SI231 - Matrix Computations, Fall 2020-21

Homework Set #4

Prof. Yue Qiu and Prof. Ziping Zhao

Name: Tian Yun **Major:** Master in CS

Student No.: 2019232102 **E-mail:** tianyun@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 $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ 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) Solution :

   $\mathbf{A}$ can be diagonalized

   Assume $|\lambda I - A| = 0$, $det(\lambda I - A) = (\lambda+4)(\lambda-5)+18 = \lambda^2 - \lambda - 2 = (\lambda-2)(\lambda+1)$

   $\lambda_1 = 2, \quad \lambda_2 = -1$

   if $\lambda_1 = 2$,$(2E - A)x = 0$,so the corresponding eignvector will be $\xi_1 = [1 - 2]^T$

   if $\lambda_1 = -1$,$(-E - A)x = 0$,so the corresponding eignvector will be $\xi_1 = [1 - 1]^T$

$$\mathbf{V} = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix} \mathbf{\Lambda} = \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix}$$

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

2) Denote a matrix $\mathbf{B}$ is similar to $\mathbf{A}$, so we have $\mathbf{A} - \mathbf{P}^{-1}\mathbf{B}\mathbf{P}, \mathbf{P}$ is a invertible matrix.

   Let $\mathbf{v}$ be an eigevector of A with eigenvvalue λ.

   So $\left(\mathbf{P}^{-1}\mathbf{B}\mathbf{P}\right)\mathbf{v} = \mathbf{P}^{-1}\mathbf{B}(\mathbf{P}\mathbf{v}) = \lambda\mathbf{v}$,

   it means that $\mathbf{B}(\mathbf{P}\mathbf{v}) = \mathbf{P}(\lambda\mathbf{v}) = \lambda\mathbf{P}\mathbf{v}$ matrix has different eigenspaces with A.

   Let $\mathbf{P} = \begin{bmatrix} 1 & 5 \\ 2 & 1 \end{bmatrix}$,$\mathbf{B} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1} = \begin{bmatrix} -\frac{4}{3} & \frac{5}{3} \\ -\frac{2}{3} & \frac{7}{3} \end{bmatrix}$, the eigenvector of $\mathbf{B}$ are $\mathbf{P}\xi_1$ and $\mathbf{P}\xi_2$

   So the matrix has different eigenspaces with $\mathbf{A}$.

Problem 2. (6 points × 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) **Solution**:

   If 0 is an eigenvalue of A, then $Ax = 0 \cdot x = 0$ for some non-zero x, which clearly means A is singular. On the other hand, if A is singular, then $Ax = 0$ for some non-zero x, which is to say that $Ax = 0 \cdot x$ for some non-zero x, which obviously means that 0 is an eigenvalue of A.

2) **Proof**:

   If a matrix A is diagonalizable, then the algebraic multiplicity of any eigenvalue of A is equal to its geometric multiplicity. In particular, if A is symmetric (Hermitian) or all the eigenvalues of A are distinct, then A is diagonalizable. From rank-nullity theorem, it is known that

   $$\operatorname{rank}(A) + \text{ nullity } (A) = \operatorname{dim}(A)$$

   Nullity is the dimension of the kernel space of A. Meaning the number of linearly independent eigenvectors x for which $Ax = 0 \cdot x$. So nullity in this case implies the multiplicity of 0 as an eigenvalue of A and hence rank implies the number of nonzero eigenvalues of A.

3) **Proof**:

   ****

   Proposition. If A is $n \times m$ and B is $m \times k$, then

   $$\operatorname{rank}(\mathbf{AB}) \leq \min(\operatorname{rank}(\mathbf{A}), \operatorname{rank}(\boldsymbol{B}))$$

   since $(AB)_{j,l} = \sum_{v=1}^{n} A_{j,v} B_{v,l}$, the rows of AB are linear combinations of the rows of B that is $R(AB) \subseteq R(B)$, whence $\operatorname{rank}(AB) \leq \operatorname{rank}(B)$. Also, $G(AB) \subseteq G(A)$, whence $\operatorname{rank}(AB) \leq \operatorname{rank}(A)$

