SI231 - Matrix Computations, Fall 2020-21

Homework Set #4

Prof. Yue Qiu and Prof. Ziping Zhao

Name: aaa **Major:** Master in CS

Student No.: 2018123456 **E-mail:** aaa@shanghaitech.edu.cn

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) This problem is graded by Lin Zhu (**zhulin@**).

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) Setting as the characteristic polynomial of $\mathbf{A}$ equals to zero gives

$$p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = \det\left(\begin{bmatrix} -4-\lambda & -3 \\ 6 & 5-\lambda \end{bmatrix}\right)$$
$$= (-4-\lambda)(5-\lambda) + 18 = \lambda^2 - \lambda + 2 = (\lambda+1)(\lambda-2) = 0\,,$$

so the eigenvalues are $\lambda_1 = -1$ and $\lambda_2 = 2$ being distinct, then $\mathbf{A}$ can be diagonalized. (3 points)

A similarity transformation $\mathbf{V}$ that diagnoalizes $\mathbf{A}$ is constructed from a complete set of independent eigenvectors. Compute a pair of eigenvectors associated with λ_1 and λ_2 to be

$$\mathbf{v}_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix},\ \mathbf{v}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix},\ \text{and } \mathbf{V} = \begin{bmatrix} -1 & -1 \\ 1 & 2 \end{bmatrix}. \text{ (2 points)}$$

Then the matrix $\mathbf{A}$ can be diagonalized as

$$\mathbf{\Lambda} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V} = \begin{bmatrix} -2 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} -4 & -3 \\ 6 & 5 \end{bmatrix}\begin{bmatrix} -1 & -1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix}. \text{ (1 point)}$$

2) The answer is "yes". From 1), we know that $\mathbf{A}$ is similar to $\mathbf{\Lambda}$, (1 point) but the two eigenspaces for $\mathbf{A}$ are

$$\text{span}\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) \text{ of } \lambda_1 \text{ and span}\left(\begin{bmatrix} -1 \\ 2 \end{bmatrix}\right) \text{ of } \lambda_2, \text{ (2 points)}$$

while the two eigenspaces for $\mathbf{\Lambda}$ are given by

$$\text{span}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \text{ of } -1 \text{ and span}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \text{ of } 2. \text{ (1 point)}$$

Problem 2. (6 points × 5) This problem is graded by Sihang Xu (**xush@**).

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 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) The matrix $\mathbf{A}$ has such a Schur decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^H, \tag{1}$$

where $\mathbf{U} \in \mathbb{C}^{n\times n}$ is unitary and $\mathbf{T} \in \mathbb{C}^{n\times n}$ is upper triangular with $\{\mathbf{T}_{ii}, 1 \leq i \leq n\}$ being exactly the set of all the eigenvalues of $\mathbf{A}$. We accordingly get

$$\det(\mathbf{A}) = \det(\mathbf{U}) \cdot \det(\mathbf{T}) \cdot \det(\mathbf{U}^H) = \det(\mathbf{T}) = \prod_{i=1}^{n} \mathbf{T}_{ii}.$$

If $\exists\, i$ such that $\mathbf{T}_{ii} = 0$, i.e., 0 is one eigenvalue of $\mathbf{A}$, it follows that $\det(\mathbf{A}) = 0$, $\mathbf{A}$ is singular; vice verse: if $\mathbf{A}$ is singular, then $\det(\mathbf{T}) = \det(\mathbf{A}) = 0$, so $\exists\, \mathbf{T}_{ii}$ is 0 being an eigenvalue of $\mathbf{A}$.

2) From the Schur decomposition (1) in 1), we have

$$\mathsf{rank}(\mathbf{A}) = \mathsf{rank}(\mathbf{U}^H\mathbf{A}\mathbf{U}) = \mathsf{rank}(\mathbf{T}).$$

If the algebraic multiplicity of the zero eigenvalue of $\mathbf{A}$ is μ_0, there will be μ_0 zero items on the diagonal of $\mathbf{T}$. Then, with $\mathbf{T}$ being upper triangular, we can consequently get $\mathsf{rank}(\mathbf{T}) = n - \dim(\mathcal{N}(\mathbf{T})) \geq n - \mu_0$. On the other hand, the number of nonzero eigenvalues of $\mathbf{A}$ is exactly $n - \mu_0$. Therefore, $\mathsf{rank}(\mathbf{A}) \geq$ number of nonzero eigenvalues of $\mathbf{A}$.

3) If $\mathbf{A}$ admits such an eigendecomposition $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ with $\mathbf{V}, \mathbf{\Lambda} \in \mathbb{C}^{n\times n}$, then

$$\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\} = \mathrm{diag}\{\underbrace{0, 0, \dots, 0}_{\text{multiplicity:}\mu_0}, \underbrace{\lambda'_1, \lambda'_1, \dots, \lambda'_1}_{\text{multiplicity:}\mu_1}, \underbrace{\lambda'_2, \lambda'_2, \dots, \lambda'_2}_{\text{multiplicity:}\mu_2}, \dots, \underbrace{\lambda'_d, \lambda'_d, \dots, \lambda'_d}_{\text{multiplicity:}\mu_d}\}.$$

Here, μ_0 is the algebraic multiplicity of the zero eigenvalue, for $1 \leq i \leq d$, μ_i is the algebraic multiplicity of the nonzero eigenvalue λ'_i and $\sum_{i=1}^{d} \mu_i = n$.

