Shifted QR Algorithm for Eigenvalue Problems

Rafikul Alam
Department of Mathematics
Indian Institute of Technology Guwahati
Guwahati - 781039, INDIA

Outline

- Hessenberg QR algorithm
- Shifted QR algorithm
- Wilkinson shift
- Tridiagonal QR with Wilkinson shift

Hessenberg QR algorithm

A QR step applied to a singular Hessenberg matrix A may not preserve the Hessenberg structure. Consider

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R}$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{A_1} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{Q}.$$

Hessenberg QR algorithm

A QR step applied to a singular Hessenberg matrix A may not preserve the Hessenberg structure. Consider

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R}$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{A_1} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{Q}.$$

Note that A is singular and Hessenberg but A_1 is not Hessenberg. The QR factorization of a singular matrix is not unique. Among many QR factorization, some may not preserve the Hessenberg structure when they are used in a QR step.

Hessenberg QR algorithm

A QR step applied to a singular Hessenberg matrix A may not preserve the Hessenberg structure. Consider

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R}$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{A_1} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{Q}.$$

Note that A is singular and Hessenberg but A_1 is not Hessenberg. The QR factorization of a singular matrix is not unique. Among many QR factorization, some may not preserve the Hessenberg structure when they are used in a QR step. However, one can choose a QR factorization that preserves the Hessenberg structure.

Hessenberg QR algorithm

Theorem Let A be a Hessenberg matrix. If a QR factorization $A = QR$ is computed using Givens rotations then $A_1 := RQ$ is a Hessenberg matrix. Further, a QR step applied to A requires $8n^2$ flops.

Hessenberg QR algorithm

Theorem Let A be a Hessenberg matrix. If a QR factorization $A = QR$ is computed using Givens rotations then $A_1 := RQ$ is a Hessenberg matrix. Further, a QR step applied to A requires $8n^2$ flops.

Proof: Since A is a Hessenberg matrix, it has only $n-1$ nonzero subdiagonal entries at $(2,1),(3,2),\ldots,(n,n-1)$ positions. We can transfom A to upper triangular form by using $n-1$ Givens rotations to create zeros at $(2,1),(3,2),\ldots,(n,n-1)$ entries of A.

Hessenberg QR algorithm

Theorem Let A be a Hessenberg matrix. If a QR factorization $A = QR$ is computed using Givens rotations then $A_1 := RQ$ is a Hessenberg matrix. Further, a QR step applied to A requires $8n^2$ flops.

Proof: Since A is a Hessenberg matrix, it has only $n-1$ nonzero subdiagonal entries at $(2,1),(3,2),\ldots,(n,n-1)$ positions. We can transfom A to upper triangular form by using $n-1$ Givens rotations to create zeros at $(2,1),(3,2),\ldots,(n,n-1)$ entries of A.

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}}_{A} \longrightarrow \underbrace{\begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}}_{Q_1^* A} \longrightarrow \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & * & * & * & * \\ & & * & * & * \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}}_{Q_2^* Q_1^* A}.$$

Let Q_1 be a rotation in the x_1-x_2 plane such that $Q_1^* A$ has a zero at the $(2,1)$ entry. The transformation $A \mapsto Q_1^* A$ alters only the first and second rows of A and costs $8n$ flops.

Hessenberg QR algorithm

Next, let Q_2 be a rotation in the x_2-x_3 plane such that $Q_2^* Q_1^* A$ has a zero at $(3,2)$ entry. The transformation $Q_1^* A \longmapsto Q_2^* Q_1^* A$ alters only the second and third rows of $Q_1^* A$ and does not affect the zero at $(2,1)$ entry of $Q_1^* A$.

Hessenberg QR algorithm

Next, let Q_2 be a rotation in the x_2-x_3 plane such that $Q_2^* Q_1^* A$ has a zero at $(3, 2)$ entry. The transformation $Q_1^* A \longmapsto Q_2^* Q_1^* A$ alters only the second and third rows of $Q_1^* A$ and does not affect the zero at $(2, 1)$ entry of $Q_1^* A$. Thus $Q_2^* Q_1^* A$ has zeros at $(2, 1)$ and $(3, 2)$ entries. It costs $8(n - 1)$ flops.

Hessenberg QR algorithm

Next, let Q_2 be a rotation in the x_2-x_3 plane such that $Q_2^* Q_1^* A$ has a zero at $(3,2)$ entry. The transformation $Q_1^* A \longmapsto Q_2^* Q_1^* A$ alters only the second and third rows of $Q_1^* A$ and does not affect the zero at $(2,1)$ entry of $Q_1^* A$. Thus $Q_2^* Q_1^* A$ has zeros at $(2,1)$ and $(3,2)$ entries. It costs $8(n-1)$ flops.