   ****

   $\mathbf{A}$ admits an eigendecomposition (eigenvalue decomposition).We write A, in spectral form $A = V\Lambda V^{-1}$, ; our goal is to show that $\operatorname{rank}(A) = \operatorname{rank}(\Lambda)$ rank $(A) \leq \operatorname{rank}(V\Lambda) \leq \operatorname{rank}(\Lambda)$. And since $\Lambda = V^{-1}AV$, we have also $\operatorname{rank}(\Lambda) \leq \operatorname{rank}\left(V^{-1}A\right) \leq \operatorname{rank}(A)$, so rank(A) = rank($\Lambda$)

   $\mathsf{rank}(\mathbf{A})$ = number of nonzero eigenvalues of Λ = number of nonzero eigenvalues of $\mathbf{A}$

4) **Proof**:

Let A be a Hermitian matrix. Then, by definition, $\mathbf{A} = \mathbf{A}^\dagger$, where † designates the conjugate transpose. Let λ be an eigenvalue of $\mathbf{A}$. Let $\mathbf{v}$ be an eigenvector corresponding to the eigenvalue λ of $\mathbf{A}$. Denote with $\langle \cdot, \cdot \rangle$ the inner product on $\mathbb{C}$.

$$\lambda * \langle v, v \rangle = \langle \lambda * v, v \rangle \quad \text{Properties of Complex Inner Product}$$

Definition of Eigenvector: $\lambda * v = \mathbf{A} * v$

$$= \langle v, \mathbf{A} * v \rangle$$

$\mathbf{A}$ is Hermitian, so $\mathbf{A}^\dagger = \mathbf{A}$

$$= \langle v, \lambda * v \rangle \quad \text{Definition of Eigenvector: } \lambda * v = \mathbf{A} * v$$
$$= \bar{\lambda} * \langle v, v \rangle \quad \text{Properties of Complex Inner Product}$$

We have that $v \neq 0$, and because of the positive definiteness, it must be that:

$$\langle v, v \rangle \neq 0$$

Thus:

$$\langle v, v \rangle \neq 0$$

So we can divide both sides by $\langle v, v \rangle$ Thus $\lambda = \bar{\lambda}$. By Complex Number equals Conjugate iff Wholly Real, λ is a real number. λ was arbitrary, so it follows that every eigenvalue is a real number. Hence the result.

5) **Proof**:

Suppose that $Au = \lambda u$ and $Av = \mu v$ for $\lambda \neq \mu$. Then

$$\lambda \langle u, v \rangle = \langle Au, v \rangle = \langle u, Av \rangle = \langle u, \mu v \rangle = \mu \langle u, v \rangle$$
66

since $\mu \neq \lambda$ it follows that $\langle u, v \rangle = 0$. Suppose that A has n distinct eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ with corresponding orthogonal eigenvectors $\{u_1, \ldots, u_n\}$. Let us also agree to scale the eigenvectors so that

$$\langle u_i, u_j \rangle = \delta_{ij}$$

where δ_{ij} is the so-called Kronecker delta

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

We recall that the eigenvalue equation can be written in matrix form

$$AU = U\Lambda$$

where $U = (u_1 u_2 \ldots u_n)$ and $\Lambda = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\}$. We now observe from the orthonor- mality of the eigenvectors that

$$U^* U = I$$

Hence $U^{-1} = U^*$ and consequently

$$A = U\Lambda U^*$$

In other words, A can be diagonalized in a particularly simple way.

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. Since A is a symmetric matrix, according to the nature of the eigendecomposition of the symmetric matrix, select its eigenvectors $\{u_1, \ldots, u_n\}$ as a group Orthogonal basis. Knowledge about subspace and basis That is, this is the n basis of the n dimensional space. Now, if there is a subspace U of the n dimensional space, its dimension is k, and the subspace $\operatorname{span}(u_k, \ldots, u_n)$, must have an intersection.

This can actually be proved as follows:

The dimension of U is k, and the dimension of span $(u_k, \ldots, u_n)$ is $n-k+1$. In other words, the sum of the dimensions of the two is greater than $n_{\circ}$ Therefore, there must be a non-zero intersection.

