QR-method - Lecture 2

SF2524 - Matrix Computations for Large-scale Systems

Outline QR-method:

1. Decompositions (last lecture)
2. Basic QR-method (last lecture)
3. Improvement 1: Two-phase approach
   - Hessenberg reduction
   - 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 (repetition last lecture)

- Method for dense eigenvalue problems
- Computes a Schur factorization

$$A = Q^* TQ$$

by using QR-factorizations

$$B = QR.$$

Basic QR-method (repetition last lecture)

- Method for dense eigenvalue problems
- Computes a Schur factorization

$$A = Q^* T Q$$

by using QR-factorizations

$$B = QR.$$

Basic QR-method

$$\begin{aligned} A_{k-1} &= Q_k R_k \\ A_k &:= R_k Q_k \end{aligned}$$

Basic QR-method (repetition last lecture)

- Method for dense eigenvalue problems
- Computes a Schur factorization

$$A = Q^* T Q$$

by using QR-factorizations

$$B = QR.$$

Basic QR-method

$$\begin{aligned} A_{k-1} &= Q_k R_k \\ A_k &:= R_k Q_k \end{aligned}$$

Generates a sequence of matrices A_k with same eigenvalues, and in general converge to a triangular matrix.

The basic QR-method is elegant and robust but not very efficient:

The basic QR-method is elegant and robust but not very efficient:

Disadvantages

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

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

The basic QR-method is elegant and robust but not very efficient:

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. (HW3, problem 1)

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

We will need matrices called Householder reflectors.

Definition

A matrix $P \in \mathbb{C}^{m\times m}$ of the form

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

is called a *Householder reflector*.

Phase 1: Hessenberg reduction

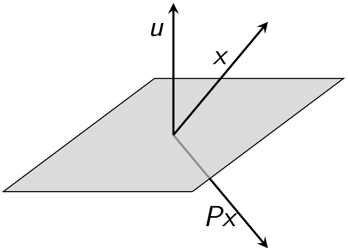
We will need matrices called Householder reflectors.

Definition

A matrix $P \in \mathbb{C}^{m\times m}$ of the form

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

is called a *Householder reflector*.



Phase 1: Hessenberg reduction

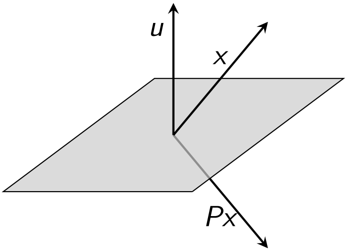
We will need matrices called Householder reflectors.

Definition

A matrix $P \in \mathbb{C}^{m \times m}$ of the form

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

is called a *Householder reflector*.



Properties

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

Phase 1: Hessenberg reduction

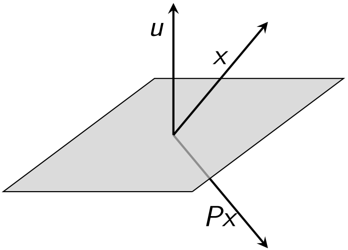
We will need matrices called Householder reflectors.

Definition

A matrix $P \in \mathbb{C}^{m\times m}$ of the form

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

is called a *Householder reflector*.



Properties

- $P^* = P^{-1} = P$
- $Pz = z - 2u(u^*z)$ can be computed with $O(m)$ operations.

Phase 1: Hessenberg reduction

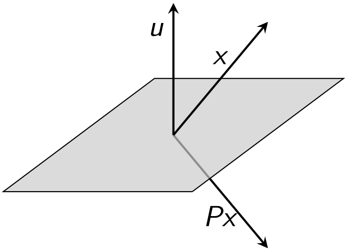
We will need matrices called Householder reflectors.

Definition

A matrix $P \in \mathbb{C}^{m\times m}$ of the form

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

is called a *Householder reflector*.



