QR-method lecture 2

SF2524 - Matrix Computations for Large-scale Systems

Agenda QR-method

1. Decompositions (previous lecture)
   - Jordan form
   - Schur decomposition
   - QR-factorization
2. Basic QR-method
3. Improvement 1: Two-phase approach
   - Hessenberg reduction (previous lecture)
   - **Hessenberg QR-method**
4. Improvement 2: Acceleration with shifts
5. Convergence theory

Agenda QR-method

1. Decompositions (previous lecture)
   - Jordan form
   - Schur decomposition
   - QR-factorization
2. Basic QR-method
3. Improvement 1: Two-phase approach
   - Hessenberg reduction (previous lecture)
   - **Hessenberg QR-method**
4. Improvement 2: Acceleration with shifts
5. Convergence theory

Reading instructions

Point 1: TB Lecture 24

Points 2-4: Lecture notes "QR-method" on course web page

Point 5: TB Chapter 28

(Extra reading: TB Chapter 25-26, 28-29)

Basic QR-method (previous lecture)

Basic QR-method = basic QR-algorithm

Simple basic idea: Let $A_0 = A$ and iterate:

- Compute QR-factorization of $A_k = QR$
- Set $A_{k+1} = RQ$.

Basic QR-method (previous lecture)

Basic QR-method = basic QR-algorithm

Simple basic idea: Let $A_0 = A$ and iterate:

- Compute QR-factorization of $A_k = QR$
- Set $A_{k+1} = RQ$.

Disadvantages

- Computing a QR-factorization is quite expensive. One iteration of the basic QR-method

$$\mathcal{O}(m^3).$$

Basic QR-method (previous lecture)

Basic QR-method = basic QR-algorithm

Simple basic idea: Let $A_0 = A$ and iterate:

- Compute QR-factorization of $A_k = QR$
- Set $A_{k+1} = RQ$.

Disadvantages

- Computing a QR-factorization is quite expensive. One iteration of the basic QR-method

$$\mathcal{O}(m^3).$$

- The method often requires many iterations.

Improvement 1: Two-phase approach

We will separate the computation into two phases:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \underset{\text{Phase 1}}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \underset{\text{Phase 2}}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

Improvement 1: Two-phase approach

We will separate the computation into two phases:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \underset{\text{Phase 1}}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \underset{\text{Phase 2}}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

Phases

- Phase 1: Reduce the matrix to a Hessenberg with similarity transformations (Section 2.2.1 in lecture notes)

Improvement 1: Two-phase approach

We will separate the computation into two phases:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \underset{\text{Phase 1}}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \underset{\text{Phase 2}}{\rightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

Phases

- Phase 1: Reduce the matrix to a Hessenberg with similarity transformations (Section 2.2.1 in lecture notes)
- Phase 2: Specialize the QR-method to Hessenberg matrices (Section 2.2.2 in lecture notes)

Phase 1: Hessenberg reduction (previous lecture)

Phase 1: Hessenberg reduction

Compute unitary P and Hessenberg matrix H such that

$$A = PHP^*$$

Phase 1: Hessenberg reduction (previous lecture)

Phase 1: Hessenberg reduction

Compute unitary P and Hessenberg matrix H such that

$$A = PHP^*$$

Unlike the Schur factorization, this can be computed with a finite number of operations.

Phase 1: Hessenberg reduction (previous lecture)

Phase 1: Hessenberg reduction

Compute unitary P and Hessenberg matrix H such that

$$A = PHP^*$$

Unlike the Schur factorization, this can be computed with a finite number of operations.

Idea

- Householder reflector:

$$P = I - 2uu^* \quad \text{where } u \in \mathbb{C}^m \text{ and } \|u\| = 1,$$

- Apply one Householder reflector at a time to eliminate the column by column.

Phase 2: Hessenberg QR-method

A QR-step on a Hessenberg matrix is a Hessenberg matrix:

* Matlab demo showing QR-step: `hessenberg_is_hessenberg.m` *

Phase 2: Hessenberg QR-method

A QR-step on a Hessenberg matrix is a Hessenberg matrix:

* Matlab demo showing QR-step: `hessenberg_is_hessenberg.m` *

Theorem (Theorem 2.2.4)

If the basic QR-method is applied to a Hessenberg matrix, then all iterates A_k are Hessenberg matrices.