Therefore, suppose v is an element on the intersection, that is, it belongs to both subspace U and subspace $\operatorname{span}(u_k, \ldots, u_n)_{\circ}$ Then, $v \in \operatorname{span}(u_k, \ldots, u_n)$, therefore has

$$x = \sum_{i=k}^{n} \alpha_i u_i$$

(Since $\|x\| = 1$, has $\sum_{i=k}^{n} \alpha_i = 1)$ Then,

$$x^H\mathbf{A}x = \sum_{i=k}^{n} \alpha_i^2 u_i^H \mathbf{A} u_i = \sum_{i=k}^{n} \lambda_i \alpha_i^2 \geq \lambda_i$$

which is:

$$\max_{x\in U, \|x\|=1} x^H\mathbf{A}x \geq \lambda_i$$

It is true for all subspaces U. which is:

$$\min_{\dim(U)=k} \max_{x\in U, \|x\|=1} x^H\mathbf{A}x \geq \lambda_i$$

At this time, we prove the other half: Obviously, the space $V = \operatorname{span}\{u_1, \ldots, u_k\}$ is the selected k dimensional space,

$$x^H\mathbf{A}x \leq \lambda_i$$

This conclusion is too obvious to explain. In other words,

$$\max_{x\in V, \|x\|=1} x^H\mathbf{A}x \leq \lambda_i$$

And V is obviously one of the subspaces U of k dimension, so:

$$\min_{\dim(U)=k} \max_{x\in U, \|x\|=1} x^H\mathbf{A}x \leq \lambda_i$$

Finally:

$$\min_{\dim(U)=k} \max_{x\in U, \|x\|=1} x^H\mathbf{A}x = \lambda_i$$

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)

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

2) Proof: First, we show that 0 is an eigenvalue of L using the vector $x = D^{-1/2}e$. Then

$$\left(D^{1/2}e\right) = D^{-1/2}L_G D^{-1/2}D^{1/2}e = D^{-1/2}L_G e = 0$$

since e is a eigenvector of L_G corresponding to eigenvalue 0. This shows that $D^{1/2}e$ is an eigenvector of L of eigenvalue 0. To show that it's the smallest eigenvalue, notice that L is positive semidefinite.
So $\lambda_n = 0$.
Notice that $x^T(I + A)x \geq 0$ implies the following chain:

$$-x^T Ax \leq x^T x \Longrightarrow x^T Ix - x^T Ax \leq 2x^T x \Longrightarrow \frac{x^T Lx}{x^T x} \leq 2 \Longrightarrow \lambda_n \leq 2$$

3) Let $\mathbf{L} \in \mathbb{R}^{n\times n}$ 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 $\mathbf{L}$. For any $k \in \{1, 2, 3, ..., n\}$, we can get that

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

If $\lambda_k = 0$, then there exists a $n-k+1$ dimensional subspace $\mathcal{S}$ such that $\forall\mathbf{x} \in \mathcal{S}$, we have $\sum_{(i,j)\in E} \left(\mathbf{x}_i - \mathbf{x}_j\right)^2 = 0$. It also implies that $\forall\mathbf{x}, \mathbf{x}_i = \mathbf{x}_j \quad \forall(i, j) \in E$ with positive weight. Therefore, for any vertices i, j in the same component, $\mathbf{x}_i = \mathbf{x}_j$. This means that $\mathbf{x} \in \mathcal{S}$ is constant within each connected component of G. Then $n - k + 1 = \dim(\mathcal{S}) \leq$ the number of connected components.
Conversely,
if G has $\geq n - k + 1$ connected components, then we let $\mathcal{S}$ be the space of vectors that are constant on each component and this $\mathcal{S}$ has dimension $\geq n-k+1$. Furthermore, $\mathbf{x} \in \mathcal{S}$ we have that $\sum_{(i,j)\in E} \left(\mathbf{x}_i - \mathbf{x}_j\right)^2 = 0$. Thus $\max_{\mathbf{x}\in\mathcal{S}_{n-k+1}\setminus\{0\}} \frac{\mathbf{x}^T\mathbf{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = 0$ for any dimension $n - k + 1$ subspace $\mathcal{S}_{n-k+1}$ of the $\mathcal{S}$ we choose.
Then it is Courant-Fischer form $\lambda_k = 0$ as any $\mathcal{S}_{n-k+1}$ provides an upper bound, i.e.,