Properties

- $P^* = P^{-1} = P$
- $Pz = z - 2u(u^*z)$ can be computed with $O(m)$ operations.
- $\dots$

Householder reflectors satisfying $Px = \alpha e_1$

Problem

Given a vector x compute a Householder reflector such that

$$Px = \alpha e_1.$$

Householder reflectors satisfying $Px = \alpha e_1$

Problem

Given a vector x compute a Householder reflector such that

$$Px = \alpha e_1.$$

Solution (Lemma 2.2.3)

Let $\rho = \text{sign}(x_1)$,

$$z := x - \rho\|x\|e_1 = \begin{bmatrix} x_1 - \rho\|x\| \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

and

$$u = z/\|z\|.$$

Householder reflectors satisfying $Px = \alpha e_1$

Problem

Given a vector x compute a Householder reflector such that

$$Px = \alpha e_1.$$

Solution (Lemma 2.2.3)

Let $\rho = \text{sign}(x_1)$,

$$z := x - \rho\|x\|e_1 = \begin{bmatrix} x_1 - \rho\|x\| \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

and

$$u = z/\|z\|.$$

Then, $P = I - 2uu^*$ is a Householder reflector that satisfies $Px = \alpha e_1$.

Householder reflectors satisfying $Px = \alpha e_1$

Problem

Given a vector x compute a Householder reflector such that

$$Px = \alpha e_1.$$

Solution (Lemma 2.2.3)

Let $\rho = \text{sign}(x_1)$,

$$z := x - \rho\|x\|e_1 = \begin{bmatrix} x_1 - \rho\|x\| \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

and

$$u = z/\|z\|.$$

Then, $P = I - 2uu^*$ is a Householder reflector that satisfies $Px = \alpha e_1$.

* Matlab demo showing Householder reflectors *

We will be able to construct $m-2$ householder reflectors that bring the matrix to Hessenberg form.

Elimination for first column

$$P_1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & 0^T \\ 0 & I - 2u_1u_1^T \end{bmatrix}.$$

We will be able to construct $m-2$ householder reflectors that bring the matrix to Hessenberg form.

Elimination for first column

$$P_1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & 0^T \\ 0 & I - 2u_1u_1^T \end{bmatrix}.$$

Use Lemma 2.2.1 with $x^T = [a_{21}, \ldots, a_{n1}]$ to select u_1 such that

$$P_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

We will be able to construct $m-2$ householder reflectors that bring the matrix to Hessenberg form.

Elimination for first column

$$P_1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & 0^T \\ 0 & I - 2u_1u_1^T \end{bmatrix}.$$

Use Lemma 2.2.1 with $x^T = [a_{21}, \ldots, a_{n1}]$ to select u_1 such that

$$P_1A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

In order to have a similarity transformation mult from right:

$$P_1AP_1^{-1} = P_1AP_1 = \text{same structure as } P_1A.$$

We will be able to construct $m-2$ householder reflectors that bring the matrix to Hessenberg form.

Elimination for first column

$$P_1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & 0^T \\ 0 & I - 2u_1u_1^T \end{bmatrix}.$$

Use Lemma 2.2.1 with $x^T = [a_{21}, \dots, a_{n1}]$ to select u_1 such that

$$P_1A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}$$

In order to have a similarity transformation mult from right:

$$P_1AP_1^{-1} = P_1AP_1 = \text{same structure as } P_1A.$$

Elimination for second column

Repeat the process with:

$$P_2 = \begin{bmatrix} 1 & 0 & 0^T \\ 0 & 1 & 0^T \\ 0 & 0 & I - 2u_2 u_2^T \end{bmatrix}$$

Elimination for second column

Repeat the process with:

$$P_2 = \begin{bmatrix} 1 & 0 & 0^T \\ 0 & 1 & 0^T \\ 0 & 0 & I - 2u_2 u_2^T \end{bmatrix}$$

where u_2 is constructed from the $n-2$ last elements of the second column of $P_1 A P_1^*$.