Phase 2: Hessenberg QR-method

A QR-step on a Hessenberg matrix is a Hessenberg matrix:

* Matlab demo showing QR-step: `hessenberg_is_hessenberg.m` *

Theorem (Theorem 2.2.4)

If the basic QR-method is applied to a Hessenberg matrix, then all iterates A_k are Hessenberg matrices.

Recall: basic QR-step is $\mathcal{O}(m^3)$.

Phase 2: Hessenberg QR-method

A QR-step on a Hessenberg matrix is a Hessenberg matrix:

* Matlab demo showing QR-step: `hessenberg_is_hessenberg.m` *

Theorem (Theorem 2.2.4)

If the basic QR-method is applied to a Hessenberg matrix, then all iterates A_k are Hessenberg matrices.

Recall: basic QR-step is $\mathcal{O}(m^3)$.

Hessenberg structure can be exploited such that we can carry out a QR-step with less operations.

Definition (Givens rotation)

The matrix $G(i,j,c,s) \in \mathbb{R}^{n\times n}$ corresponding to a Givens rotation is defined by

$$G(i,j,c,s) := \begin{bmatrix} I & & & & \\ & c & & -s & \\ & & I & & \\ & s & & c & \\ & & & & I \end{bmatrix},$$

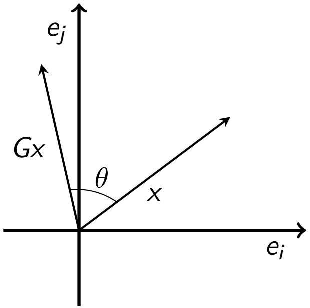
which deviates from identity at row and column i and j.

Definition (Givens rotation)

The matrix $G(i, j, c, s) \in \mathbb{R}^{n \times n}$ corresponding to a Givens rotation is defined by

$$G(i, j, c, s) := \begin{bmatrix} I & & & & \\ & c & & -s & \\ & & I & & \\ & s & & c & \\ & & & & I \end{bmatrix},$$

which deviates from identity at row and column i and j.

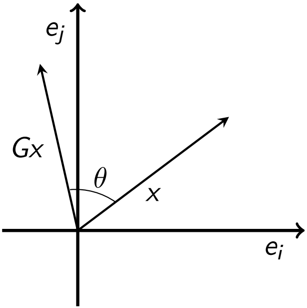


Definition (Givens rotation)

The matrix $G(i, j, c, s) \in \mathbb{R}^{n \times n}$ corresponding to a Givens rotation is defined by

$$G(i,j,c,s) := \begin{bmatrix} I & & & & \\ & c & & -s & \\ & & I & & \\ & s & & c & \\ & & & & I \end{bmatrix},$$

which deviates from identity at row and column i and j.



Properties

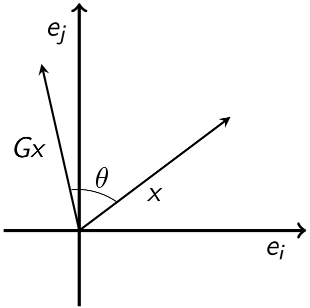
- $G^* = G^{-1}$

Definition (Givens rotation)

The matrix $G(i, j, c, s) \in \mathbb{R}^{n\times n}$ corresponding to a Givens rotation is defined by

$$G(i, j, c, s) := \begin{bmatrix} I & & & & \\ & c & & -s & \\ & & I & & \\ & s & & c & \\ & & & & I \end{bmatrix},$$

which deviates from identity at row and column i and j.



Properties

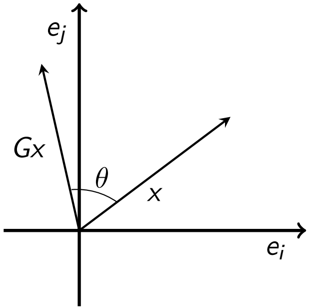
- $G^* = G^{-1}$
- Gz can be computed with $O(1)$ operations

Definition (Givens rotation)

The matrix $G(i,j,c,s) \in \mathbb{R}^{n\times n}$ corresponding to a Givens rotation is defined by

$$G(i,j,c,s) := \begin{bmatrix} I & & & & \\ & c & & -s & \\ & & I & & \\ & s & & c & \\ & & & & I \end{bmatrix},$$

which deviates from identity at row and column i and j.



Properties

- $G^* = G^{-1}$
- Gz can be computed with $O(1)$ operations
- $\cdots$

The Q-matrix in the QR-factorization of a Hessenberg matrix can be factorized as a product of $m-1$ Givens rotators.