It follows that $\mathsf{rank}(\mathbf{A}) = \mathsf{rank}(\mathbf{V}^{-1}\mathbf{A}\mathbf{V}) = \mathsf{rank}(\mathbf{\Lambda}) = \sum_{i=1}^{c} \mu_i = n - \mu_0$, which is exactly the number of nonzero eigenvalues.

4) Assume λ is an eigenvalue of $\mathbf{A}$ and $\mathbf{v}$ is the corresponding eigenvector. Then $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. We have $\mathbf{v}^H\mathbf{A}\mathbf{v} = \mathbf{v}^H\lambda\mathbf{v} = \lambda\mathbf{v}^H\mathbf{v}$ and $\mathbf{v}^H\mathbf{A}^H\mathbf{v} = (\mathbf{A}\mathbf{v})^H\mathbf{v} = (\lambda\mathbf{v})^H\mathbf{v} = \bar{\lambda}\mathbf{v}^H\mathbf{v}$. For $\mathbf{A} = \mathbf{A}^H$, we have $\mathbf{v}^H\mathbf{A}\mathbf{v} = \mathbf{v}^H\mathbf{A}^H\mathbf{v}$, which means $\lambda = \bar{\lambda}$ and hence λ is real.

5) Assume $\mathbf{A}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1$ and $\mathbf{A}\mathbf{v}_2 = \lambda_2 \mathbf{v}_2$ where $\lambda_1 \neq \lambda_2$ and $\mathbf{v}_1, \mathbf{v}_2$ are nonzero. By 4), λ_1 and λ_2 are real. Then

$$\begin{aligned} \lambda_1 \mathbf{v}_2^H \mathbf{v}_1 = \mathbf{v}_2^H (\lambda_1 \mathbf{v}_1) = \mathbf{v}_2^H (\mathbf{A}\mathbf{v}_1) = \mathbf{v}_2^H \mathbf{A}\mathbf{v}_1 &\xlongequal{\text{Hermitian } \mathbf{A}} \mathbf{v}_2^H \mathbf{A}^H \mathbf{v}_1 \\ = (\mathbf{A}\mathbf{v}_2)^H \mathbf{v}_1 = (\lambda_2 \mathbf{v}_2)^H \mathbf{v}_1 = \bar{\lambda}_2 \mathbf{v}_2^H \mathbf{v}_1 &\xlongequal{\text{real } \lambda_2} \lambda_2 \mathbf{v}_2^H \mathbf{v}_1 \end{aligned}$$

Therefore, for $\lambda_1 \neq \lambda_2$, we must have $\mathbf{v}_2^H \mathbf{v}_1 = 0$. The proof completes.

Common Mistake:

1) Some student use $(\lambda \mathbf{v})^H = \lambda \mathbf{v}^H$ instead of $\bar{\lambda} \mathbf{v}^H$ in the pro, in this case two point will be deducted.

II. Understanding The Eigenvalues of Real Symmetric Matrices

Problem 3. (12 points) This problem is graded by Yijia Chang (**changyj**@).

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: Let $u_1, u_2, ..., u_n$ be orthogonal eigenvectors of the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$. Let $\mathcal{V}$ denote a subspace of $\mathbb{R}^n$ of dimension $n - k + 1$. Let $\mathcal{W}$ denote the subspace spanned by $\{u_1, u_2, ..., u_k\}$. We can observe that $\dim(\mathcal{V} \cap \mathcal{W}) \geq 1$, which implies that $\mathcal{V} \cap \mathcal{W} \neq \emptyset$.

Consider such a vector $\mathbf{a} \in \mathcal{V} \cap \mathcal{W}$ that $||\mathbf{a}||_2 = 1$. Since $\mathbf{a} \in \mathcal{W}$, we can derive that $\mathbf{a} = \sum_{i=1}^{k} c_i u_i$ with $||\mathbf{a}||_2 = 1$, and then

$$\mathbf{a}^T\mathbf{A}\mathbf{a} = \sum_{i=1}^{k} \lambda_i c_i^2 \geq \lambda_k \sum_{i=1}^{k} c_i^2 = \lambda_k,$$

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

Taking a special $\mathcal{V}$ as the subspace spanned by $\{u_k, u_{k+1}, ..., u_n\}$, we can similarly get

$$\lambda_k \geq \max_{\mathbf{x}\in\mathcal{V},||\mathbf{x}||_2=1} \mathbf{x}^T\mathbf{A}\mathbf{x}.$$

Hence, the claim is proved that

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

Problem 4. (5 points+8 points+10 points) This problem is graded by Yijia Chang (**changyj@**).

