AMS526: Numerical Analysis I
(Numerical Linear Algebra)
Lecture 17: QR Algorithm with Shifts

Xiangmin Jiao

SUNY Stony Brook
Outline

1. QR Algorithm with Shifts

2. Review for Midterm #2
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 $P$.

$$P = \begin{bmatrix}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
\vdots & \ddots & \ddots & \vdots \\
1 & \cdots & 1 & 1
\end{bmatrix}$$

Simultaneous inverse iteration

$$\hat{Q}^{(0)} = P$$

(for $k = 1, 2, \ldots$)

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

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

“Pure” QR Algorithm

$$A^{(0)} = A$$

(for $k = 1, 2, \ldots$)

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

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

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

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

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

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

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

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

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

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

- 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 $Q^{(k)}$

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

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

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

- Rayleigh quotient shift corresponds to setting $\mu^{(k)} = A_{mm}^{(k)}$
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

$$\begin{align*}
(Q^{(0)})^T A^{(0)} Q^{(0)} &= A \ \{\text{tridiagonolization of } A\} \\
\text{for } k &= 1, 2, \ldots \\
\text{Pick a shift } \mu^{(k)} \\
Q^{(k)} R^{(k)} &= A^{(k-1)} - \mu^{(k)} I \\
A^{(k)} &= R^{(k)} Q^{(k)} + \mu^{(k)} I \\
\text{If any off-diagonal element } a_{j,j+1}^{(k)} \text{ is sufficiently close to zero} \\
\text{set } a_{j,j+1} = a_{j+1,j} = 0 \text{ to obtain} \\
\begin{bmatrix}
A_1 \\
A_2
\end{bmatrix} &= A^{(k)} \text{ and apply QR algorithm to } A_1 \text{ and } A_2
\end{align*}$$
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
Outline

1. QR Algorithm with Shifts

2. Review for Midterm #2
Algorithms

- **Cholesky Factorization**
  - Symmetric positive definite matrices
  - Cholesky factorization vs. LU factorization
  - $LDL^T$ factorization; $LDL^T$ with pivoting

- **QR factorization**
  - Classical and modified Gram-Schmidt
  - QR factorization using Householder triangularization

- **Solutions of linear least squares**
  - Solution using Householder QR and other QR factorization
  - Alternative solutions: normal equation; SVD
Eigenvalue Problem

- **Eigenvalue problem** of $m \times m$ matrix $A$ is $Ax = \lambda x$
- **Characteristic polynomial** is $\det(A - \lambda I)$
- **Eigenvalue decomposition** of $A$ is $A = X\Lambda X^{-1}$ (does not always exist)
- **Geometric multiplicity** of $\lambda$ is $\dim(\text{null}(A - \lambda I))$, and **algebraic multiplicity** of $\lambda$ is its multiplicity as a root of $p_A$, where algebraic multiplicity $\geq$ geometric multiplicity
- Similar matrices have the same eigenvalues, and algebraic and geometric multiplicities
- **Schur factorization** $A = QTQ^*$ uses unitary similarity transformations
Eigenvalue Algorithms

- Underlying concepts: power iterations, Rayleigh quotient, inverse iterations, convergence rate
- Schur factorization is typically done in two steps
  - Reduction to Hessenberg form for non-Hermitian matrices or reduction to tridiagonal form for hermitian matrices by unitary similarity transformation
  - Finding eigenvalues of Hessenberg or tridiagonal form
- Finding eigenvalue of tridiagonal forms
  - QR algorithm with shifts, and their interpretations as (inverse) simultaneous iterations