Next, let Q_3 be a rotation in the x_3-x_4 plane such that $Q_3^* Q_2^* Q_1^* A$ has a zero in the $(4,3)$ position. Since Q_3 alters only the third and fourth rows, $Q_3^* Q_2^* Q_1^* A$ has zeros at $(2,1), (3,2)$ and $(4,3)$ entries.

Hessenberg QR algorithm

Next, let Q_2 be a rotation in the x_2-x_3 plane such that $Q_2^* Q_1^* A$ has a zero at $(3, 2)$ entry. The transformation $Q_1^* A \longmapsto Q_2^* Q_1^* A$ alters only the second and third rows of $Q_1^* A$ and does not affect the zero at $(2, 1)$ entry of $Q_1^* A$. Thus $Q_2^* Q_1^* A$ has zeros at $(2, 1)$ and $(3, 2)$ entries. It costs $8(n-1)$ flops.

Next, let Q_3 be a rotation in the x_3-x_4 plane such that $Q_3^* Q_2^* Q_1^* A$ has a zero in the $(4, 3)$ position. Since Q_3 alters only the third and fourth rows, $Q_3^* Q_2^* Q_1^* A$ has zeros at $(2, 1), (3, 2)$ and $(4, 3)$ entries.

Continuing in this manner, we have an upper triangular matrix

$$R = Q_{n-1}^* Q_{n-2}^* \cdots Q_1^* A \Longrightarrow A = \underbrace{Q_1 Q_2 \cdots Q_{n-1}}_{Q} R = QR.$$

The QR factorization requires $\sum_{k=1}^{n-1} 8(n-k) \sim 4n^2$ flops.

Hessenberg QR algorithm

Now, we compute $A_1 = RQ = RQ_1Q_2 \cdots Q_{n-1}$. The transformation $R \mapsto RQ_1$ alters only the first and second columns of R by their linear combinations and creates a nonzero entry at $(2,1)$ entry of RQ_1.

Hessenberg QR algorithm

Now, we compute $A_1 = RQ = RQ_1 Q_2 \cdots Q_{n-1}$. The transformation $R \mapsto RQ_1$ alters only the first and second columns of R by their linear combinations and creates a nonzero entry at $(2,1)$ entry of RQ_1. The transformation $RQ_1 \longmapsto RQ_1 Q_2$ alters only the second and third columns by their linear combinations and creates a nonzero entry at $(3,2)$ entry of $RQ_1 Q_2$. Schematically

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{R} \longrightarrow \underbrace{\begin{bmatrix} * & * & \times & \times & \times \\ * & * & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{RQ_1} \longrightarrow \underbrace{\begin{bmatrix} \times & * & * & \times & \times \\ \times & * & * & \times & \times \\ & * & * & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{RQ_1 Q_2}.$$

Thus $RQ_1 Q_2$ has nonzero entries at $(2,1)$ and $(3,2)$ positions.

Hessenberg QR algorithm

Now, we compute $A_1 = RQ = RQ_1Q_2\cdots Q_{n-1}$. The transformation $R \mapsto RQ_1$ alters only the first and second columns of R by their linear combinations and creates a nonzero entry at $(2,1)$ entry of RQ_1. The transformation $RQ_1 \longmapsto RQ_1Q_2$ alters only the second and third columns by their linear combinations and creates a nonzero entry at $(3,2)$ entry of RQ_1Q_2. Schematically

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{R} \longrightarrow \underbrace{\begin{bmatrix} * & * & \times & \times & \times \\ * & * & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{RQ_1} \longrightarrow \underbrace{\begin{bmatrix} \times & * & * & \times & \times \\ \times & * & * & \times & \times \\ & * & * & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{RQ_1Q_2}.$$

Thus RQ_1Q_2 has nonzero entries at $(2,1)$ and $(3,2)$ positions. Continuing in this manner, $R \longmapsto RQ_1Q_2\cdots Q_{n-1} = A_1$ introduces nonzero entries at $(2,1),(3,2),\ldots,(n,n-1)$ positions of A_1 showing that A_1 is a Hessenberg matrix.