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

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if vertex } i \text{ and vertex } j \text{ are adjacent;} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

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

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{d}\sum_{(i,j)\in E}(\mathbf{x}_i - \mathbf{x}_j)^2, \tag{3}$$

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) Equation (3) can be proved by verifying that the both sides equal

$$\sum_{i\in V}\mathbf{x}_i^2 - \frac{2}{d}\sum_{(i,j)\in E}\mathbf{x}_i\mathbf{x}_j.$$

2) ① Since $\mathbf{L}$ is a real symmetric matrix, by the Courant-Fischer Minimax Theorem, we get

$$\lambda_n = \min_{\mathbf{x}\in\mathbb{R}^n, ||\mathbf{x}||_2=1}\mathbf{x}^T\mathbf{L}\mathbf{x} = \min_{\mathbf{x}\in\mathbb{R}^n, ||\mathbf{x}||_2=1}\frac{1}{d}\sum_{(i,j)\in E}(\mathbf{x}_i - \mathbf{x}_j)^2 \geq 0.$$

Let $\mathbf{x} = (1/\sqrt{n}, 1/\sqrt{n}, 1/\sqrt{n}, ..., 1/\sqrt{n})^T \in \mathbb{R}^n$, then $||\mathbf{x}||_2 = 1$ and $\frac{1}{d}\sum\limits_{(i,j)\in E}(\mathbf{x}_i - \mathbf{x}_j)^2 = 0$. Hence, $\lambda_n = 0$.

② By Equation (3), we can also derive that

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

According to the Courant-Fischer Minimax Theorem, we have

$$2-\lambda_1 = \min_{\mathbf{x}\in\mathbb{R}^n,||\mathbf{x}||_2=1}\mathbf{x}^T(2\mathbf{I}-\mathbf{L})\mathbf{x} = \min_{\mathbf{x}\in\mathbb{R}^n,||\mathbf{x}||_2=1}\frac{1}{d}\sum_{(i,j)\in E}(\mathbf{x}_i+\mathbf{x}_j)^2 \geq 0,$$

which implies that $\lambda_1 \leq 2$.

3) According to the Courant-Fischer Minimax Theorem, we have

$$\lambda_k = \min_{\mathcal{S}_{n-k+1}\subseteq\mathbb{R}^n}\max_{\mathbf{x}\in\mathcal{S}_{n-k+1},||\mathbf{x}||_2=1}\mathbf{x}^T\mathbf{L}\mathbf{x} = \min_{\mathcal{S}_{n-k+1}\subseteq\mathbb{R}^n}\max_{\mathbf{x}\in\mathcal{S}_{n-k+1},||\mathbf{x}||_2=1}\frac{1}{d}\sum_{(i,j)\in E}(\mathbf{x}_i-\mathbf{x}_j)^2 \geq 0.$$

($\Leftarrow$) If $\lambda_k = 0$, then there exists a space $\mathcal{S}$ of dimension $(n-k+1)$ such that $\mathbf{x}_i = \mathbf{x}_j$ for every vector $\mathbf{x}\in\mathcal{S}$ and every edge $(i,j)\in E$. In other words, $\forall\mathbf{x}\in\mathcal{S}$, if two vertices i,j belong to the same connected component, then we have $\mathbf{x}_i = \mathbf{x}_j$. Hence, the dimension of $\mathcal{S}$ can not be larger than the number of connected components, which implies that G has at least $(n-k+1)$ connected components.

($\Rightarrow$) Let $\mathcal{S}$ denote the following vector space:

$$\{\mathbf{x} \mid \mathbf{x}_i = \mathbf{x}_j,\ \text{if } i \text{ and } j \text{ belong to the same connected component}\}.$$

If G has at least $(n-k+1)$ connected components, then the dimension of $\mathcal{S}$ is at least $(n-k+1)$ and $\frac{1}{d}\sum_{(i,j)\in E}(\mathbf{x}_i-\mathbf{x}_j)^2 = 0$. Suppose the dimension of $\mathcal{S}$ is $n-k_1+1$, then we have $\lambda_{k_1} = 0$ and $k_1 \leq k$. Hence, $\lambda_k \leq \lambda_{k_1} = 0$.

III. Eigenvalue Computations

A. *Power Iteration*

Problem 5. (20 points) This problem is graded by Zhihang Xu (**xuzhh**@).

