AMS526: Numerical Analysis I
(Numerical Linear Algebra)
Lecture 17: QR Algorithm without Shifts (NLA§28); QR Algorithm with Shifts (NLA§29)

Xiangmin Jiao

SUNY Stony Brook
Outline

1. QR Algorithm without Shifts

2. QR Algorithm with Shifts
QR Algorithm

- Most basic version of QR algorithm is remarkably simple:

<table>
<thead>
<tr>
<th>Algorithm: “Pure” QR Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^{(0)} = A$</td>
</tr>
<tr>
<td>$\text{for } k = 1, 2, \ldots$</td>
</tr>
<tr>
<td>$Q^{(k)} R^{(k)} = A^{(k-1)}$</td>
</tr>
<tr>
<td>$A^{(k)} = R^{(k)} Q^{(k)}$</td>
</tr>
</tbody>
</table>

- With some suitable assumptions, $A^{(k)}$ converge to Schur form of $A$ (diagonal if $A$ is symmetric)
- Similarity transformation of $A$:

  $$A^{(k)} = R^{(k)} Q^{(k)} = \left( Q^{(k)} \right)^T A^{(k-1)} Q^{(k)}$$

- But why it works?
Unnormalized Simultaneous Iteration

- To understand QR algorithm, first consider simple algorithm
- Simultaneous iteration is power iteration applied to several vectors
- Start with linearly independent $v_1^{(0)}, \ldots, v_n^{(0)}$
- We know from power iteration that $A^k v_1^{(0)}$ converge to $q_1$
- With some assumptions, the space $\langle A^k v_1^{(0)}, \ldots, A^k v_n^{(0)} \rangle$ should converge to $\langle q_1, \ldots, q_n \rangle$
- Notation: Define initial matrix $V^{(0)}$ and matrix $V^{(k)}$ at step $k$:

$$V^{(0)} = \begin{bmatrix} v_1^{(0)} & \cdots & v_n^{(0)} \end{bmatrix}, \quad V^{(k)} = A^k V^{(0)} = \begin{bmatrix} v_1^{(k)} & \cdots & v_n^{(k)} \end{bmatrix}$$
Unnormalized Simultaneous Iteration

- Define orthogonal basis for column space of $V^{(k)}$ by reduced QR factorization $\hat{Q}^{(k)} \hat{R}^{(k)} = V^{(k)}$

- We assume that
  1. leading $n + 1$ eigenvalues are distinct, and
  2. all leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular where $\hat{Q} = [q_1 | \cdots | q_n]$

- We then have columns of $\hat{Q}^{(k)}$ converge to eigenvectors of $A$: 
  $$\|q_j^{(k)} - (\pm q_j)\| = O(c^k),$$

  where $c = \max_{1 \leq k \leq n} |\lambda_{k+1}|/|\lambda_k|$

- Proof idea: Show that subspace of any leading $j$ columns of $V^{(k)} = A^k V^{(0)}$ converges to subspace of first $j$ eigenvectors of $A$, so does the subspace of any leading $j$ columns of $\hat{Q}^{(k)}$. 
Simultaneous Iteration

- Matrices $V^{(k)} = A^k V^{(0)}$ are highly ill-conditioned
- Orthonormalize at each step rather than at the end

Algorithm: Simultaneous Iteration

1. Pick $\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$
2. For $k = 1, 2, \ldots$
   - $Z = A\hat{Q}^{(k-1)}$
   - $\hat{Q}^{(k)} \hat{R}^{(k)} = Z$

- Column spaces of $\hat{Q}^{(k)}$ and $Z^{(k)}$ are both equal to column space of $A^k \hat{Q}^{(0)}$, therefore same convergence as before
Simultaneous Iteration $\iff$ QR Algorithm

**Algorithm: Simultaneous Iteration**

Pick $\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$

for $k = 1, 2, \ldots$

\[
Z = A\hat{Q}^{(k-1)} \\
\hat{Q}^{(k)}\hat{R}^{(k)} = Z
\]

