LECTURE 24

Consider a sequence of vectors \( \{q_k\} \) in \( \mathbb{C}^n \) such that
\[
\lim_{k \to \infty} q_k = q_*
\]

**DEFN (Linear Convergence)**

We say \( \{q_k\} \) converges to \( q_* \) linearly if there exists a constant \( c \in [0,1) \) s.t.
\[
\lim_{k \to \infty} \frac{\|q_{k+1} - q_*\|}{\|q_k - q_*\|} = c.
\]

**EXAMPLE**

\( \{2^{-k}\} = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots \} \) converges to 0 linearly, i.e.
\[
\lim_{k \to \infty} \frac{12^{-(k+1)} - 0}{12^{-k} - 0} = \frac{1}{2}
\]

**DEFN (Q-Rate of Convergence)**

We say that the q-rate of convergence is p if there exists a constant \( c > 0 \) s.t.
\[
\lim_{k \to \infty} \frac{\|q_{k+1} - q_*\|}{\|q_k - q_*\|^p} = c.
\]
q-quadratic convergence when $p=2$
q-cubic convergence when $p=3$

**EXAMPLE**

\[
\left\{ 2^{-2^k} \right\} = \left\{ 2^{-2}, 2^{-4}, 2^{-8}, 2^{-16}, \ldots \right\}
\]

converges to 0 q-quadratically, i.e.

\[
\lim_{k \to \infty} \frac{|2^{-2^k} - 0|}{|2^{-2^k} - 0|^2} = \lim_{k \to \infty} \frac{|2^{-2^k+1}|}{|2^{-2^k} \cdot 2^{-2^k}|} = \frac{2^{-2 \cdot 2^k}}{2^{-2^{k+1}}} = 2^{-2^{k+1}}
\]

**RATE OF CONVERGENCE FOR POWER ITERATION**

Recall the estimate generated by power iteration of the form

\[
q_k = \frac{\lambda_1^k}{1 |\lambda_1|^k} \frac{c_1 v_1 + c_2 (\frac{\lambda_2}{\lambda_1})^k v_2 + \ldots + c_n (\frac{\lambda_n}{\lambda_1})^k v_n}{\| c_1 v_1 + c_2 (\frac{\lambda_2}{\lambda_1})^k v_2 + \ldots + c_n (\frac{\lambda_n}{\lambda_1})^k v_n \|}
\]

As $k \to \infty$

\[
\left\| q_k - \frac{\lambda_1^k}{1 |\lambda_1|^k} \frac{c_1 v_1}{\| c_1 v_1 \|} \right\| \to \frac{|c_2|}{\| c_1 v_1 \|} \left( \frac{\lambda_2}{\lambda_1} \right)^k \| v_2 \|
\]

Consequently

\[
\lim_{k \to \infty} \frac{\| q_{k+1} - \tilde{c}_{k+1} \frac{v_1}{\|v_1\|} \|}{\| q_k - \tilde{c}_k \frac{v_1}{\|v_1\|} \|} = \frac{|\lambda_2|}{|\lambda_1|}
\]

\[\text{(2)}\]
where \( \{ \hat{c}_k \} \) denotes a sequence of complex signs (i.e. \( \hat{c}_k = e^{i\theta_k} \) for some \( \theta_k \)).

**REMARKS**

* Power iteration converges linearly.
* Convergence is especially slow when \( |\lambda_1| \approx |\lambda_2| \).

**INVERSE ITERATION**

Suppose \( \sigma \) is a good estimate for an eigenvalue \( \lambda_j \) of \( A \).

Let's say the eigenvalue \( \lambda_k \), second closest to \( \sigma \) is significantly further away, i.e.

\[
|\lambda_j - \sigma| \ll |\lambda_k - \sigma|.
\]

**BASIC OBSERVATION**

Suppose \( (\lambda, \nu) \) is an eigenpair of \( A \). Then

\[
A\nu = \lambda\nu \iff (A - \sigma I)\nu = (\lambda - \sigma)\nu
\]

\[
\iff (A - \sigma I)^{-1}\nu = (\lambda - \sigma)^{-1}'\nu
\]

That is \( ((\lambda - \sigma)^{-1}, \nu) \) is an eigenpair of \( (A - \sigma I)^{-1} \).