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 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 $k = 1, \ldots,$ 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  : $\mathbf{A} \in \mathbb{C}^{n \times n}$, $\mu$
   Initilization: random choose $\mathbf{q}^{(0)}$.
   for $k = 1, \ldots,$ 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: $\lambda^{(k)}$
```

Recap. First, assume that $\mathbf{A} \in \mathbb{C}^{n\times n}$ is diagonalizable with eigenvalues $\lambda_1, \dots, \lambda_n$ ordered as

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \,.$$

and n linearly independent eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$. Diagonalize $\mathbf{A}$ as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$, and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$, then we have

$$\mathbf{A}^k = \underbrace{(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})\cdots(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})}_{k} = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1} \,.$$

For a randomly chosen vector $\widehat{\mathbf{v}} \in \mathbb{C}^n$, we can write it as a linear combination of the eigenvectors, i.e, $\widehat{\mathbf{v}} = \sum_{i=1}^n \alpha_i \mathbf{v}_i$. Then

$$\mathbf{A}^k\widehat{\mathbf{v}} = \mathbf{A}^k \sum_{i=1}^n \alpha_i \mathbf{v}_i = \sum_{i=1}^n \alpha_i \mathbf{A}^k \mathbf{v}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{v}_i = \lambda_1^k \left[\alpha_1 \mathbf{v}_1 + \sum_{i=2}^n \left(\frac{\lambda_i}{\lambda_1}\right)^k \alpha_i \mathbf{v}_i \right] \,, \tag{4}$$

if $|\lambda_1| > |\lambda_2|$, then $\lim_{k\to\infty}(\lambda_i/\lambda_1)^k = 0$ for $i = 2, \dots, n$. And $\mathbf{A}^k\widehat{\mathbf{v}}$ converges to $\mathbf{v}_1$. Adding a normalizing step, this leads to the most basic method of computing an eigenvalue and eigenvector, the Power method (see Algorithm 1). And the convergence rate for the power method is

$$|\lambda^{(k)} - \lambda_1| = \mathcal{O}\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) \,.$$

Now suppose that $f(\mathbf{z})$ is any function defined locally by a convergent power series. Then as long as the eigenvalues are within the radius of convergence, we can define $f(\mathbf{A})$ via the same power series, and

$$f(\mathbf{A}) = \mathbf{V} f(\mathbf{\Lambda}) \mathbf{V}^{-1} \,,$$

where $f(\mathbf{\Lambda}) = \mathrm{diag}(f(\lambda_1), \dots, f(\lambda_n))$. So the spectrum of $f(\mathbf{A})$ is the image of the spectrum of $\mathbf{A}$ under the mapping f, a fact known as the spectral mapping theorem. As a particular instance, consider the function $f(z) = (z-\mu)^{-1}$. This gives us

$$(\mathbf{A} - \mu\mathbf{I})^{-1} = \mathbf{V}(\mathbf{\Lambda} - \mu\mathbf{I})^{-1}\mathbf{V}^{-1} \,,$$

so if we run power iteration on $(\mathbf{A} - \mu\mathbf{I})^{-1}$, we will converge to the eigenvector corresponding to the eigenvalue λ_j for which $(\lambda_j - \mu)^{-1}$ is maximal, i.e., we can find the eigenvalue closest to μ. Running the power iteration on $(\mathbf{A} - \mu\mathbf{I})^{-1}$ is called the inverse iteration or inverse power method (see Algorithm.2). Here we usually call μ *shift*. The convergence rate for the inverse iteration is

$$\left| \frac{\mu - \lambda_j}{\mu - \lambda_k} \right|$$

where λ_j and λ_k are the closest and second closest eigenvalues to μ.

Solution.

1) Setting the characteristic polynomial of $\mathbf{A}$ equals to zero gives