$$\lambda_k = \min_{\mathcal{S}\subseteq\mathbb{R}^n \dim(\mathcal{S})=n-k+1 \mathbf{x}\in\mathcal{S}\setminus\{0\}} \frac{\mathbf{x}^T\mathbf{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \min_{\mathcal{S}\subseteq\mathbb{R}^n \dim(\mathcal{S})=n-k+1} 0 = 0$$

Finally, we prove the graph G has at least $(n - k + 1)$ connected components if and only if $\lambda_k = 0$.

III. Eigenvalue Computations

A. Power Iteration

Problem 5. (20 points)

Consider the 2×2 matrix $\mathbf{A}$

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

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

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

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

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

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

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

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

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

end

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

Solution

1) Solution :

   $\mathbf{A}$ can be diagonalized

   Assume $|\lambda I - A| = 0$, $det(\lambda I - A) = \lambda^2 - \alpha\beta$

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

   if $\lambda = \sqrt{\alpha\beta}$,$(\sqrt{\alpha\beta}E - A)x = 0$,so the corresponding eignvector will be $\xi = [1 \quad \sqrt{\frac{\beta}{\alpha}}]^T$

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

   if $\lambda = -\sqrt{\alpha\beta}$,$(-\sqrt{\alpha\beta}E - A)x = 0$,so the corresponding eignvector will be $\xi = [-1 \quad \sqrt{\frac{\beta}{\alpha}}]^T$

2) set A as

$$\mathbf{A} = \begin{bmatrix} 0 & 4.6 \\ 3.4 & 0 \end{bmatrix}, \qquad \text{with } \alpha, \beta > 0\,.$$

   power iteration can not converge. You can see figure 1.

   inverse iteration can converge. You can see and figure 2.

   Some analysis:

   a) $v_0 = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n \quad (\alpha_1 \neq 0)$

$$v_1 = Av_0 = \alpha_1\lambda_1 x_1 + \alpha_2\lambda_2 x_2 + \cdots + \alpha_n\lambda_n x_n$$
$$v_k = Av_{k-1} = \alpha_1\lambda_1^k x_1 + \alpha_2\lambda_2^k x_2 + \cdots + \alpha_n\lambda_n^k x_n$$
$$= \lambda_1^k \left[\alpha_1 x_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1}\right)^k x_n\right]$$

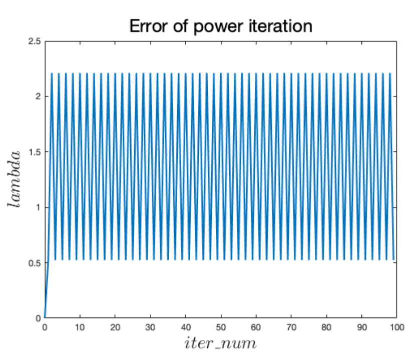
   The smaller the value of $|\frac{\lambda_1}{\lambda_2}|$, the faster the convergence will come. If the absolute value of the two eigenvalues is the same, then it will not converge. If the eigenvalue difference is relatively small, the convergence will be very slow.

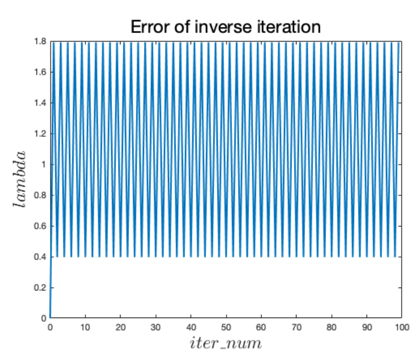
   For example:

   set A as

$$\mathbf{A} = \begin{bmatrix} 0 & 5 \\ 1 & 0 \end{bmatrix}$$

   $\lambda_1 = 2.23$ $\lambda_2 = -2.23$





   Iteration cannot converge, as you can see in figure 3.

   b) For the inverse power method, mu should not be placed in the middle of the two eigenvalues.

set A as

$$\mathbf{A} = \begin{bmatrix} 0 & 5 \\ 1 & 0 \end{bmatrix}$$

$\lambda_1 = 2.23$ $\lambda_2 = -2.23, \mu = 0$ Iteration cannot converge, as you can see in figure 4.