Hessenberg QR algorithm

Now, we compute $A_1 = RQ = RQ_1 Q_2 \cdots Q_{n-1}$. The transformation $R \mapsto RQ_1$ alters only the first and second columns of R by their linear combinations and creates a nonzero entry at $(2,1)$ entry of RQ_1. The transformation $RQ_1 \longmapsto RQ_1 Q_2$ alters only the second and third columns by their linear combinations and creates a nonzero entry at $(3,2)$ entry of $RQ_1 Q_2$. Schematically

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{R} \longrightarrow \underbrace{\begin{bmatrix} * & * & \times & \times & \times \\ * & * & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{RQ_1} \longrightarrow \underbrace{\begin{bmatrix} \times & * & * & \times & \times \\ \times & * & * & \times & \times \\ & * & * & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}_{RQ_1 Q_2}.$$

Thus $RQ_1 Q_2$ has nonzero entries at $(2,1)$ and $(3,2)$ positions. Continuing in this manner, $R \longmapsto RQ_1 Q_2 \cdots Q_{n-1} = A_1$ introduces nonzero entries at $(2,1), (3,2), \ldots, (n, n-1)$ positions of A_1 showing that A_1 is a Hessenberg matrix. Note that the transformation $R \mapsto RQ$ requires $4n^2$ flops. Hence a QR step applied to A requires $8n^2$ flops. $\blacksquare$

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Definition: A Hessenberg matrix A is said to be proper Hessenberg if $A(j+1, j) \neq 0$ for $j = 1 : n-1$. In other words, all sub-diagonal entries of A are nonzero.

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Definition: A Hessenberg matrix A is said to be proper Hessenberg if $A(j+1, j) \neq 0$ for $j = 1 : n-1$. In other words, all sub-diagonal entries of A are nonzero.

Note that if A is proper Hessenberg then the first $n-1$ columns of A are linearly independent. Hence $\mathrm{rank}(A) \geq n-1$.

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Definition: A Hessenberg matrix A is said to be proper Hessenberg if $A(j+1, j) \neq 0$ for $j = 1 : n-1$. In other words, all sub-diagonal entries of A are nonzero.

Note that if A is proper Hessenberg then the first $n-1$ columns of A are linearly independent. Hence $\mathrm{rank}(A) \geq n-1$.

If A is Hessenberg and if $A(j+1, j) = 0$ then A is block upper triangular and of the form

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & A_{22} \end{array}\right]$$

where $A_{11} \in \mathbb{C}^{j \times j}$ and $A_{22} \in \mathbb{C}^{(n-j)\times(n-j)}$ are Hessenberg matrices.

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Definition: A Hessenberg matrix A is said to be proper Hessenberg if $A(j+1, j) \neq 0$ for $j = 1 : n-1$. In other words, all sub-diagonal entries of A are nonzero.

Note that if A is proper Hessenberg then the first $n-1$ columns of A are linearly independent. Hence $\mathrm{rank}(A) \geq n-1$.

If A is Hessenberg and if $A(j+1, j) = 0$ then A is block upper triangular and of the form

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & A_{22} \end{array}\right]$$

where $A_{11} \in \mathbb{C}^{j\times j}$ and $A_{22} \in \mathbb{C}^{(n-j)\times(n-j)}$ are Hessenberg matrices.

Note that $\Lambda(A) = \Lambda(A_{11}) \cup \Lambda(A_{22})$.

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Definition: A Hessenberg matrix A is said to be proper Hessenberg if $A(j+1, j) \neq 0$ for $j = 1 : n-1$. In other words, all sub-diagonal entries of A are nonzero.

Note that if A is proper Hessenberg then the first $n-1$ columns of A are linearly independent. Hence $\mathrm{rank}(A) \geq n-1$.

If A is Hessenberg and if $A(j+1, j) = 0$ then A is block upper triangular and of the form

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & A_{22} \end{array}\right]$$

where $A_{11} \in \mathbb{C}^{j \times j}$ and $A_{22} \in \mathbb{C}^{(n-j)\times(n-j)}$ are Hessenberg matrices.

Note that $\Lambda(A) = \Lambda(A_{11}) \cup \Lambda(A_{22})$. Hence eigenvalues of A_{11} and A_{22} can be computed independently.

Proper Hessenberg matrix

If A is symmetric and tridiagonal then a QR step using rotations applied to A preserves the tridiangonal form and requires only $12n$ flops. Indeed, $A = QR$ requires $6n$ flops and $A_1 = RQ$ requires $6n$ flops.