The Q-matrix in the QR-factorization of a Hessenberg matrix can be factorized as a product of $m-1$ Givens rotators.

Theorem (Theorem 2.2.6)

Suppose $A \in \mathbb{C}^{m\times m}$ is a Hessenberg matrix. Let H_i be generated as follows $H_1 = A$

$$H_{i+1} = G_i^T H_i, \quad i = 1, \dots, m-1$$

where $G_i = G(i, i+1, (H_i)_{i,i}/r_i, (H_i)_{i+1,i}/r_i)$ and $r_i = \sqrt{(H_i)_{i,i}^2 + (H_i)_{i+1,i}^2}$ and we assume $r_i \neq 0$. Then, H_n is upper triangular and

$$A = (G_1 G_2 \cdots G_{m-1}) H_n = QR$$

is a QR-factorization of A.

The Q-matrix in the QR-factorization of a Hessenberg matrix can be factorized as a product of $m-1$ Givens rotators.

Theorem (Theorem 2.2.6)

Suppose $A\in\mathbb{C}^{m\times m}$ is a Hessenberg matrix. Let H_i be generated as follows $H_1=A$

$$H_{i+1}=G_i^T H_i,\;\; i=1,\ldots,m-1$$

where $G_i=G(i,i+1,(H_i)_{i,i}/r_i,(H_i)_{i+1,i}/r_i)$ and $r_i=\sqrt{(H_i)_{i,i}^2+(H_i)_{i+1,i}^2}$ and we assume $r_i\neq 0$. Then, H_n is upper triangular and

$$A=(G_1G_2\cdots G_{m-1})H_n=QR$$

is a QR-factorization of A.

Proof idea: Only one rotator required to bring one column of a Hessenberg matrix to a triangular.

* Matlab: Explicit QR-factorization of Hessenberg $\mathtt{qr}_g ivens.m$ *

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators: Let

$$A_{k-1} = (G_1 G_2 \cdots G_{m-1}) R_m$$

and

$$A_k = R_m (G_1 G_2 \cdots G_{m-1}) =$$

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators: Let

$$A_{k-1} = (G_1 G_2 \cdots G_{m-1}) R_m$$

and

$$A_k = R_m (G_1 G_2 \cdots G_{m-1}) = (\cdots((R_m G_1) G_2) \cdots) G_m$$

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators: Let

$$A_{k-1} = (G_1 G_2 \cdots G_{m-1}) R_m$$

and

$$A_k = R_m (G_1 G_2 \cdots G_{m-1}) = (\cdots((R_m G_1) G_2) \cdots) G_m$$

Complexity of one QR-step for a Hessenberg matrix

We need to apply $2(m-1)$ givens rotators to compute one QR-step.

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators: Let

$$A_{k-1} = (G_1 G_2 \cdots G_{m-1}) R_m$$

and

$$A_k = R_m (G_1 G_2 \cdots G_{m-1}) = (\cdots((R_m G_1) G_2) \cdots) G_m$$

Complexity of one QR-step for a Hessenberg matrix

We need to apply $2(m-1)$ givens rotators to compute one QR-step.

- One givens rotator applied to a vector can be computed in $O(1)$ operations.

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators: Let

$$A_{k-1} = (G_1 G_2 \cdots G_{m-1}) R_m$$

and

$$A_k = R_m (G_1 G_2 \cdots G_{m-1}) = (\cdots((R_m G_1) G_2) \cdots) G_m$$

Complexity of one QR-step for a Hessenberg matrix

We need to apply $2(m-1)$ givens rotators to compute one QR-step.

- One givens rotator applied to a vector can be computed in $O(1)$ operations.
- One givens rotator applied to matrix can be computed in $O(m)$ operations.

Idea of Hessenberg QR-method: Do not explicitly compute the Q-matrix but only implicitly apply the Givens rotators: Let

$$A_{k-1} = (G_1 G_2 \cdots G_{m-1}) R_m$$

and

$$A_k = R_m (G_1 G_2 \cdots G_{m-1}) = (\cdots((R_m G_1) G_2) \cdots) G_m$$

Complexity of one QR-step for a Hessenberg matrix

We need to apply $2(m-1)$ givens rotators to compute one QR-step.

- One givens rotator applied to a vector can be computed in $O(1)$ operations.
- One givens rotator applied to matrix can be computed in $O(m)$ operations.