Consequently, the largest eigenvalue (in modulus) of \( (A - \sigma I)^{-1} \) is significantly larger than second largest, i.e.

\[
\frac{1}{|\lambda_j - \sigma|} \gg \frac{1}{|\lambda_k - \sigma|}
\]

The largest eigenvalue of \( (A - \sigma I)^{-1} \) is the second largest.
Inverse iteration is power iteration applied to \((A - \sigma I)^{-1}\)

**Algorithm (Inverse Iteration)**

* Given \(A \in \mathbb{C}^{n \times n}, q_0 \in \mathbb{C}^n\) s.t. \(\|q_0\| = 1\) and \(\sigma \in \mathbb{C}\)

* Produce an estimate for the eigenpair \((\lambda_j, v_j)\) where \(\lambda_j\) is the eigenvalue of \(A\) closest to \(\sigma\).

Compute an LU factorization of \((A - \sigma I)\)

for \(k = 1, \ldots, m\)

\[ \text{forward} \]
\[ L \hat{x} = q_{k-1} \]
\[ \text{backward substitution} \]
\[ U \hat{x} = \hat{x} \]

\[ q_k = \hat{x} / \|\hat{x}\| \]

end

\(v \leftarrow q_m, \quad \lambda = q_m^* A q_m\)

**Remarks**

* At the \(k\)th iteration the linear system \((A - \sigma I)x = q_{k-1}\) must be solved.

* This can be done efficiently if an LU factorization for \((A - \sigma I)\) is computed initially.
Inverse iteration is commonly used to compute eigenvectors given eigenvalues.

Rate of convergence

Still linear, but with a small constant.

\[
\lim_{k \to \infty} \frac{\|q_{k+1} - \tilde{v}_{k+1} \|}{\|q_k - \tilde{v}_k \|} = \frac{1}{\lambda_k - \sigma} = \frac{1}{\lambda_j - \sigma}
\]

RAYLEIGH ITERATION

Rayleigh quotient

\[ r(q) = \frac{q^* A q}{q^* q} \]

gives a very good estimate for an eigenvalue of \( A \) if \( q \) is close to an eigenvector of \( A \).

In particular

\[ r(v_j) = \frac{v_j^* A v_j}{v_j^* v_j} = \frac{v_j^* (\lambda_j v_j)}{v_j^* v_j} = \lambda_j \]

where \((\lambda_j, v_j)\) is an eigenpair.
THEM (Accuracy of Rayleigh Quotient)

Let \((\lambda_j, v_j)\) be an eigenpair of \(A \in \mathbb{C}^{n \times n}\)
and \(q \in \mathbb{C}^n\) s.t. \(\|v_j\| = \|q\| = 1\). Then
\[
|r(q) - \lambda_j| \leq 2 \|A\| \|q - v_j\|
\]

**Proof**

\[
r(q) - \lambda_j = r(q) - r(v_j)
\]
\[
= q^* A q - v_j^* A v_j
\]
\[
= (q^* A q - q^* A v_j) + (q^* A v_j - v_j^* A v_j)
\]
\[
= q^* A (q - v_j) + (q - v_j)^* A v_j
\]

\[
\Rightarrow (\text{By triangular inequality and submultiplicative properties})
\]
\[
|r(q) - \lambda_j| \leq |q^* A (q - v_j)| + |(q - v_j)^* A v_j|
\]
\[
\leq \|q^*\| \|A\| \|q - v_j\| + \|(q - v_j)^*\| \|A\| \|v_j\|
\]
\[
= 2 \|A\| \|q - v_j\|
\]

\[
\square
\]

Rayleigh iteration is the inverse iteration but with shifts chosen as Rayleigh quotients.
ALGORITHM (Rayleigh Iteration)

* Given $A \in \mathbb{C}^{n \times n}$ and $q_0 \in \mathbb{C}^n$ s.t. $\|q_0\|_2 = 1$.
* Produce an estimate $(\lambda, v)$ for an eigenpair of $A$.

for $k = 1, \ldots, m$

$$\sigma_{k-1} = q_{k-1}^* A q_{k-1}$$

Solve $(A - \sigma_{k-1} I) x = q_{k-1}$ for $x$

$$q_k = x/\|x\|$$

end

$v = q_m$, $\lambda = q_m^* A q_m$

Rate of Convergence

Suppose $\text{span}\{q_k\} \to \text{span}\{v_j\}$ as $k \to \infty$.

Then

$$\lim_{k \to \infty} \frac{\|q_{k+1} - \tilde{c}_{k+1} \frac{v_j}{\|v_j\|}\|}{\|q_k - \tilde{c}_k \frac{v_j}{\|v_j\|}\|} = c,$$

that is the $q$-rate of convergence is quadratic.