Definition: A Hessenberg matrix A is said to be proper Hessenberg if $A(j+1, j) \neq 0$ for $j = 1 : n-1$. In other words, all sub-diagonal entries of A are nonzero.

Note that if A is proper Hessenberg then the first $n-1$ columns of A are linearly independent. Hence $\mathrm{rank}(A) \geq n-1$.

If A is Hessenberg and if $A(j+1, j) = 0$ then A is block upper triangular and of the form

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & A_{22} \end{array}\right]$$

where $A_{11} \in \mathbb{C}^{j \times j}$ and $A_{22} \in \mathbb{C}^{(n-j)\times(n-j)}$ are Hessenberg matrices.

Note that $\Lambda(A) = \Lambda(A_{11}) \cup \Lambda(A_{22})$. Hence eigenvalues of A_{11} and A_{22} can be computed independently. Therefore, we assume that A is proper Hessenberg.The fact is that proper Hessenberg form is also important for convergence of QR algorithm.

Proper Hessenberg QR Algorithm

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proper Hessenberg QR Algorithm

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proof. Note that $\mathrm{rank}(A) = n-1$ and the first $n-1$ columns of A are linearly independent.

Proper Hessenberg QR Algorithm

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proof. Note that $\mathrm{rank}(A) = n-1$ and the first $n-1$ columns of A are linearly independent. Now $A = QR \Longrightarrow \mathrm{rank}(R) = n-1$ and that the first $n-1$ columns of A are linearly independent.

Proper Hessenberg QR Algorithm

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proof. Note that $\mathrm{rank}(A) = n-1$ and the first $n-1$ columns of A are linearly independent. Now $A = QR \Longrightarrow \mathrm{rank}(R) = n-1$ and that the first $n-1$ columns of A are linearly independent.Hence the last row of R is a zero row which implies that the last row of $RQ = A_1$ is a zero row. ■

Proper Hessenberg QR Algorithm

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proof. Note that $\mathrm{rank}(A) = n-1$ and the first $n-1$ columns of A are linearly independent. Now $A = QR \Longrightarrow \mathrm{rank}(R) = n-1$ and that the first $n-1$ columns of A are linearly independent.Hence the last row of R is a zero row which implies that the last row of $RQ = A_1$ is a zero row. ■

Observation: If A is proper Hessenberg and singular then QR algorithm takes just one QR step to compute the zero eigenvalue.

Proper Hessenberg QR Algorithm

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A} & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proof. Note that $\mathrm{rank}(A) = n-1$ and the first $n-1$ columns of A are linearly independent. Now $A = QR \implies \mathrm{rank}(R) = n-1$ and that the first $n-1$ columns of A are linearly independent.Hence the last row of R is a zero row which implies that the last row of $RQ = A_1$ is a zero row. ■

Observation: If A is proper Hessenberg and singular then QR algorithm takes just one QR step to compute the zero eigenvalue. This suggests a shifting strategy:

Proper Hessenberg QR Algorithm

rank(A) >= (n-1) rank(A) < n

Theorem Let A be proper Hessenberg and singular. Consider a QR step $A = QR$ and $A_1 := RQ$. Then

$$A_1 = \left[\begin{array}{ccc|c} & \widehat{A} & & \begin{matrix} * \\ \vdots \\ * \end{matrix} \\ \hline 0 & \cdots & 0 & 0 \end{array}\right],$$

where $\widehat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$ is Hessenberg.

Proof. Note that $\operatorname{rank}(A) = n-1$ and the first $n-1$ columns of A are linearly independent. Now $A = QR \Longrightarrow \operatorname{rank}(R) = n-1$ and that the first $n-1$ columns of ~~A~~ R are linearly independent.Hence the last row of R is a zero row which implies that the last row of $RQ = A_1$ is a zero row. ■

Observation: If A is proper Hessenberg and singular then QR algorithm takes just one QR step to compute the zero eigenvalue. This suggests a shifting strategy: If $A - \mu I$ is nearly singular then QR algorithm applied to $A - \mu I$ can result in faster convergence.

Shifted QR algorithm

Henceforth a QR factorization of a Hessenberg or tridigonal matrix will always mean a Givens QR factorization.

Shifted QR algorithm

Henceforth a QR factorization of a Hessenberg or tridigonal matrix will always mean a Givens QR factorization. Let A_m denote the matrix obtained after m QR steps performed on A.

Shifted QR algorithm

Henceforth a QR factorization of a Hessenberg or tridigonal matrix will always mean a Givens QR factorization. Let A_m denote the matrix obtained after m QR steps performed on A.