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 $k = 1, \dots,$ **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** $k = 1, \dots,$ **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 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 $\mathbf{A}^{(1)}_{3,1}$ 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 $\mathbf{A}^{(2)}_{4,2}$ 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`

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

for $k = 1 : n-1$ **do**

$$\mathbf{A}(k : k+1, k : n) = \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}^T \mathbf{A}(k : k+1, k : n)$$

end

`% Matrix computation`

for $k = 1 : n-1$ **do**

$$H(1 : k+1, k : k+1) = H(1 : k+1, k : k+1) \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}$$

end

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

1.(b) Perform matrix multiplication such that $\mathbf{A}^{(k)}$ is of Hessenberg form, Every time we perform householder transformation, we only need to eliminate the elements of the diagonal line one by one.

Assuming that A has n rows, then it needs to be eliminated n-1 times given elimination

Every Givens elimination we multiplies A by 2×2 rotation matrix, we need to caculate the rotation matrix as following:

$Q = I - 2vv^T$, $v = ||a_i||_2 - a_i$, $v = \frac{v}{||v||_2}$. it costs const flops.

1. rotation matrix cost 10 flops.

2. $\begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}^T \mathbf{A}(k : k+1, k : n)$ cost n flops,as data is accessed across rows

3. $H(1 : k+1, k : k+1) \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}$ cost n flops,as data is accessed across rows

All in all total cost is $n * (10 + n + c) + n * n$. so is $\mathcal{O}(n^2)$

2.

$$\mathbf{G_1^T} = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 & 0 \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{G_1^T A^{(0)}} = \begin{bmatrix} \sqrt{5} & \frac{8}{\sqrt{5}} & \sqrt{5} & \frac{8}{\sqrt{5}} \\ 0 & -\frac{1}{\sqrt{5}} & -\sqrt{5} & -\frac{6}{\sqrt{5}} \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix} \mathbf{A^{(1)}} = \mathbf{G_1^T A^{(0)} G_1} = \begin{bmatrix} \frac{21}{5} & -\frac{2}{5} & \sqrt{5} & \frac{8}{\sqrt{5}} \\ -\frac{2}{5} & -\frac{1}{5} & -\sqrt{5} & -\frac{6}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

The "bulge" is $\frac{2}{\sqrt{5}}$.

$$\mathbf{G_2^T}=\begin{bmatrix}1&0&0&0\\0&-\frac{1}{\sqrt{6}}&\frac{\sqrt{30}}{6}&0\\0&-\frac{1}{\sqrt{6}}&-\frac{1}{\sqrt{6}}&0\\0&0&0&1\end{bmatrix}\mathbf{G_2^T A^{(1)}}=\begin{bmatrix}\frac{21}{5}&-\frac{2}{5}&\sqrt{5}&\frac{8}{\sqrt{5}}\\\frac{2\sqrt{6}}{5}&\frac{\sqrt{6}}{5}&\frac{2\sqrt{30}}{3}&\frac{8\sqrt{20}}{15}\\0&0&\frac{6}{3}&\frac{2\sqrt{6}}{3}\\0&0&2&1\end{bmatrix}\mathbf{A^{(2)}}=\mathbf{G_2^T A^{(1)} G_2}=\begin{bmatrix}\frac{21}{5}&\frac{9\sqrt{6}}{10}&-\frac{\sqrt{30}}{10}&\frac{8}{\sqrt{5}}\\\frac{2\sqrt{6}}{5}&\frac{47}{15}&-\frac{13\sqrt{5}}{15}&\frac{8\sqrt{30}}{15}\\0&\frac{\sqrt{5}}{3}&-\frac{1}{3}&\frac{2\sqrt{6}}{3}\\0&\frac{\sqrt{30}}{3}&-\frac{\sqrt{6}}{3}&1\end{bmatrix}$$