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \det\left(\begin{bmatrix} -\lambda & \alpha \\ \beta & -\lambda \end{bmatrix} \right) = \lambda^2 - \alpha\beta = 0 \Rightarrow \lambda_1 = \sqrt{\alpha\beta} \,, \lambda_2 = -\sqrt{\alpha\beta} \,.$$

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

$$(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{v}_1 = \begin{bmatrix} -\sqrt{\alpha\beta} & \alpha \\ \beta & -\sqrt{\alpha\beta} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \mathbf{0}\,, \quad \Rightarrow \mathbf{v}_1 = \begin{bmatrix} \sqrt{\alpha} \\ \sqrt{\beta} \end{bmatrix},$$

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

$$(\mathbf{A} - \lambda_2 \mathbf{I})\mathbf{v}_2 = \begin{bmatrix} \sqrt{\alpha\beta} & \alpha \\ \beta & \sqrt{\alpha\beta} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \mathbf{0}\,, \quad \Rightarrow \mathbf{v}_2 = \begin{bmatrix} \sqrt{\alpha} \\ -\sqrt{\beta} \end{bmatrix}.$$

2) Take $\alpha = 100$ and $\beta = 1$ for example, the eigenvalues for $\mathbf{A}$ are $\sigma(\mathbf{A}) = \{10, -10\}$. Run the power iteration and the inverse iteration for $\mathbf{A}$,

- The power itertion can not converge (See Fig 1(a)).
- In inverse iteration, the algorithm can converge to the largest eigevalue for arbitrary positive μ (See Fig. 1(c)). The algorithm can converge to the smallest eigenvalue for arbitrary negative μ (See Fig. 1(d)) and cannot converge for $\mu = 0$ (See Fig. 1(b)).
- Different μ correspond to different convergence rates in inverse iteration (See Fig 1(e)).

There are several perspectives to explain why power iteration does not converge:

a) Eigenvalues of matrix $\mathbf{A}$ is $\sigma(\mathbf{A}) = \{-\sqrt{\alpha\beta}, \sqrt{\alpha\beta}\}$, which demonstrates that: there is no dominating eigenvalue since the absolute value of eigenvalues are the same.

b) Ignore the normalizing part of the algorithm, the power iteration iterates the following steps:

$$\mathbf{q}^{(k)} = \mathbf{A}\mathbf{q}^{(k-1)} = \cdots = \mathbf{A}^k \mathbf{q}^{(0)} = \lambda_1^k \left[\alpha_1 \mathbf{v}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k \mathbf{v}_i \right],$$

if $|\lambda_1| = |\lambda_2|$, then we can see that when $k \to \infty$, $\mathbf{q}^{(k)}$ cannot converge.

c) Compute $\mathbf{q}_k$ for each step and found out that they are periodic.

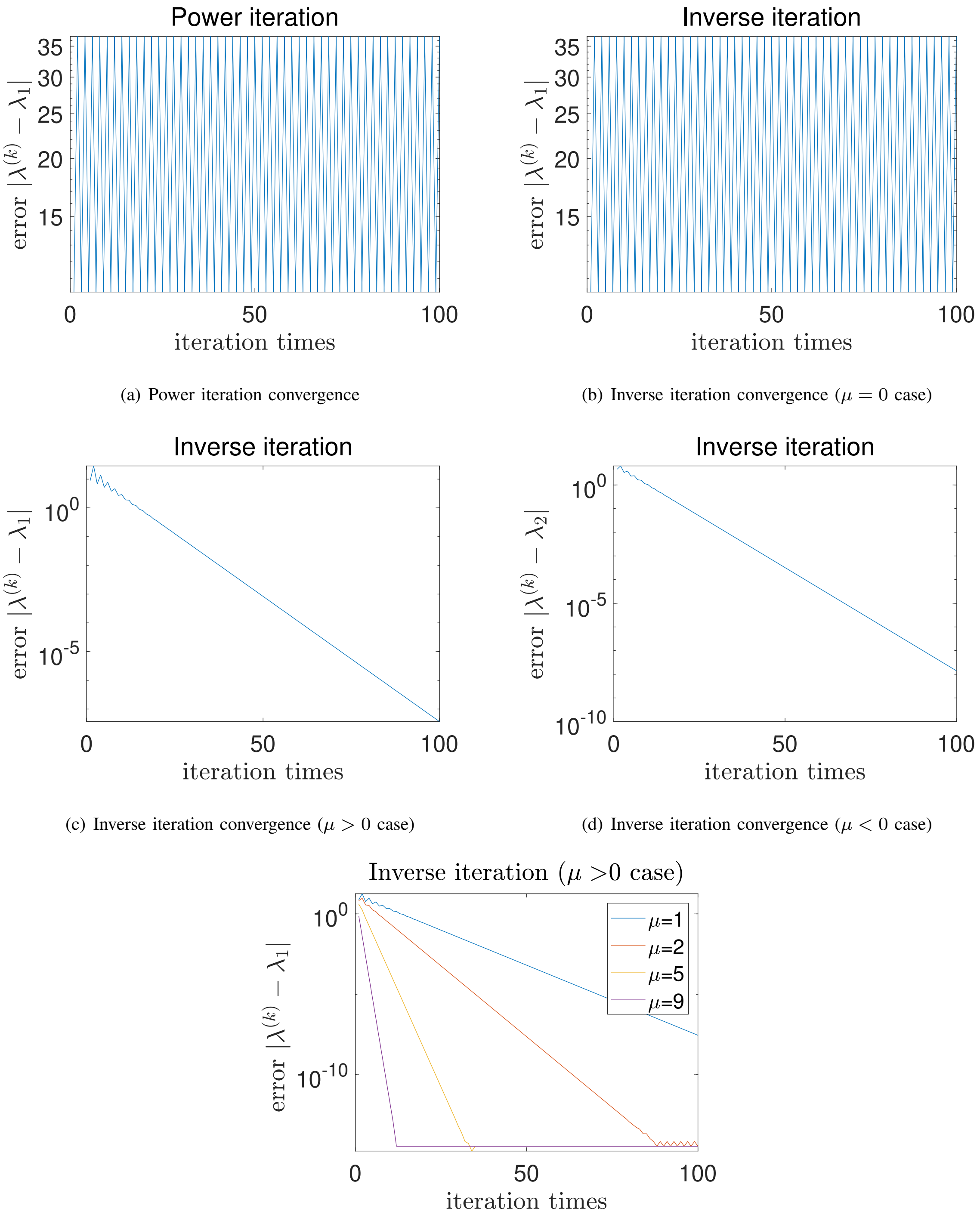
Remarks. This problem contains two sub-problems,

1) In sub-problem 1), you are required to give the eigenvalues and eigenvectors $\mathbf{A}$.

2) In sub-problem 2), first, programming two algorithms is required, and run the algorithm with $\mathbf{A}$.

Grading policy.

1) In sub-problem 2), you will lose 5 points if you did not give explanation of why power iteration fails to converge.



(a) Power iteration convergence

(b) Inverse iteration convergence ($\mu = 0$ case)

(c) Inverse iteration convergence ($\mu > 0$ case)

(d) Inverse iteration convergence ($\mu < 0$ case)

(e) Inverse iteration convergence with different μ ($\mu > 0$ case)

Figure 1. Convergence plots for the power iteration and the inverse iteration

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  : $\mathbf{A} \in \mathbb{C}^{n \times n}$
1 Initilization: $\mathbf{A}^{(0)} = \mathbf{A}$.
2 for $k = 1, \ldots,$ do
3     $\mathbf{Q}^{(k)}\mathbf{R}^{(k)} = \mathbf{A}^{(k-1)}$   % Perform QR for $\mathbf{A}^{(k-1)}$
4     $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$
5 end
  Output: $\mathbf{A}^{(k)}$
```