**Algorithm: “Pure” QR Algorithm**

\[
A^{(0)} = A \\
A^{(k)} = R^{(k)}Q^{(k)}
\]

- QR algorithm is equivalent to simultaneous iteration with $\hat{Q}^{(0)} = I$
- Replace $\hat{R}^{(k)}$ by $R^{(k)}$ and $\hat{Q}^{(k)}$ by $Q^{(k)}$, and introduce new statement

\[
A^{(k)} = \left(Q^{(k)}\right)^T A Q^{(k)}
\]

Simultaneous iteration

\[
Q^{(0)} = I \\
Z = A Q^{(k-1)} \\
Q^{(k)} R^{(k)} = Z \\
A^{(k)} = \left(Q^{(k)}\right)^T A Q^{(k)}
\]

QR algorithm

\[
A^{(0)} = A \\
Q^{(k)} R^{(k)} = A^{(k-1)} \\
A^{(k)} = R^{(k)}Q^{(k)} \\
Q^{(k)} = Q^{(1)} Q^{(2)} \ldots Q^{(k)}
\]
Simultaneous Iteration $\iff$ QR Algorithm

- $Q^{(k)} = Q^{(1)} Q^{(2)} \ldots Q^{(k)}$. Let $R^{(k)} = R^{(k)} R^{(k-1)} \ldots R^{(1)}$
- Both schemes generate QR factorization $A^k = Q^{(k)} R^{(k)}$ and projection $A^{(k)} = \left( Q^{(k)} \right)^T A Q^{(k)}$

Proof by induction. For $k = 0$ it is trivial for both algorithms.
For $k \geq 1$ with simultaneous iteration, $A^{(k)}$ is given by definition, and

$$A^k = A Q^{(k-1)} R^{(k-1)} = Q^{(k)} R^{(k)} R^{(k-1)} = Q^{(k)} R^{(k)}$$

For $k \geq 1$ with QR algorithm,

$$A^k = A Q^{(k-1)} R^{(k-1)} = Q^{(k-1)} A^{(k-1)} R^{(k-1)} = Q^{(k)} R^{(k)}$$

and

$$A^{(k)} = \left( Q^{(k)} \right)^T A^{(k-1)} Q^{(k)} = \left( Q^{(k)} \right)^T A Q^{(k)}$$
Convergence of QR Algorithm

- Since $Q^{(k)} = \hat{Q}^{(k)}$ in simultaneous iteration, column vectors of $Q^{(k)}$ converge linearly to eigenvectors if $A$ has distinct eigenvalues.

- From $A^{(k)} = \left(Q^{(k)}\right)^T AQ^{(k)}$, diagonal entries of $A^{(k)}$ are Rayleigh quotients of column vectors of $Q^{(k)}$, so they converge linearly to eigenvalues of $A$.

- Off-diagonal entries of $A^{(k)}$ converge to zeros, as they are generalized Rayleigh quotients involving approximations of distinct eigenvectors.

- Overall, $A = Q^{(k)} A^{(k)} \left(Q^{(k)}\right)^T$. For a symmetric matrix, it converges to eigenvalue decomposition of $A$.

- Convergence rate is only linear: columns of $\hat{Q}^{(k)}$ converge to eigenvectors of $A$ 
  $\|q_j^{(k)} - (\pm q_j)\| = O(c^k)$, where 
  $c = \max_{1 \leq k \leq n} |\lambda_{k+1}|/|\lambda_k|$.
Outline

1. QR Algorithm without Shifts
2. QR Algorithm with Shifts
Simultaneous Inverse Iteration ⇔ QR Algorithm

- Similar to inverse iteration, QR algorithm can be sped-up by introducing shifts at each step.
- Assume $A$ is real and symmetric. QR algorithm is equivalent to *simultaneous inverse iteration*, applied to “flipped” identity matrix $P$.

\[
P = \begin{bmatrix}
1 & & & \\
1 & 1 & & \\
& \ddots & \ddots & \\
& & 1 & 1
\end{bmatrix}
\]

