \\ 0 & 0.9129 & 0.4082 & 0 \\ 0 & 0 & 0 & -1.0000 \end{vmatrix}$$

$$G_3 = \begin{vmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & -0.3780 & -0.9258 \\ 0 & 0 & 0.9258 & -0.3780 \end{vmatrix}, A_1 = \begin{vmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & -0.4472 & -2.2361 & -2.6833 \\ 0 & 1.0000 & 3.0000 & 2.0000 \\ 0 & 0 & 2.0000 & 1.0000 \end{vmatrix}$$

$$A_2 = \begin{vmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & 1.0954 & 3.6515 & 2.9212 \\ 0 & 0 & -0.8165 & -1.6330 \\ 0 & 0 & -2.0000 & -1.0000 \end{vmatrix}, A_3 = \begin{vmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & 1.0954 & 3.6515 & 2.9212 \\ 0 & 0 & 2.1602 & 1.5430 \\ 0 & 0 & 0 & -1.1339 \end{vmatrix}$$

- (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").

Answer is on the below:

$$\mathbf{G}_1^H \mathbf{A}^{(0)} = \begin{vmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & -0.4472 & -2.2361 & -2.6833 \\ 0 & 1.0000 & 3.0000 & 2.0000 \\ 0 & 0 & 2.0000 & 1.0000 \end{vmatrix}$$

$$\mathbf{A}^{(1)} = \mathbf{G}_1^H \mathbf{A}^{(0)} \mathbf{G}_1 = \begin{vmatrix} 1.0000 & -2.0000 & 0.4472 & 0 \\ -2.0000 & 3.0000 & 3.1305 & 4.4721 \\ -0.8944 & 0.4472 & 3.0000 & 2.0000 \\ 0 & 0 & 2.0000 & 1.0000 \end{vmatrix}$$

- (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.

Answer is on the below:

$$\mathbf{G}_2^H \mathbf{A}^{(1)} = \begin{vmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & 1.0954 & 3.6515 & 2.9212 \\ 0 & 0 & -0.8165 & -1.6330 \\ 0 & 0 & -2.0000 & -1.0000 \end{vmatrix}$$

$$\mathbf{A}^{(2)} = \mathbf{G}_2^H \mathbf{A}^{(1)} \mathbf{G}_2 = \begin{vmatrix} 2.2361 & 0.5806 & 4.1789 & -3.5777 \\ 0 & 2.8861 & 2.4907 & -2.9212 \\ 0 & -0.7454 & -0.3333 & 1.6330 \\ 0 & -1.8257 & -0.8165 & 1.0000 \end{vmatrix}$$

- (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".

Answer is on the below:

$$\mathbf{G}_3^H \mathbf{A}^{(2)} = \begin{vmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & 1.0954 & 3.6515 & 2.9212 \\ 0 & 0 & 2.1602 & 1.5430 \\ 0 & 0 & 0 & -1.1339 \end{vmatrix}$$