$$P_1 A P_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \underset{\substack{\text{mult. from} \\ \text{left with } P_2}}{\longrightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

$$\underset{\substack{\text{mult. from} \\ \text{right with } P_2}}{\longrightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix} = P_2 P_1 A P_1 P_2$$

Elimination for second column

Repeat the process with:

$$P_2 = \begin{bmatrix} 1 & 0 & 0^T \\ 0 & 1 & 0^T \\ 0 & 0 & I - 2u_2u_2^T \end{bmatrix}$$

where u_2 is constructed from the $n-2$ last elements of the second column of $P_1AP_1^*$.

$$P_1AP_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \underset{\substack{\text{mult. from} \\ \text{left with } P_2}}{\longrightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}$$

$$\underset{\substack{\text{mult. from} \\ \text{right with } P_2}}{\longrightarrow} \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix} = P_2P_1AP_1P_2$$

* Matlab demo of the first two steps of the Hessenberg reduction *

The iteration can be implemented without explicit use of the P matrices.

Algorithm 2 Reduction to Hessenberg form

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

Output: A Hessenberg matrix H such that $H = U^*AU$.

for $k = 1, \ldots, n-2$ **do**

  Compute u_k using (2.4) where $x^T = [a_{k+1,k}, \cdots, a_{n,k}]$

  Compute P_kA: $A_{k+1:n,k:n} := A_{k+1:n,k:n} - 2u_k(u_k^* A_{k+1:n,k:n})$

  Compute $P_kAP_k^*$: $A_{1:n,k+1:n} := A_{1:n,k+1:n} - 2(A_{1:n,k+1:n}u_k)u_k^*$

end for

Let H be the Hessenberg part of A.

Phase 2: Hessenberg QR-method

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

* Matlab demo showing QR-step of Hessenberg is Hessenberg *

Phase 2: Hessenberg QR-method

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

* Matlab demo showing QR-step of Hessenberg is Hessenberg *

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 of Hessenberg is Hessenberg *

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 of Hessenberg is Hessenberg *

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

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

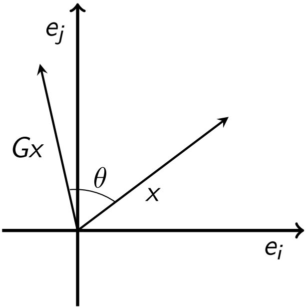
which deviates from identity at row and column i and j.

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.

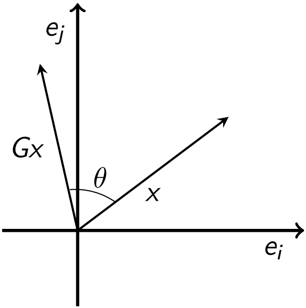


Definition (Givens rotation)

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

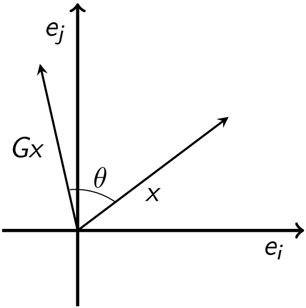
- $G^* = G^{-1}$

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

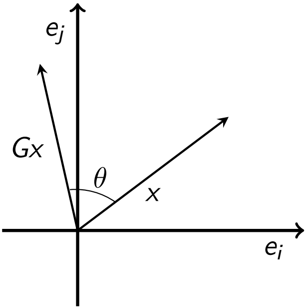
- $G^* = G^{-1}$
- Gz can be computed with $O(1)$ operations

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
- $\dots$

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

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

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

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

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

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

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

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

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

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

is a QR-factorization of A.