Simultaneous inverse iteration

\[
\hat{Q}^{(0)} = P
\]

\[\text{for } k = 1, 2, \ldots\]

\[
Z = A^{-1} \hat{Q}^{(k-1)}
\]

\[
\hat{Q}^{(k)} \hat{R}^{(k)} = Z
\]

“Pure” QR Algorithm

\[
A^{(0)} = A
\]

\[\text{for } k = 1, 2, \ldots\]

\[
Q^{(k)} R^{(k)} = A^{(k-1)}
\]

\[
A^{(k)} = R^{(k)} Q^{(k)}
\]
Simultaneous Inverse Iteration $\iff$ QR Algorithm

Let $\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \ldots Q^{(k)}$ and $\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \ldots R^{(1)}$. Then $A^k = \underline{Q}^{(k)} \underline{R}^{(k)}$

Inverting $A^k$, we have $A^{-k} = \left(\underline{R}^{(k)}\right)^{-1} \left(\underline{Q}^{(k)}\right)^T$

Because $A^{-k}$ is symmetric, $A^{-k} = \underline{Q}^{(k)} \left(\underline{R}^{(k)}\right)^{-T}$

Use “flipped” permutation matrix $P$ and write that last expression as

$$A^{-k} P = \begin{bmatrix} \underline{Q}^{(k)} P \end{bmatrix} \begin{bmatrix} P \left(\underline{R}^{(k)}\right)^{-T} P \end{bmatrix},$$

which is QR factorization of $A^{-k} P$

Therefore, simultaneous inverse iteration applied to $\hat{Q}^{(0)} = P$ is “equivalent” to QR algorithm, in that it produces

$$\hat{Q}^{(k)} = \underline{Q}^{(k)} P \text{ and } \hat{R}^{(k)} \hat{R}^{(k-1)} \ldots \hat{R}^{(1)} = P \left(\underline{R}^{(k)}\right)^{-T} P$$

Question: How to obtain $A^{(k)}$ in simultaneous inverse iteration?
Simultaneous Inverse Iteration $\iff$ QR Algorithm

- Let $Q^{(k)} = Q^{(1)} Q^{(2)} \cdots Q^{(k)}$ and $R^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)}$. Then $A^k = Q^{(k)} R^{(k)}$.

- Inverting $A^k$, we have $A^{-k} = \left( R^{(k)} \right)^{-1} \left( Q^{(k)} \right)^T$.

- Because $A^{-k}$ is symmetric, $A^{-k} = Q^{(k)} \left( R^{(k)} \right)^{-T}$.

- Use “flipped” permutation matrix $P$ and write that last expression as

\[
A^{-k} P = \begin{bmatrix} Q^{(k)} P \end{bmatrix} \begin{bmatrix} P \left( R^{(k)} \right)^{-T} P \end{bmatrix},
\]

which is QR factorization of $A^{-k} P$.

- Therefore, simultaneous inverse iteration applied to $\hat{Q}^{(0)} = P$ is “equivalent” to QR algorithm, in that it produces

\[
\hat{Q}^{(k)} = Q^{(k)} P \text{ and } \hat{R}^{(k)} \hat{R}^{(k-1)} \cdots \hat{R}^{(1)} = P \left( R^{(k)} \right)^{-T} P
\]

- Question: How to obtain $A^{(k)}$ in simultaneous inverse iteration?

Answer: $A^{(k)} = \left( Q^{(k)} \right)^T A Q^{(k)} = P \left( \hat{Q}^{(k)} \right)^T A \hat{Q}^{(k)} P$.
QR Algorithm with Shifts

- Similar to inverse iteration, we can introduce shifts $\mu^{(k)}$ to accelerate convergence

Algorithm: QR Algorithm with Shifts

\[ A^{(0)} = A \]

for $k = 1, 2, \ldots$

Pick a shift $\mu^{(k)}$

\[ Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I \]

\[ A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I \]
Properties of QR Algorithm with Shift