$\Rightarrow$

$$\text{the complexity of one Hessenberg QR step} = \mathcal{O}(m^2)$$

Givens rotators only modify very few elements.
Several optimizations possible. $\Rightarrow$

Algorithm 3 Hessenberg QR algorithm

Input: A Hessenberg matrix $A \in \mathbb{C}^{n\times n}$

Output: Upper triangular T such that $A = UTU^*$ for an orthogonal matrix U.

Set $A_0 := A$
for $k = 1,\ldots$ **do**
  // One Hessenberg QR step
  $H = A_{k-1}$
  for $i = 1,\ldots,n-1$ **do**
    $[c_i, s_i] = \text{givens}(h_{i,i}, h_{i+1,i})$
    $H_{i:i+1,i:n} = \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} H_{i:i+1,i:n}$
  end for
  for $i = 1,\ldots,n-1$ **do**
    $H_{1:i+1,i:i+1} = H_{1:i+1,i:i+1} \begin{bmatrix} c_i & -s_i \\ s_i & c_i \end{bmatrix}$
  end for
  $A_k = H$
end for
Return $T = A_\infty$

Show animation again:

http://www.youtube.com/watch?v=qmgxzsWWsNc

Show animation again:

http://www.youtube.com/watch?v=qmgxzsWWsNc

Acceleration still remains

Outline:

- Basic QR-method
- Improvement 1: Two-phase approach
  - Hessenberg reduction
  - Hessenberg QR-method
- **Improvement 2: Acceleration with shifts**
- Convergence theory

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method: Let $H_k = H$

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method: Let $H_k = H$

$$\begin{aligned} H - \mu I &= QR \\ \bar{H} &= RQ + \mu I \end{aligned}$$

and $H_{k+1} := \bar{H}$.

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method: Let $H_k = H$

$$\begin{aligned} H - \mu I &= QR \\ \bar{H} &= RQ + \mu I \end{aligned}$$

and $H_{k+1} := \bar{H}$.

Note:

$$H_{k+1} = \bar{H} = RQ + \mu I =$$

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method: Let $H_k = H$

$$\begin{aligned} H - \mu I &= QR \\ \bar{H} &= RQ + \mu I \end{aligned}$$

and $H_{k+1} := \bar{H}$.

Note:

$$H_{k+1} = \bar{H} = RQ + \mu I = Q^T(H - \mu I))Q + \mu I$$

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method: Let $H_k = H$

$$\begin{aligned} H - \mu I &= QR \\ \bar{H} &= RQ + \mu I \end{aligned}$$

and $H_{k+1} := \bar{H}$.

Note:

$$H_{k+1} = \bar{H} = RQ + \mu I = Q^T(H - \mu I))Q + \mu I = Q^T H_k Q$$

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method: Let $H_k = H$

$$\begin{aligned} H - \mu I &= QR \\ \bar{H} &= RQ + \mu I \end{aligned}$$

and $H_{k+1} := \bar{H}$.

Note:

$$H_{k+1} = \bar{H} = RQ + \mu I = Q^T(H - \mu I))Q + \mu I = Q^T H_k Q$$

$\Rightarrow$ One step of shifted QR-method is a similarity transformation, with a different Q matrix.

Idealized situation: Let $\mu = \lambda(H)$

Suppose μ is an eigenvalue:
$\Rightarrow$ $H - \mu I$ is a singular Hessenberg matrix.

Idealized situation: Let $\mu = \lambda(H)$

Suppose μ is an eigenvalue:
$\Rightarrow H - \mu I$ is a singular Hessenberg matrix.

QR-factorization of singular Hessenberg matrices (Lemma 2.3.1)

The R-matrix in the QR-decomposition of a singular unreduced Hessenberg matrix has the structure

$$R = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & 0 \end{bmatrix}.$$

Idealized situation: Let $\mu = \lambda(H)$

Suppose μ is an eigenvalue:
$\Rightarrow H - \mu I$ is a singular Hessenberg matrix.