A QR step applied to A_{m-1} computes a QR factorization $A_{m-1} = Q_m R_m$ and performs the similarity transformation $A_m := Q_m^* A_{m-1} Q_m = R_m Q_m$.

Shifted QR algorithm

Henceforth a QR factorization of a Hessenberg or tridigonal matrix will always mean a Givens QR factorization. Let A_m denote the matrix obtained after m QR steps performed on A.

A QR step applied to A_{m-1} computes a QR factorization $A_{m-1} = Q_m R_m$ and performs the similarity transformation $A_m := Q_m^* A_{m-1} Q_m = R_m Q_m$.

Let $\mu_{m-1} \in \mathbb{C}$. Compute a QR factorization $A_{m-1} - \mu_{m-1} I = Q_m R_m$. Then

$$Q_m^*(A_{m-1} - \mu_{m-1})Q_m = R_m Q_m$$

Shifted QR algorithm

Henceforth a QR factorization of a Hessenberg or tridigonal matrix will always mean a Givens QR factorization. Let A_m denote the matrix obtained after m QR steps performed on A.

A QR step applied to A_{m-1} computes a QR factorization $A_{m-1} = Q_m R_m$ and performs the similarity transformation $A_m := Q_m^* A_{m-1} Q_m = R_m Q_m$.

Let $\mu_{m-1} \in \mathbb{C}$. Compute a QR factorization $A_{m-1} - \mu_{m-1} I = Q_m R_m$. Then

$$Q_m^*(A_{m-1} - \mu_{m-1})Q_m = R_m Q_m \Longrightarrow A_m := Q^* A_{m-1} Q = R_m Q_m + \mu_{m-1} I.$$

This yields the shifted QR algorithm

Shifted QR algorithm

Henceforth a QR factorization of a Hessenberg or tridigonal matrix will always mean a Givens QR factorization. Let A_m denote the matrix obtained after m QR steps performed on A.

A QR step applied to A_{m-1} computes a QR factorization $A_{m-1} = Q_m R_m$ and performs the similarity transformation $A_m := Q_m^* A_{m-1} Q_m = R_m Q_m$.

Let $\mu_{m-1} \in \mathbb{C}$. Compute a QR factorization $A_{m-1} - \mu_{m-1} I = Q_m R_m$. Then

$$Q_m^*(A_{m-1} - \mu_{m-1})Q_m = R_m Q_m \Longrightarrow A_m := Q^* A_{m-1} Q = R_m Q_m + \mu_{m-1} I.$$

This yields the shifted QR algorithm

Choose shift $\mu_{m-1} \in \mathbb{C}$,
Compute QR factorization $A_{m-1} - \mu_{m-1} I = Q_m R_m$,
Perform similarity transformation $A_m := R_m Q_m + \mu_{m-1} I$.

The shifted QR algorithm has potential to achieve a fast convergence.

Deflation

Theorem Let A_{m-1} be proper Hessenberg. Let λ be an eigenvalue of A. Consider a QR step $A_{m-1} - \lambda I = Q_m R_m$ and $A_m := R_m Q_m + \lambda I$. Then

$$A_m = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & \lambda \end{array}\right],$$

where $\widehat{A}_m$ is Hessenberg. Set $U_m := Q_1 Q_2 \cdots Q_m$ and $u := U_m e_n$. Then $A_m = U_m^* A U_m$ and (λ, u) is a left eigenpair of A, that is, $u^* A = \lambda u^*$.

Deflation

Theorem Let A_{m-1} be proper Hessenberg. Let λ be an eigenvalue of A. Consider a QR step $A_{m-1} - \lambda I = Q_m R_m$ and $A_m := R_m Q_m + \lambda I$. Then

$$A_m = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & \lambda \end{array}\right],$$

where $\widehat{A}_m$ is Hessenberg. Set $U_m := Q_1 Q_2 \cdots Q_m$ and $u := U_m e_n$. Then $A_m = U_m^* A U_m$ and (λ, u) is a left eigenpair of A, that is, $u^* A = \lambda u^*$.

????? HOW? => As Am-1 will also have lambda as eigenvalue, bcoz of similarity trasnformation.

Proof. Note that $\operatorname{rank}(A_{m-1} - \lambda I) < n$. Since $A_{m-1} - \lambda I$ is proper Hessenberg, the first $n-1$ columns of $A_{m-1} - \lambda I$ are linearly independent. Consequently, $\operatorname{rank}(A_{m-1} - \lambda I) = n - 1$. Now $A_{m-1} - \lambda I = Q_m R_m \Longrightarrow \operatorname{rank}(R_m) = n - 1$.