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

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), 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.

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

Problem 6. (15 points +10 points) This problem is graded by Zhihang Xu (**xuzhh@**).

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

   (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 $\mathbf{A}^{(1)}_{3,1}$ may be nonzero (known as "bulge").

   (b) Compute another Givens rotation $\mathbf{G}_2$ such that $(\mathbf{G}_2^H\mathbf{A}^{(1)})_{3,1} = 0$ (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 $\mathbf{A}^{(2)}_{4,2}$ will now be nonzero.

   (c) Then, we try to find $\mathbf{G}_3$ such that $(\mathbf{G}_3^H\mathbf{A}^{(2)})_{4,2} = 0$. 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. Completed algorithm see in Algorithm 5. In Algorithm 5, the flops for determination of the parameters c_i and s_i is

$$\underbrace{2}_{\text{2 squares}} + \underbrace{1}_{\text{1 addition}} + \underbrace{1}_{\text{1 root}} + \underbrace{2}_{\text{2 divisions}} = 6 \text{ flops},$$

the flops of step i in loop 4-7 is given by

$$\underbrace{3}_{\text{matrix multiplication for each element}} \times \underbrace{2 \times (n-i+1)}_{\text{number of elements in sub-matrix } \mathbf{A}^{(k)}_{i:i+1,i:n}},$$

therefore we can count the complexity of first loop (step 4-7 in Algorithm 5) as

$$\begin{aligned}\sum_{i=1}^{n-1} [3 \times 2 \times (n-i+1) + 6] &= 6(n-1) + \sum_{j=2}^{n} 3 \times 2 \times j \\ &= 6(n-1) + 6\frac{(n+2)(n-1)}{2} = 3n^2 + 9n - 12 \text{ flops},\end{aligned}$$

similarly, the flops for the second loop (step 9-11 in Algorithm 5) is also

$$\sum_{i=1}^{n-1} 3 \times 2 \times (i+1) = \sum_{j=2}^{n} 3 \times 2 \times j = 6\frac{(n+2)(n-1)}{2} = 3n^2 + 3n - 6 \text{ flops}.$$

Altogether, Algorithm 5 takes about $6n^2 + 12n - 18$ flops and the complexity is $\mathcal{O}(n^2)$, which is more efficient than a full matrix QR step.

If neglect the determination of the parameters c_i and s_i, the total flops is

$$\sum_{i=1}^{n-1} 3 \times 2 \times (n-i+1) + \sum_{i=1}^{n-1} 3 \times 2 \times (i+1) = 6n^2 + 6n - 12 \text{ flops}.$$

And the total computational complexity is still $\mathcal{O}(n^2)$.

Remarks. In this problem, you are required to complete the Algorithm 5 and then derive the computational complexity for the Algorithm. The completed Algorithm takes 7 points and the derivation for computation complexity takes 8 points, but even though your derivation is not right, you still will get 4 points for writing down the derivation process.

Grading policy.

1) You will lose half of points if the summation in your derivation is $\sum_{i=1}^{n-1} 6i$. Perhaps you directly utilize the results in http://people.inf.ethz.ch/arbenz/ewp/Lnotes/chapter4.pdf i.e., *Lecture notes on Large-scale eigenvalue computations, Chapter 4.2.2*, but the derivation in this notes is not right. Therefore the derivation is wrong, but the process of derivation itself earns you half of the points.
2) You will lose 4 points if you did not consider the difference in each iteration of the loop. Since the input $\mathbf{A}^{(k)}$ is of Hessenberg form, there is no need to do the matrix multiplication with the zero elements. This *sparsity* property actually makes Gives rotations more suitable. By your computation, the total complexity is about $12n^2$ flops. Therefore even though the final computational complexity is of $\mathcal{O}(n^2)$, the derivation is still wrong. However, the process of derivation itself earns you half of the points.

3) When writting the algorithm, you will get a **Reminder** for the incompleteness of the algorithm/minor mistakes. But basically no points will be taken.
   a) If the $\mathbf{G}_i$ you give is actually $\mathbf{G}_i^H$ in the algorithm.
   b) You lack the initialization step (step 3 in Algorithm 5).
4) When compute the flops, you can neglect the determination of parameters c_i s_i since this step is of $\mathcal{O}(1)$, and $n-1$ repeated determination takes $\mathcal{O}(n)$ and can be eliminated compared to $\mathcal{O}(n^2)$. Also, if you count the flops of determination of parameters c_i s_i as 10, 8 instead of 6, it is also accepted.