QR-factorization of singular Hessenberg matrices (Lemma 2.3.1)

The R-matrix in the QR-decomposition of a singular unreduced Hessenberg matrix has the structure

$$R = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & 0 \end{bmatrix}.$$

* Matlab demo: Show QR-factorization of singular Hessenberg matrix in matlab *

Shifted QR for exact shift: $\mu = \lambda$

If $\mu = \lambda$ is an eigenvalue of H, then $H - \mu I$ is singular. Suppose Q, R a QR-factorization of a Hessenberg matrix and

$$R = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & 0 \end{bmatrix}.$$

Then,

* Prove on blackboard *

Shifted QR for exact shift: $\mu = \lambda$

If $\mu = \lambda$ is an eigenvalue of H, then $H - \mu I$ is singular. Suppose Q, R a QR-factorization of a Hessenberg matrix and

$$R = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & 0 \end{bmatrix}.$$

Then,

* Prove on blackboard *

$$RQ = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & & 0 \end{bmatrix}$$

and

Shifted QR for exact shift: $\mu = \lambda$

If $\mu = \lambda$ is an eigenvalue of H, then $H - \mu I$ is singular. Suppose Q, R a QR-factorization of a Hessenberg matrix and

$$R = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & 0 \end{bmatrix}.$$

Then,

* Prove on blackboard *

$$RQ = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & & 0 \end{bmatrix}$$

and

$$\bar{H} = RQ + \lambda I = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & & \lambda \end{bmatrix}.$$

Shifted QR for exact shift: $\mu = \lambda$

If $\mu = \lambda$ is an eigenvalue of H, then $H - \mu I$ is singular. Suppose Q, R a QR-factorization of a Hessenberg matrix and

$$R = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & 0 \end{bmatrix}.$$

Then, * Prove on blackboard *

$$RQ = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & & 0 \end{bmatrix}$$

and

$$\bar{H} = RQ + \lambda I = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & & \lambda \end{bmatrix}.$$

$\Rightarrow \lambda$ is an eigenvalue of $\bar{H}$.

More precisely:

Lemma (Lemma 2.3.2)

Suppose λ is an eigenvalue of the Hessenberg matrix H. Let $\bar{H}$ be the result of one shifted QR-step. Then,

$$\begin{aligned} \bar{h}_{n,n-1} &= 0 \\ \bar{h}_{n,n} &= \lambda. \end{aligned}$$

Select the shift

How to select the shifts?

- Shifted QR-method with $\mu = \lambda$ computes an eigenvalue in one step.

Select the shift

How to select the shifts?

- Shifted QR-method with $\mu = \lambda$ computes an eigenvalue in one step.
- The exact eigenvalue not available. How to select the shift?

Select the shift

How to select the shifts?

- Shifted QR-method with $\mu = \lambda$ computes an eigenvalue in one step.
- The exact eigenvalue not available. How to select the shift?

Rayleigh shifts

If we are close to convergence the diagonal element will be an approximate eigenvalue.

Select the shift

How to select the shifts?

- Shifted QR-method with $\mu = \lambda$ computes an eigenvalue in one step.
- The exact eigenvalue not available. How to select the shift?

Rayleigh shifts

If we are close to convergence the diagonal element will be an approximate eigenvalue. Rayleigh shifts:

$$\mu := r_{m,m}.$$

Select the shift

How to select the shifts?

- Shifted QR-method with $\mu = \lambda$ computes an eigenvalue in one step.
- The exact eigenvalue not available. How to select the shift?

Rayleigh shifts

If we are close to convergence the diagonal element will be an approximate eigenvalue. Rayleigh shifts:

$$\mu := r_{m,m}.$$

Explanation

- The QR-method can be interpreted as equivalent to variant of Power Method applied to A. (Will be shown later)

Select the shift

How to select the shifts?

- Shifted QR-method with $\mu = \lambda$ computes an eigenvalue in one step.
- The exact eigenvalue not available. How to select the shift?

Rayleigh shifts

If we are close to convergence the diagonal element will be an approximate eigenvalue. Rayleigh shifts:

$$\mu := r_{m,m}.$$

Explanation

- The QR-method can be interpreted as equivalent to variant of Power Method applied to A. (Will be shown later)
- The QR-method can be interpreted as equivalent to variant of Power Method applied to A^{-1}. (Proof sketched in TB Chapter 29) $\Rightarrow$ Rayleigh shifts can be interpreted as Rayleigh quotient iteration.

Deflation

QR-step on reduced Hessenberg matrix

Suppose

$$H = \begin{pmatrix} H_0 & H_1 \\ 0 & H_3 \end{pmatrix},$$

where H_3 is upper triangular and let

$$\bar{H} = \begin{pmatrix} \bar{H}_0 & \bar{H}_1 \\ \bar{H}_2 & \bar{H}_3 \end{pmatrix},$$

be the result of one (shifted) QR-step.