Deflation

Since R_m is upper triangular and the first $n-1$ columns of R_m are linearly independent, the last row of R_m must be a zero row, that is,

$$R_m = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{R}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right].$$

Deflation

Since R_m is upper triangular and the first $n-1$ columns of R_m are linearly independent, the last row of R_m must be a zero row, that is,

$$R_m = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{R}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right].$$

Consequently, the last row of $R_m Q_m$ is a zero row. Hence we have

$$A_m = R_m Q_m + \lambda I = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & \lambda \end{array}\right].$$

Deflation

Since R_m is upper triangular and the first $n-1$ columns of R_m are linearly independent, the last row of R_m must be a zero row, that is,

$$R_m = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{R}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & 0 \end{array}\right].$$

Consequently, the last row of $R_m Q_m$ is a zero row. Hence we have

$$A_m = R_m Q_m + \lambda I = \left[\begin{array}{ccc|c} & & & * \\ & \widehat{A}_m & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & \lambda \end{array}\right].$$

Note that $A_m = Q_m^* A_{m-1} Q_m = Q_m^* Q_{m-1}^* A_{m-2} Q_{m-1} Q_m = U_m^* A U_m$ and that $e_n^* A_m = \lambda e_n^* \Longrightarrow u^* A = \lambda u^*$. ■

Observation

- Shifted QR algorithm applied to a proper Hessenberg matrix takes just one iteration to converge when an eigenvalue of the matrix is chosen as the shift. This indicates that a good shifting strategy can accelerate convergence of the QR algorithm.

Observation

- Shifted QR algorithm applied to a proper Hessenberg matrix takes just one iteration to converge when an eigenvalue of the matrix is chosen as the shift. This indicates that a good shifting strategy can accelerate convergence of the QR algorithm.
- If $a_{nn-1}^{(m)} \to 0$ as $m \to \infty$ then the entry $a_{nn}^{(m)}$ of A_m converges to an eigenvalue of A. Thus after a QR step, if $|a_{nn-1}^{(m)}|$ is sufficiently small then set $a_{nn-1}^{(m)} = 0$ and accept $\lambda := a_{nn}^{(m)}$ as an eigenvalue of A.

Observation

- Shifted QR algorithm applied to a proper Hessenberg matrix takes just one iteration to converge when an eigenvalue of the matrix is chosen as the shift. This indicates that a good shifting strategy can accelerate convergence of the QR algorithm.
- If $a^{(m)}_{nn-1} \to 0$ as $m \to \infty$ then the entry $a^{(m)}_{nn}$ of A_m converges to an eigenvalue of A. Thus after a QR step, if $|a^{(m)}_{nn-1}|$ is sufficiently small then set $a^{(m)}_{nn-1} = 0$ and accept $\lambda := a^{(m)}_{nn}$ as an eigenvalue of A.
- In practice, we set $a^{(m)}_{nn-1} = 0$ when $|a^{(m)}_{nn-1}| \leq \mathbf{u}(|a^{(m)}_{nn} + a^{(m)}_{n-1,n-1}|)$, where $\mathbf{u}$ is the unit roundoff. Hence we can delete the last row and last column of A_m and continue shifted QR algorithm with $(n-1) \times (n-1)$ matrix $\widehat{A}_m$. This process is called deflation.

Observation

- Shifted QR algorithm applied to a proper Hessenberg matrix takes just one iteration to converge when an eigenvalue of the matrix is chosen as the shift. This indicates that a good shifting strategy can accelerate convergence of the QR algorithm.
- If $a_{nn-1}^{(m)} \to 0$ as $m \to \infty$ then the entry $a_{nn}^{(m)}$ of A_m converges to an eigenvalue of A. Thus after a QR step, if $|a_{nn-1}^{(m)}|$ is sufficiently small then set $a_{nn-1}^{(m)} = 0$ and accept $\lambda := a_{nn}^{(m)}$ as an eigenvalue of A.
- In practice, we set $a_{nn-1}^{(m)} = 0$ when $|a_{nn-1}^{(m)}| \leq \mathbf{u}(|a_{nn}^{(m)} + a_{n-1,n-1}^{(m)}|)$, where $\mathbf{u}$ is the unit roundoff. Hence we can delete the last row and last column of A_m and continue shifted QR algorithm with $(n-1) \times (n-1)$ matrix $\widehat{A}_m$. This process is called deflation.
- Deflation progressively leads to smaller and smaller problem until all the eigenvalues of A are computed.