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
% This algorithm outputs $\mathbf{A}^{(k)}$ with $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$ where $\mathbf{A}^{(k-1)} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)}$ is
 the QR decomposition obtained via Givens QR.
% Perform QR for $\mathbf{A}^{(k-1)}$ using Givens rotations
$\mathbf{A}^{(k)} = \mathbf{A}^{(k-1)}$.
for $i = 1, \dots, n-1$ do
    $[c_i, s_i] = \text{Givens}(\mathbf{A}^{(k)}_{ii}, \mathbf{A}^{(k)}_{i+1,i})$
    $\mathbf{A}^{(k)}_{i:i+1,i:n} = \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix}^H \mathbf{A}^{(k)}_{i:i+1,i:n}$
end
% Matrix computation
for $i = 1, \dots, n-1$ do
    $\mathbf{A}^{(k)}_{1:i+1,i:i+1} = \mathbf{A}^{(k)}_{1:i+1,i:i+1} \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix}$
end
Output: $\mathbf{A}^{(k)}$
```

Following the steps in implicit QR algorithm,

$$\mathbf{G}_1^H = \begin{bmatrix} \frac{\sqrt{5}}{5} & \frac{2\sqrt{5}}{5} & 0 & 0 \\ -\frac{2\sqrt{5}}{5} & \frac{\sqrt{5}}{5} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G}_1 = \begin{bmatrix} \frac{\sqrt{5}}{5} & -\frac{2\sqrt{5}}{5} & 0 & 0 \\ \frac{2\sqrt{5}}{5} & \frac{\sqrt{5}}{5} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{G}_1^H\mathbf{A}^{(0)} = \begin{bmatrix} \sqrt{5} & \frac{8\sqrt{5}}{5} & \frac{5}{\sqrt{5}} & \frac{8\sqrt{5}}{5} \\ 0 & -\frac{\sqrt{5}}{5} & -\sqrt{5} & -\frac{6\sqrt{5}}{5} \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{A}^{(1)} = \mathbf{G}_1^H\mathbf{A}^{(0)}\mathbf{G}_1 = \begin{bmatrix} \frac{21}{5} & -\frac{2}{5} & \sqrt{5} & \frac{8\sqrt{5}}{5} \\ -\frac{2}{5} & -\frac{1}{5} & -\sqrt{5} & -\frac{6\sqrt{5}}{5} \\ \frac{2\sqrt{5}}{5} & \frac{\sqrt{5}}{5} & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

The bulge appears in $\mathbf{A}^{(1)}_{3,1}$ where we colored red.

$$\mathbf{G}_2^H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{6} & \frac{\sqrt{30}}{6} & 0 \\ 0 & -\frac{\sqrt{30}}{6} & -\frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{6} & -\frac{\sqrt{30}}{6} & 0 \\ 0 & \frac{\sqrt{30}}{6} & -\frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{G}_2^H \mathbf{A}^{(1)} = \begin{bmatrix} \frac{21}{5} & -\frac{2}{5} & \sqrt{5} & \frac{8\sqrt{5}}{5} \\ \frac{2\sqrt{6}}{5} & \frac{\sqrt{6}}{5} & \frac{2\sqrt{30}}{3} & \frac{8\sqrt{30}}{15} \\ 0 & 0 & \frac{\sqrt{6}}{3} & \frac{2\sqrt{6}}{3} \\ 0 & 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{A}^{(2)} = \mathbf{G}_2^H \mathbf{A}^{(1)} \mathbf{G}_2 = \begin{bmatrix} \frac{21}{5} & \frac{9\sqrt{6}}{10} & -\frac{\sqrt{30}}{10} & \frac{8\sqrt{5}}{5} \\ \frac{2\sqrt{6}}{5} & \frac{47}{15} & -\frac{13\sqrt{5}}{\sqrt{15}} & \frac{8\sqrt{30}}{15} \\ 0 & \frac{\sqrt{5}}{3} & -\frac{1}{3} & \frac{2\sqrt{6}}{3} \\ 0 & \frac{\sqrt{30}}{3} & -\frac{\sqrt{6}}{3} & 1 \end{bmatrix}.$$

The bulge appears in $\mathbf{A}^{(2)}_{4,2}$ where we colored red.

$$\mathbf{G}_3^H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{7}}{7} & \frac{\sqrt{42}}{7} \\ 0 & 0 & -\frac{\sqrt{42}}{7} & \frac{\sqrt{7}}{7} \end{bmatrix}, \quad \mathbf{G}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{7}}{7} & -\frac{\sqrt{42}}{7} \\ 0 & 0 & \frac{\sqrt{42}}{7} & \frac{\sqrt{7}}{7} \end{bmatrix}$$

$$\mathbf{G}_3^H \mathbf{A}^{(2)} = \begin{bmatrix} \frac{21}{5} & \frac{9\sqrt{6}}{10} & -\frac{\sqrt{30}}{10} & \frac{8\sqrt{5}}{5} \\ \frac{2\sqrt{6}}{5} & \frac{47}{15} & -\frac{13\sqrt{5}}{15} & \frac{8\sqrt{30}}{15} \\ 0 & \frac{\sqrt{35}}{3} & -\frac{\sqrt{7}}{3} & \frac{5\sqrt{42}}{21} \\ 0 & 0 & 0 & -\frac{3\sqrt{7}}{7} \end{bmatrix}, \quad \mathbf{A}^{(3)} = \mathbf{G}_3^H \mathbf{A}^{(2)} \mathbf{G}_3 = \begin{bmatrix} \frac{21}{5} & \frac{9\sqrt{6}}{10} & \frac{3\sqrt{210}}{14} & \frac{11\sqrt{35}}{35} \\ \frac{2\sqrt{6}}{5} & \frac{47}{15} & \frac{\sqrt{35}}{3} & \frac{\sqrt{210}}{5} \\ 0 & \frac{\sqrt{35}}{3} & \frac{23}{21} & \frac{4\sqrt{6}}{7} \\ 0 & 0 & -\frac{3\sqrt{6}}{7} & -\frac{3}{7} \end{bmatrix}$$