The "bulge" is $\frac{\sqrt{30}}{3}$.

$$\mathbf{G_3^T}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&-\frac{1}{\sqrt{7}}&-\frac{\sqrt{42}}{7}\\0&0&\frac{\sqrt{42}}{7}&-\frac{1}{\sqrt{7}}\end{bmatrix}\mathbf{G_3^T A^{(2)}}=\begin{bmatrix}\frac{21}{5}&\frac{9\sqrt{6}}{10}&\frac{30}{10}&\frac{8}{\sqrt{5}}\\\frac{2\sqrt{6}}{5}&\frac{47}{15}&-\frac{13\sqrt{5}}{15}&\frac{8\sqrt{30}}{15}\\0&-\frac{\sqrt{35}}{3}&\frac{7}{3}&\frac{5\sqrt{42}}{21}\\0&0&0&\frac{3\sqrt{7}}{7}\end{bmatrix}\mathbf{A^{(3)}}=\begin{bmatrix}\frac{21}{5}&\frac{9\sqrt{6}}{10}&-\frac{3\sqrt{15}}{\sqrt{14}}&\frac{11}{\sqrt{35}}\\\frac{2\sqrt{6}}{5}&\frac{47}{15}&-\frac{\sqrt{35}}{3}&\frac{\sqrt{42}}{\sqrt{5}}\\0&-\frac{\sqrt{35}}{3}&\frac{23}{21}&\frac{4\sqrt{6}}{7}\\0&0&-\frac{3\sqrt{6}}{7}&-\frac{3}{7}\end{bmatrix}$$

Now, the matrix $\mathbf{A^{(3)}}$ is the Hessenberg form. According to the computational process, we have

$$\mathbf{A^{(3)}}=\mathbf{G_1^T G_2^T G_3^T A^{(0)} G_1 G_2 G_3}=\mathbf{(G_1 G_2 G_3)^T A^{(0)} (G_1 G_2 G_3)}$$

And the $\mathbf{A^0}$ corresponding to $\mathbf{A^{(k-1)}}$ of Algorithm 5, the $\mathbf{A^3}$ corresponding to $\mathbf{A^{(k)}}$ of Algorithm 5. $\mathbf{G_1^T G_2^T G_3^T A^{(0)}}$ corresponding to $\mathbf{R^{(k)}}$ of Algorithm 5.

By definition of Hessenberg matrix, $\mathbf{A}^{(3)}$ is Hessenberg matrix. preserved Hessenberg form. I will show the implicit QR iteration is equivalent to the Algorithm 5. In Algorithm 5 ,

$$\begin{cases}\left(\mathbf{G}_{n-1}^H\mathbf{G}_{m-2}^H\cdots\mathbf{G}_1^H\right)\mathbf{A}^{(k-1)}=\left(\mathbf{Q}^{(k)}\right)^H\mathbf{A}^{(k-1)}=\mathbf{R}^{(k)}\\ \mathbf{A}^{(k)}=\mathbf{R}^{(k)}\mathbf{Q}^{(k)}=\mathbf{R}^{(k)}\mathbf{G}_1\cdots\mathbf{G}_{n-1}\end{cases}$$

We can rewrite it as

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

In implicit QR iteration, we have

$$\widetilde{\mathbf{A}}^{(1)}=\mathbf{G}_1^H\mathbf{A}^{(k-1)}\mathbf{G}_1$$
$$\widetilde{\mathbf{A}}^{(2)}=\mathbf{G}_2^H\widetilde{\mathbf{A}}^{(1)}\mathbf{G}_2$$
$$\cdots$$
$$\widetilde{\mathbf{A}}^{(n-1)}=\mathbf{G}_{n-1}^H\widetilde{\mathbf{A}}^{(n-2)}\mathbf{G}_{n-1}$$