- From $Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$ and $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$, we have
  
  $$A^{(k)} = \left( Q^{(k)} \right)^T A^{(k-1)} Q^{(k)}$$

- Then by induction, $A^{(k)} = \left( Q^{(k)} \right)^T A Q^{(k)}$

- However, instead of $A^k = Q^{(k)} R^{(k)}$, we now have
  
  $$\left( A - \mu^{(k)} I \right) \left( A - \mu^{(k-1)} I \right) \cdots \left( A - \mu^{(1)} I \right) = Q^{(k)} R^{(k)}$$

  which can be shown by induction

- In other words, $Q^{(k)}$ is orthogonalization of $\prod_{j=k}^{1} (A - \mu^{(j)} I)$

- If $\mu^{(k)}$ are good estimates of eigenvalues, then last column of $Q^{(k)}$ converges to corresponding eigenvector
Choosing $\mu^{(k)}$: Rayleigh Quotient Shift

- Natural choice of $\mu^{(k)}$ is Rayleigh quotient for last column of $Q^{(k)}$

$$\mu^{(k)} = r(q^{(k)}_m) = (q^{(k)}_m)^T A q^{(k)}_m$$

- As in Rayleigh quotient iteration, last column $q^{(k)}_m$ converges cubically
- This Rayleigh quotient appears as $(m, m)$ entry of $A^{(k)}$ since

$$A^{(k)} = \left(\begin{array}{c} Q^{(k)} \end{array}\right)^T A Q^{(k)}$$

- Rayleigh quotient shift corresponds to setting $\mu^{(k)} = A^{(k)}_{mm}$
Choosing $\mu^{(k)}$: Wilkinson Shift

- QR algorithm with Rayleigh quotient shift might fail sometimes, e.g., $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, for which $A^{(k)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\mu$ is always 0.
- Wilkinson breaks symmetry by considering lower-rightmost $2 \times 2$ submatrix of $A^{(k)}$: $B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}$
- Choose eigenvalue of $B$ closer to $a_m$, with arbitrary tie-breaking:
  \[ \mu = a_m - \text{sign}(\delta)b_{m-1}^2/\left(|\delta| + \sqrt{\delta^2 + b_{m-1}^2}\right) \]
  where $\delta = (a_{m-1} - a_m)/2$; if $\delta = 0$, set \text{sign}(\delta) to 1 (or -1).
- QR algorithm always converges with this shift; quadratically in worst case, and cubically in general.
“Practical” QR Algorithm

- Practical QR algorithm involves two additional components:
  - tridiagonalization of $A$ at the beginning. The tridiagonal structure is preserved by $A^{(k)}$ (Exercise 28.2)
  - deflation of $A$ into submatrices when $A^{(k)}$ is separable

Algorithm: “Practical” QR Algorithm

\[
(Q^{(0)})^T A^{(0)} Q^{(0)} = A \{\text{tridiagonalization of } A\}
\]

for $k = 1, 2, \ldots$

Pick a shift $\mu^{(k)}$

\[
Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I
\]

\[
A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I
\]

If any off-diagonal element $a^{(k)}_{j,j+1}$ is sufficiently close to zero

set $a_{j,j+1} = a_{j+1,j} = 0$ to obtain

\[
\begin{bmatrix}
A_1 & A_2 \\
A_2 & A_1
\end{bmatrix} = A^{(k)}
\]

and apply QR algorithm to $A_1$ and $A_2$
Stability and Accuracy

Theorem

QR algorithm is backward stable

$$\tilde{Q}\tilde{\Lambda}\tilde{Q} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

where $\tilde{\Lambda}$ is computed $\Lambda$ and $\tilde{Q}$ is exactly orthogonal matrix

- Its combination with Hessenberg reduction is also backward stable
- Furthermore, eigenvalues are always well conditioned for normal matrices: it can be show that $|\tilde{\lambda}_j - \lambda_j| \leq \|\delta A\|_2$, and therefore,

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{\|A\|} = O(\epsilon_{\text{machine}})$$

where $\tilde{\lambda}_j$ are the computed eigenvalues
- However, sensitivity of eigenvectors depends on distances between adjacent eigenvalues, so error in eigenvectors may be arbitrarily large