Shifted Hessenberg QR algorithm

Shifted QR algorithm converges locally quadratically for Hessenberg matrices and the convergence is cubic for Hermitian tridiagonal matrices.

Algorithm. (Shifted QR algorithm)
Input: An $n \times n$ Hessenberg matrix A
Output: Upper triangular matrix $T = Q^* AQ$

$A_0 := A$
for $m = 1, 2, \ldots$
    Choose a shift μ_{m-1}
    $A_{m-1} - \mu_{m-1} I = Q_m R_m$ % QR factorization
    $A_m := R_m Q_m + \mu_{m-1} I$ % similarity transform $Q_m^* A_{m-1} Q_m$
end

Shifted Hessenberg QR algorithm

Shifted QR algorithm converges locally quadratically for Hessenberg matrices and the convergence is cubic for Hermitian tridiagonal matrices.

Algorithm. (Shifted QR algorithm)
Input: An $n \times n$ Hessenberg matrix A
Output: Upper triangular matrix $T = Q^* A Q$

$A_0 := A$
for $m = 1, 2, \ldots$
  Choose a shift μ_{m-1}
  $A_{m-1} - \mu_{m-1} I = Q_m R_m$ % QR factorization
  $A_m := R_m Q_m + \mu_{m-1} I$ % similarity transform $Q_m^* A_{m-1} Q_m$
end

The aim of the shifted QR algorithm is to choose the shift parameters that speed up convergence. How to choose a good shift?

Rayleigh quotient shift

The subdiagonal entry $a_{nn-1}^{(m)}$ of A_m tends to zero when the shift is close to an eigenvalue of A. Then $a_{nn}^{(m)}$ becomes close to an eigenvalue of A.

Rayleigh quotient shift

The subdiagonal entry $a_{nn-1}^{(m)}$ of A_m tends to zero when the shift is close to an eigenvalue of A. Then $a_{nn}^{(m)}$ becomes close to an eigenvalue of A.

Hence $\mu_m := a_{nn}^{(m)} = e_n^* A_m e_n$ can be used as a shift. Since μ_m is the Rayleigh quotient of A_m at e_n, it is known as Rayleigh quotient shift.

Rayleigh quotient shift

The subdiagonal entry $a_{nn-1}^{(m)}$ of A_m tends to zero when the shift is close to an eigenvalue of A. Then $a_{nn}^{(m)}$ becomes close to an eigenvalue of A.

Hence $\mu_m := a_{nn}^{(m)} = e_n^* A_m e_n$ can be used as a shift. Since μ_m is the Rayleigh quotient of A_m at e_n, it is known as Rayleigh quotient shift.

However, it is well known that the Rayleigh quotient shift strategy does not always work. It fails for the symmetric matrix $A := \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ which has eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 3$.

Rayleigh quotient shift

The subdiagonal entry $a_{nn-1}^{(m)}$ of A_m tends to zero when the shift is close to an eigenvalue of A. Then $a_{nn}^{(m)}$ becomes close to an eigenvalue of A.

Hence $\mu_m := a_{nn}^{(m)} = e_n^* A_m e_n$ can be used as a shift. Since μ_m is the Rayleigh quotient of A_m at e_n, it is known as Rayleigh quotient shift.

However, it is well known that the Rayleigh quotient shift strategy does not always work. It fails for the symmetric matrix $A := \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ which has eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 3$.

The Rayleigh quotient shift is $\mu = 2$, which is the mid point between the eigenvalues. Then

$$A - 2I = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{Q_1} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{R_1}$$

shows that $Q_1 := A - 2I$ and $R_1 := I$. Thus $A_1 = R_1 Q_1 + 2I = A$. This shows that the QR step leaves the matrix A unchanged.

Wilkinson shift

Rayleigh quotient shift fails occasionally. By contrast, Wilkinson shift fails rarely. The Wilkinson shift is that eigenvalue of the submatrix

$$\begin{bmatrix} a_{n-1,n-1}^{(m-1)} & a_{n-1,n}^{(m-1)} \\ a_{n,n-1}^{(m-1)} & a_{nn}^{(m-1)} \end{bmatrix}$$

of A_{m-1} which is closest to $a_{nn}^{(m-1)}$. In the case of complex conjugate eigenvalues, use both the eigenvalues as shifts one after another.