The output $\mathbf{A}^{(3)}$ is still of Hessenberg form.

Using a fractional representation, the above procedure is

$$\mathbf{G}_1^H = \begin{bmatrix} 0.4472 & 0.8944 & 0 & 0 \\ -0.8944 & 0.4472 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G}_1 = \begin{bmatrix} 0.4472 & -0.8944 & 0 & 0 \\ 0.8944 & 0.4472 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{G}_1^H \mathbf{A}^{(0)} = \begin{bmatrix} 2.2361 & 3.5777 & 2.2361 & 3.5777 \\ 0 & -0.4472 & -0.2361 & -2.6833 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{A}^{(1)} = \mathbf{G}_1^H \mathbf{A}^{(0)} \mathbf{G}_1 = \begin{bmatrix} 4.2 & -0.4 & 2.2361 & 3.5777 \\ -0.4 & -0.2 & -2.2361 & -2.6833 \\ 0.8944 & 0.4472 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}.$$

The bulge appears in $\mathbf{A}^{(1)}_{3,1}$ where we colored red.

$$\mathbf{G}_2^H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.4082 & 0.9129 & 0 \\ 0 & -0.9129 & -0.4082 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.4082 & -0.9129 & 0 \\ 0 & 0.9129 & -0.4082 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{G}_2^H \mathbf{A}^{(1)} = \begin{bmatrix} 4.2 & -0.4 & 2.2361 & 3.5777 \\ 0.9798 & 0.4899 & 3.6515 & 2.9212 \\ 0 & 0 & 0.8165 & 1.6330 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{A}^{(2)} = \mathbf{G}_2^H \mathbf{A}^{(1)} \mathbf{G}_2 = \begin{bmatrix} 4.2 & 2.2045 & -0.5477 & 3.5777 \\ 0.9798 & 3.1333 & -1.9397 & 2.9212 \\ 0 & 0.7454 & -0.3333 & 1.6330 \\ 0 & 1.8257 & -0.8165 & 1 \end{bmatrix}.$$

The bulge appears in $\mathbf{A}^{(2)}_{4,2}$ where we colored red.

$$\mathbf{G}_3^H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.3780 & 0.9258 \\ 0 & 0 & -0.9258 & 0.3780 \end{bmatrix}, \quad \mathbf{G}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.3780 & -0.9258 \\ 0 & 0 & 0.9258 & 0.3780 \end{bmatrix}$$

$$\mathbf{G}_3^H \mathbf{A}^{(2)} = \begin{bmatrix} 4.2 & 2.2045 & -0.5477 & 3.5777 \\ 0.9798 & 3.1333 & -1.9379 & 2.9212 \\ 0 & 1.9720 & -0.8819 & 1.5430 \\ 0 & 0 & 0 & -1.1339 \end{bmatrix}, \quad \mathbf{A}^{(3)} = \mathbf{G}_3^H \mathbf{A}^{(2)} \mathbf{G}_3 = \begin{bmatrix} 4.2 & 2.2045 & 3.1053 & 1.8593 \\ 0.9798 & 3.1333 & 1.9720 & 2.8983 \\ 0 & 1.9720 & 1.0952 & 1.3997 \\ 0 & 0 & -1.0498 & -0.4286 \end{bmatrix}.$$

To explain, we can rewrite Algorithm 5 in this way, first corresponding to the first loop of Algorithm 5 (step 4-7),

$$\mathbf{A}^{(k)} = \mathbf{G}_{n-1}^H \mathbf{G}_{n-2}^H \cdots \mathbf{G}_1^H \mathbf{A}^{(k-1)} = \mathbf{R}^{(k)}$$

where $\mathbf{G}_1, \ldots, \mathbf{G}_{n-1}$ is a sequence of Givens rotations, $\mathbf{Q}^{(k)} = \mathbf{G}_1 \cdots \mathbf{G}_{n-1}$, then

$$\mathbf{A}^{(k)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)} = \mathbf{A}^{(k)} \mathbf{G}_1 \cdots \mathbf{G}_{n-1}.$$

Equivalently, rewrite the implicit QR as, for $i = 1, \ldots, n-1$

$$\mathbf{A}^{(k)} = \mathbf{G}_i^H \mathbf{A}^{(k)},$$
$$\mathbf{A}^{(k)} = \mathbf{A}^{(k)} \mathbf{G}_i.$$

Therefore, the algorithms are indeed equivalent.

Remarks: In this problem, first, you are required to give detailed derivation for carrying out the implicit QR for $\mathbf{A}^{(0)}$ and give $\mathbf{G}_i, \mathbf{G}_i^H \mathbf{A}^{i-1}$ and $\mathbf{A}^{(i)}$ for $i = 1, 2, 3$ explicitly. The detailed derivation takes 4 points and the pointing out the bulge takes 1 point. Secondly, you are required to state the equivalence between the implicit QR and the Algorithm 5.

Grading policy:

1) You will lose 5 points if you did not give the detailed derivation for carrying out the implicit QR step for $\mathbf{A}^{(0)}$. Here the detailed derivation means explicitly giving $\mathbf{G}_i$, $\mathbf{G}_i^H\mathbf{A}^{(i-1)}$ etc, which is clearly stated in the problem description.
2) Basically, in the explanation for equivalence, if your explanation states that $\mathbf{G}_i$ in implicit QR is the same as the ones in Algorithm 5, and the two algorithms only changes the order of multiplication, you will get 5 points.