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

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 of 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 of 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 of 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 of 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, ... do
    // One Hessenberg QR step
    H = A_{k-1}
    for i = 1, ..., n-1 do
        [c_i, s_i] = givens(h_{i,i}, h_{i+1,i})
        H_{i:i+1,i:n} = [c_i  s_i; -s_i  c_i] H_{i:i+1,i:n}
    end for
    for i = 1, ..., n-1 do
        H_{1:i+1,i:i+1} = H_{1:i+1,i:i+1} [c_i  -s_i; s_i  c_i]
    end for
    A_k = H
end for
Return T = A_∞
```

$$H_{i:i+1,i:n} = \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} H_{i:i+1,i:n}, \qquad H_{1:i+1,i:i+1} = H_{1:i+1,i:i+1} \begin{bmatrix} c_i & -s_i \\ s_i & c_i \end{bmatrix}$$

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:

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method:

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

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method:

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

Note:

$$\bar{H} = RQ + \mu I =$$

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method:

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

Note:

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

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method:

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

Note:

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

Improvement 2: Acceleration with shifts (Section 2.3)

Shifted QR-method

One step of shifted QR-method:

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

Note:

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

$\Rightarrow$ 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

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

* Show QR-factorization of singular Hessenberg matrix in matlab *

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,

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

and

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,

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

and

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

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 next)

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 next)
- 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.

$\Rightarrow$ 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.

$\Rightarrow$ 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

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

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

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 characterization

- A generalization of power method: USI (Unnormalized simultaneous iteration)

Convergence theory - TB Chapter 28

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

Convergence characterization

- A generalization of power method: USI (Unnormalized simultaneous iteration)
- Show convergence properties of USI

Convergence theory - TB Chapter 28

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

Convergence characterization

- A generalization of power method: USI (Unnormalized simultaneous iteration)
- Show convergence properties of USI
- A variant Normalized Simultaneous Iteration (NSI)

Convergence theory - TB Chapter 28

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

Convergence characterization

- A generalization of power method: USI (Unnormalized simultaneous iteration)
- Show convergence properties of USI
- A variant Normalized Simultaneous Iteration (NSI)
- 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"

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

Define

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

Definition: Unnormalized simultaneous iteration (USI)

A generalization of power method with n vectors "simultaneously"

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

Define