Deflation

QR-step on reduced Hessenberg matrix

Suppose

$$H = \begin{pmatrix} H_0 & H_1 \\ 0 & H_3 \end{pmatrix},$$

where H_3 is upper triangular and let

$$\bar{H} = \begin{pmatrix} \bar{H}_0 & \bar{H}_1 \\ \bar{H}_2 & \bar{H}_3 \end{pmatrix},$$

be the result of one (shifted) QR-step. Then, $\bar{H}_2 = 0$, $\bar{H}_3 = H_3$ and $\bar{H}_0$ is the result of one (shifted) QR-step applied to H_0.

Deflation

QR-step on reduced Hessenberg matrix

Suppose

$$H = \begin{pmatrix} H_0 & H_1 \\ 0 & H_3 \end{pmatrix},$$

where H_3 is upper triangular and let

$$\bar{H} = \begin{pmatrix} \bar{H}_0 & \bar{H}_1 \\ \bar{H}_2 & \bar{H}_3 \end{pmatrix},$$

be the result of one (shifted) QR-step. Then, $\bar{H}_2 = 0$, $\bar{H}_3 = H_3$ and $\bar{H}_0$ is the result of one (shifted) QR-step applied to H_0. * show proof *

Deflation

QR-step on reduced Hessenberg matrix

Suppose

$$H = \begin{pmatrix} H_0 & H_1 \\ 0 & H_3 \end{pmatrix},$$

where H_3 is upper triangular and let

$$\bar{H} = \begin{pmatrix} \bar{H}_0 & \bar{H}_1 \\ \bar{H}_2 & \bar{H}_3 \end{pmatrix},$$

be the result of one (shifted) QR-step. Then, $\bar{H}_2 = 0$, $\bar{H}_3 = H_3$ and $\bar{H}_0$ is the result of one (shifted) QR-step applied to H_0. * show proof *

$\Rightarrow$ We can reduce the active matrix when an eigenvalue is converged.

Deflation

QR-step on reduced Hessenberg matrix

Suppose

$$H = \begin{pmatrix} H_0 & H_1 \\ 0 & H_3 \end{pmatrix},$$

where H_3 is upper triangular and let

$$\bar{H} = \begin{pmatrix} \bar{H}_0 & \bar{H}_1 \\ \bar{H}_2 & \bar{H}_3 \end{pmatrix},$$

be the result of one (shifted) QR-step. Then, $\bar{H}_2 = 0$, $\bar{H}_3 = H_3$ and $\bar{H}_0$ is the result of one (shifted) QR-step applied to H_0. * show proof *

$\Rightarrow$ We can reduce the active matrix when an eigenvalue is converged.

This is called deflation.

Rayleigh shifts can be combined with deflation $\Rightarrow$

Algorithm 4 Hessenberg QR algorithm with Rayleigh quotient shift and deflation

Input: A Hessenberg matrix $A \in \mathbb{C}^{n\times n}$

Set $H^{(0)} := A$
for $m = n, \ldots, 2$ **do**
  $k = 0$
  repeat
    $k = k + 1$
    $\sigma_k = h_{m,m}^{(k-1)}$
    $H_{k-1} - \sigma_k I =: Q_k R_k$
    $H_k := R_k Q_k + \sigma_k I$
  until $|h_{m,m-1}^{(k)}|$ is sufficiently small
  Save $h_{m,m}^{(k)}$ as a converged eigenvalue
  Set $H^{(0)} = H_{1:(m-1),1:(m-1)}^{(k)} \in \mathbb{C}^{(m-1)\times(m-1)}$
end for

* show Hessenberg qr with shifts in matlab *
* `http://www.youtube.com/watch?v=qmgxzsWWsNc` *

Outline:

- Basic QR-method
- Improvement 1: Two-phase approach
  - Hessenberg reduction
  - Hessenberg QR-method
- Improvement 2: Acceleration with shifts
- **Convergence theory**

Convergence theory - TB Chapter 28

Didactic simplification for convergence of QR-method: Assume $A = A^T$.

Convergence theory - TB Chapter 28

Didactic simplification for convergence of QR-method: Assume $A = A^T$.

Convergence characterization

(1) Artificial algorithm: USI - Unnormalized Simultaneous Iteration

Convergence theory - TB Chapter 28

Didactic simplification for convergence of QR-method: Assume $A = A^T$.