JUSTIFICATION

$$\frac{\|q_{k+1} - \tilde{c}_{k+1} \frac{v_j}{\|v_j\|}\|}{\|q_k - \tilde{c}_k \frac{v_j}{\|v_j\|}\|} = \frac{|r(q_{k+1}) - \lambda_j|}{|r(q_k) - \lambda_j|} = O(\|q_k - \tilde{c}_k \frac{v_j}{\|v_j\|}\|_{\text{second closest to } r(q_k)}).$$
QR ALGORITHM

Given a Hessenberg matrix $A \in \mathbb{C}^{n \times n}$.
Generates a sequence $\{A_k\}$ satisfying

(i) $A_0 = A$

For $k \geq 0$, (ii) $A_k = Q_{k+1} R_{k+1}$ and $A_{k+1} = R_{k+1} Q_{k+1}$

QR Factorization of $A_k$

ALGORITHM (QR Algorithm)

* Given $A \in \mathbb{C}^{n \times n}$ in Hessenberg form.
* Produce a sequence $\{A_k\}$ such that typically $\lim_{k \to \infty} A_k$ is upper triangular.

$A_0 = A$

for $k = 0, 1, \ldots$

Compute a QR factorization $A_k = Q_{k+1} R_{k+1}$

end

EXAMPLE

$A = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$ has the eigenvalues $\lambda_1 = 5$, $\lambda_2 = -1$
Apply the QR algorithm to \( \mathbf{A} \)

\[
\mathbf{A}_0 = \mathbf{A} = \begin{bmatrix}
-0.83 & -0.55 \\
-0.55 & 0.83 \\
\end{bmatrix} \begin{bmatrix}
-3.61 & -3.88 \\
0 & -1.39 \\
\end{bmatrix}
\]

\[
\mathbf{Q}_1, \quad \mathbf{R}_1
\]

\[
\mathbf{A}_1 = \mathbf{R}_1 \mathbf{Q}_1 = \begin{bmatrix}
5.15 & -1.23 \\
0.77 & -1.15 \\
\end{bmatrix}
\]

\[
\mathbf{A}_2 = \mathbf{R}_2 \mathbf{Q}_2 = \begin{bmatrix}
4.95 & 2.14 \\
0.14 & -0.95 \\
\end{bmatrix}
\]

\[
\mathbf{A}_3 = \mathbf{R}_3 \mathbf{Q}_3 = \begin{bmatrix}
5.01 & -1.97 \\
0.03 & -1.01 \\
\end{bmatrix}
\]

**Remarks**

(i) \( \mathbf{A}_k \) and \( \mathbf{A}_{k+1} \) are unitarily similar

\[
\mathbf{A}_k = \mathbf{Q}_{k+1} \mathbf{R}_{k+1}, \quad \text{and} \quad \mathbf{A}_{k+1} = \mathbf{R}_{k+1} \mathbf{Q}_{k+1}
\]

\[
\implies \mathbf{R}_{k+1} = \mathbf{Q}_{k+1}^{*} \mathbf{A}_k \quad \text{and} \quad \mathbf{A}_{k+1} = \mathbf{R}_{k+1} \mathbf{Q}_{k+1}
\]

\[
\implies \mathbf{A}_{k+1} = \mathbf{Q}_{k+1}^{*} \mathbf{A}_k \mathbf{Q}_{k+1}
\]

(ii) QR factorization can be computed so that

\( \mathbf{A}_k \) is Hessenberg \( \implies \mathbf{A}_{k+1} \) is Hessenberg implying that the sequence \( \mathbf{A}_k \) is Hessenberg

\( \Box \)
THM (Invariance of Hessenberg Form)
Let \( A_k \in \mathbb{C}^{n \times n} \) be non-singular and Hessenberg. Then the matrix \( A_{k+1} \in \mathbb{C}^{n \times n} \) generated by the QR algorithm is also Hessenberg.

PROOF
First note that product of a Hessenberg matrix with an upper triangular matrix from left or right is Hessenberg.

Now
\[
A_k = Q_{k+1} R_{k+1} \implies Q_{k+1} = A_k R_{k+1}^{-1}
\]
\[
\implies Q_{k+1} \text{ is Hessenberg}
\]

Furthermore
\[
A_{k+1} = R_{k+1} Q_{k+1} \implies A_{k+1} \text{ is Hessenberg}
\]

(iii) The QR factorization
\[
A_k = Q_k R_k
\]
can be computed at a cost of \( O(n^2) \) (by HH reflectors, since \( A_k \) is Hessenberg) and the multiplication
\[
A_{k+1} = R_{k+1} Q_{k+1}
\]
can be performed at a cost of \( O(n^2) \). (since \( Q_k \) is formed by HH reflectors)
Therefore the cost of each QR iteration

\[ A_k \rightarrow A_{k+1} = Q_{k+1}^* A_k Q_{k+1} \]

is \( O(n^2) \).