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

A QR-factorization generalizes the normalization step:

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

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:

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

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 $\hat{Q} = (q_1, \ldots, q_m)$ 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 $\hat{Q} = (q_1, \ldots, q_m)$ are the eigenvectors.

Theorem (TB Theorem 28.1)

Suppose simultaneous iteration is started with $V^{(0)}$ and assumptions above are satisfied. Let q_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 $\hat{Q} = (q_1, \ldots, q_m)$ are the eigenvectors.

Theorem (TB Theorem 28.1)

Suppose simultaneous iteration is started with $V^{(0)}$ and assumptions above are satisfied. Let q_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

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

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

Normalized Simultaneous Iteration (NSI)

Variants of the power method. Equivalent:

(i) $v_k = \frac{A^k v_0}{\|A^k v_0\|}$

Normalized Simultaneous Iteration (NSI)

Variants of the power method. Equivalent:

(i) $v_k = \frac{A^k v_0}{\|A^k v_0\|}$

(ii) $v_k = \frac{A v_{k-1}}{\|A v_{k-1}\|}$

Normalized Simultaneous Iteration (NSI)

Variants of the power method. Equivalent:

(i) $v_k = \frac{A^k v_0}{\|A^k v_0\|}$

(ii) $v_k = \frac{A v_{k-1}}{\|A v_{k-1}\|}$

USI is a generlization of (i).

Normalized Simultaneous Iteration (NSI)

Variants of the power method. Equivalent:

(i) $v_k = \frac{A^k v_0}{\|A^k v_0\|}$

(ii) $v_k = \frac{A v_{k-1}}{\|A v_{k-1}\|}$

USI is a generlization of (i).
NSI is a generalization of (ii).

Normalized Simultaneous Iteration (NSI)

Variants of the power method. Equivalent:

(i) $v_k = \frac{A^k v_0}{||A^k v_0||}$

(ii) $v_k = \frac{A v_{k-1}}{||A v_{k-1}||}$

USI is a generlization of (i).
NSI is a generalization of (ii).

Algorithm: (Normalized) Simultaneous Iteration

- Input $\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$
- For $k = 1, \ldots,$
    - Set $Z = A\hat{Q}^{k-1}$
    - Compute QR-factorization $\hat{Q}^{(k)}\hat{R}^{(k)} = Z$

USI and NSI are equivalent.

USI and NSI are equivalent. More precisely:

Equivalence USI and NSI (TB Thm 28.2)

Suppose assumptions above are satisfied. If USI and NSI are started with the same vector they will generate the same sequence of matrices $\hat{Q}^k$ and $\hat{R}^k$.

Simultaneous iteration and QR-method

We will establish:

$$\text{basic QR-method} \Leftrightarrow \text{Simultaneous iteration with } \hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}.$$

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)} R^{(k)}$

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)} R^{(k)}$

QR-method satisfies

- $A^{(0)} = A$
- $A^{(k-1)} = Q^{(k)} R^{(k)}$
- $A^{(k)} = R^{(k)} Q^{(k)}$

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)}R^{(k)}$
- $A^{(k)} := (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)})$

QR-method satisfies

- $A^{(0)} = A$
- $A^{(k-1)} = Q^{(k)}R^{(k)}$
- $A^{(k)} = R^{(k)}Q^{(k)}$
- $\underline{Q}^{(k)} := Q^{(1)}\cdots Q^{(k)}$

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)}R^{(k)}$
- $A^{(k)} := (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)})$

QR-method satisfies

- $A^{(0)} = A$
- $A^{(k-1)} = Q^{(k)}R^{(k)}$
- $A^{(k)} = R^{(k)}Q^{(k)}$
- $\underline{Q}^{(k)} := Q^{(1)} \dots Q^{(k)}$

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)} R^{(k)}$
- $A^{(k)} := (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)})$

QR-method satisfies

- $A^{(0)} = A$
- $A^{(k-1)} = Q^{(k)} R^{(k)}$
- $A^{(k)} = R^{(k)} Q^{(k)}$
- $\underline{Q}^{(k)} := Q^{(1)} \dots Q^{(k)}$

Define: $\underline{R}^{(k)} := R^{(k)} \dots R^{(1)}$

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)} R^{(k)}$
- $A^{(k)} := (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)})$

QR-method satisfies

- $A^{(0)} = A$
- $A^{(k-1)} = Q^{(k)} R^{(k)}$