Convergence characterization

(1) Artificial algorithm: USI - Unnormalized Simultaneous Iteration

(2) Show convergence properties of USI

Convergence theory - TB Chapter 28

Didactic simplification for convergence of QR-method: Assume $A = A^T$.

Convergence characterization

(1) Artificial algorithm: USI - Unnormalized Simultaneous Iteration

(2) Show convergence properties of USI

(3) Artificial algorithm: NSI - Normalized Simultaneous Iteration

Convergence theory - TB Chapter 28

Didactic simplification for convergence of QR-method: Assume $A = A^T$.

Convergence characterization

(1) Artificial algorithm: USI - Unnormalized Simultaneous Iteration
(2) Show convergence properties of USI
(3) Artificial algorithm: NSI - Normalized Simultaneous Iteration
(4) Show: USI $\Leftrightarrow$ NSI $\Leftrightarrow$ QR-method

Definition: Unnormalized simultaneous iteration (USI)

A generalization of power method with n vectors “simultaneously”

Definition: Unnormalized simultaneous iteration (USI)

A generalization of power method with n vectors “simultaneously”

$$V^{(0)} = [v_1^{(0)}, \ldots, v_n^{(0)}] \in \mathbb{R}^{m \times n}.$$

Define

$$V^{(k)} := A^k V^{(0)}.$$

Definition: Unnormalized simultaneous iteration (USI)

A generalization of power method with n vectors "simultaneously"

$$V^{(0)} = [v_1^{(0)}, \dots, v_n^{(0)}] \in \mathbb{R}^{m\times n}.$$

Define

$$V^{(k)} := A^k V^{(0)}.$$

A QR-factorization generalizes the normalization step:

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

Definition: Unnormalized simultaneous iteration (USI)

A generalization of power method with n vectors "simultaneously"

$$V^{(0)} = [v_1^{(0)}, \ldots, v_n^{(0)}] \in \mathbb{R}^{m \times n}.$$

Define

$$V^{(k)} := A^k V^{(0)}.$$

A QR-factorization generalizes the normalization step:

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

Convergence of USI

Convergence of USI

Assumptions:

- Let eigenvalues ordered and assume:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|.$$

Convergence of USI

Assumptions:

- Let eigenvalues ordered and assume:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|.$$

- Assume leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular, where $\hat{Q} = (q_1, \ldots, q_n)$ are the eigenvectors.

Convergence of USI

Assumptions:

- Let eigenvalues ordered and assume:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|.$$

- Assume leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular, where $\hat{Q} = (q_1, \ldots, q_n)$ are the eigenvectors.

Theorem (TB Theorem 28.1)

Suppose simultaneous iteration is started with $V^{(0)}$ and assumptions above are satisfied. Let q_j, $j = 1, \ldots, n$ be the first n eigenvectors of A. Then, as $k \to \infty$, the columns of the matrices $\hat{Q}^{(k)}$ convergence linearly to q_j

$$\|q_j^{(k)} - \pm q_j\| = \mathcal{O}(C^k), \;\; j = 1, \ldots, n,$$

Convergence of USI

Assumptions:

- Let eigenvalues ordered and assume:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|.$$

- Assume leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular, where $\hat{Q} = (q_1, \ldots, q_n)$ are the eigenvectors.

Theorem (TB Theorem 28.1)

Suppose simultaneous iteration is started with $V^{(0)}$ and assumptions above are satisfied. Let q_j, $j = 1, \ldots, n$ be the first n eigenvectors of A. Then, as $k \to \infty$, the columns of the matrices $\hat{Q}^{(k)}$ convergence linearly to q_j

$$\|q_j^{(k)} - \pm q_j\| = \mathcal{O}(C^k), \;\; j = 1, \ldots, n,$$

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

Convergence of USI

Assumptions:

- Let eigenvalues ordered and assume:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|.$$

- Assume leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular, where $\hat{Q} = (q_1, \ldots, q_n)$ are the eigenvectors.

Theorem (TB Theorem 28.1)

Suppose simultaneous iteration is started with $V^{(0)}$ and assumptions above are satisfied. Let q_j, $j = 1, \ldots, n$ be the first n eigenvectors of A. Then, as $k \to \infty$, the columns of the matrices $\hat{Q}^{(k)}$ convergence linearly to q_j

$$\|q_j^{(k)} - \pm q_j\| = \mathcal{O}(C^k), \;\; j = 1, \ldots, n,$$

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

* Show matlab demo on USI (video) *