- $A^{(k)} = R^{(k)} Q^{(k)}$
- $\underline{Q}^{(k)} := Q^{(1)} \cdots Q^{(k)}$

Define: $\underline{R}^{(k)} := R^{(k)} \cdots R^{(1)}$

Essentially: The above equations generate the same sequence of matrices

Simultaneous iteration and QR-method

We will establish:

basic QR-method $\Leftrightarrow$ Simultaneous iteration with $\hat{Q}^{(0)} = I \in \mathbb{R}^{m\times m}$.

Simultaneous iteration satisfies

- $\underline{Q}^0 = I$
- $Z_k = A\underline{Q}^{(k-1)}$
- $Z_k = \underline{Q}^{(k)}R^{(k)}$
- $A^{(k)} := (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)})$

QR-method satisfies

- $A^{(0)} = A$
- $A^{(k-1)} = Q^{(k)}R^{(k)}$
- $A^{(k)} = R^{(k)}Q^{(k)}$
- $\underline{Q}^{(k)} := Q^{(1)}\cdots Q^{(k)}$

Define: $\underline{R}^{(k)} := R^{(k)}\cdots R^{(1)}$

Essentially: The above equations generate the same sequence of matrices

More precisely ...

TB Theorem 38.3:

Theorem (Equivalence simultaneous iteration and QR-method)

The above processes generate identical sequences of vectors. In particular,

$$A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$$

and

$$A^{(k)} = (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)}).$$

TB Theorem 38.3:

Theorem (Equivalence simultaneous iteration and QR-method)

The above processes generate identical sequences of vectors. In particular,

$$A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$$

and

$$A^{(k)} = (\underline{Q}^{(k)})^T A(\underline{Q}^{(k)}).$$

Beware: QR-factorization is not unique and equivalence only holds with one QR-factorization.

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy
$$q_i^{(k)} = \pm q_i + O(C^k).$$
where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy

$$q_i^{(k)} = \pm q_i + O(C^k).$$

  where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.
- $(A^k)_{i,j} = (q_i^{(k)})^T A q_j^{(k)}$
  - Diagonal $i = j$: $(A^{(k)})_{i,i} = (q_i^{(k)})^T A q_i^{(k)}$

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy

$$q_i^{(k)} = \pm q_i + O(C^k).$$

  where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.

- $(A^k)_{i,j} = (q_i^{(k)})^T A q_j^{(k)}$
  - Diagonal $i = j$: $(A^{(k)})_{i,i} = (q_i^{(k)})^T A q_i^{(k)} = r(q_i^{(k)}) =$Rayleigh quotient

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy

$$q_i^{(k)} = \pm q_i + O(C^k).$$

  where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.
- $(A^k)_{i,j} = (q_i^{(k)})^T A q_j^{(k)}$
  - Diagonal $i = j$: $(A^{(k)})_{i,i} = (q_i^{(k)})^T A q_i^{(k)} = r(q_i^{(k)})$ =Rayleigh quotient $\Rightarrow (A^{(k)})_{i,i} = \lambda_i + O(C^{2k})$

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy

$$q_i^{(k)} = \pm q_i + O(C^k).$$

  where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.

- $(A^k)_{i,j} = (q_i^{(k)})^T A q_j^{(k)}$
  - Diagonal $i = j$: $(A^{(k)})_{i,i} = (q_i^{(k)})^T A q_i^{(k)} = r(q_i^{(k)})$ =Rayleigh quotient $\Rightarrow (A^{(k)})_{i,i} = \lambda_i + O(C^{2k})$
  - Off-diagonal $i \neq j$: $(A^{(k)})_{i,j}$

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy
$$q_i^{(k)} = \pm q_i + O(C^k).$$
where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.
- $(A^k)_{i,j} = (q_i^{(k)})^T A q_j^{(k)}$
  - Diagonal $i = j$: $(A^{(k)})_{i,i} = (q_i^{(k)})^T A q_i^{(k)} = r(q_i^{(k)})$ =Rayleigh quotient $\Rightarrow (A^{(k)})_{i,i} = \lambda_i + O(C^{2k})$
  - Off-diagonal $i \neq j$: $(A^{(k)})_{i,j} = (q_i^{(k)})^T A q_j^{(k)} = O(C^k)$

Important property:

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

Consequences

- Recall from USI-NSI equivalence and USI convergence. The columns in $\hat{Q}^{(k)}$ satisfy
$$q_i^{(k)} = \pm q_i + O(C^k).$$
where $C = \max_{1<i<n} |\lambda_{i+1}|/|\lambda_i|$.
- $(A^k)_{i,j} = (q_i^{(k)})^T A q_j^{(k)}$
  - Diagonal $i = j$: $(A^{(k)})_{i,i} = (q_i^{(k)})^T A q_i^{(k)} = r(q_i^{(k)})$ =Rayleigh quotient $\Rightarrow (A^{(k)})_{i,i} = \lambda_i + O(C^{2k})$
  - Off-diagonal $i \neq j$: $(A^{(k)})_{i,j} = (q_i^{(k)})^T A q_j^{(k)} = O(C^k)$

Hence, $A^{(k)}$ will approach a triangular matrix

* Matlab demo *