Wilkinson shift

Rayleigh quotient shift fails occasionally. By contrast, Wilkinson shift fails rarely. The Wilkinson shift is that eigenvalue of the submatrix

$$\begin{bmatrix} a_{n-1,n-1}^{(m-1)} & a_{n-1,n}^{(m-1)} \\ a_{n,n-1}^{(m-1)} & a_{nn}^{(m-1)} \end{bmatrix}$$

of A_{m-1} which is closest to $a_{nn}^{(m-1)}$. In the case of complex conjugate eigenvalues, use both the eigenvalues as shifts one after another.

For real symmetric tridiagonal matrices, the shifted QR algorithm with Wilkinson shift always converges and the rate of convergence is usually cubic or better. However, for general matrices there still remain some very special cases for which the Wilkinson shift fails.

Wilkinson shift

Rayleigh quotient shift fails occasionally. By contrast, Wilkinson shift fails rarely. The Wilkinson shift is that eigenvalue of the submatrix

$$\begin{bmatrix} a_{n-1,n-1}^{(m-1)} & a_{n-1,n}^{(m-1)} \\ a_{n,n-1}^{(m-1)} & a_{nn}^{(m-1)} \end{bmatrix}$$

of A_{m-1} which is closest to $a_{nn}^{(m-1)}$. In the case of complex conjugate eigenvalues, use both the eigenvalues as shifts one after another.

For real symmetric tridiagonal matrices, the shifted QR algorithm with Wilkinson shift always converges and the rate of convergence is usually cubic or better. However, for general matrices there still remain some very special cases for which the Wilkinson shift fails.

Wilkinson shift strategy works very well for the vast majority of matrices. On an average, the eigenvalues start to emerge after a few QR steps. For Hermitian matrices the situation is even better - about two to three QR steps are needed per eigenvalue.

Wilkinson shift

For real symmetric tridiagonal matrices the Wilkinson shift has a very elegant formula. Suppose that A is real symmetric and tridiagonal

$$A := \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix}.$$

For Wilkinson shift we choose the eigenvalues of $\begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$ that is closest to a_n.

Wilkinson shift

For real symmetric tridiagonal matrices the Wilkinson shift has a very elegant formula. Suppose that A is real symmetric and tridiagonal

$$A := \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix}.$$

For Wilkinson shift we choose the eigenvalues of $\begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$ that is closest to a_n. The Wilkinson shift is given by

$$\mu := a_n + \delta - \operatorname{sign}(\delta)\sqrt{\delta^2 + b_{n-1}^2}, \text{ where } \delta := \frac{a_{n-1} - a_n}{2}.$$

This formula is a numerically stable method for computing the Wilkinson shift.

Tridiagonal shifted QR algorithm

The final algorithm for calculating eigenvalues of a real symmetric tridiagonal matrix is given below.

Algorithm. (Tridiagonal shifted QR algorithm)
Input: An $n \times n$ real symmetric tridiagonal matrix A
Output: A diagonal matrix $D = Q^* A Q$

$$
\begin{array}{l}
\mathtt{u} := \mathtt{eps}/2 \\
\mathtt{for}\ \ \mathtt{k} = \mathtt{n}: -1: 1 \\
\qquad \mathtt{while}\ \ |\mathtt{A}(\mathtt{k},\mathtt{k}-1)| > \mathtt{u}(|\mathtt{A}(\mathtt{k}-1,\mathtt{k}-1) + |\mathtt{A}(\mathtt{k},\mathtt{k})|) \\
\qquad\qquad \delta := (\mathtt{A}(\mathtt{k}-1,\mathtt{k}-1) - \mathtt{A}(\mathtt{k},\mathtt{k}))/2 \\
\qquad\qquad \mu := \mathtt{A}(\mathtt{k},\mathtt{k}) + \delta - \mathrm{sign}(\delta)\sqrt{\delta^2 + |\mathtt{A}(\mathtt{k},\mathtt{k}-1)|^2} \\
\qquad\qquad \mathtt{A}(1:\mathtt{k}, 1:\mathtt{k}) - \mu \mathtt{I} = \mathtt{QR} \\
\qquad\qquad \mathtt{A}(1:\mathtt{k}, 1:\mathtt{k}) := \mathtt{RQ} + \mu \mathtt{I} \\
\qquad \mathtt{end} \\
\qquad \mathtt{A}(\mathtt{k},\mathtt{k}-1) := 0 \qquad \text{\% accept } \mathtt{A(k,k)} \text{ as eigenvalue} \\
\qquad \mathtt{A}(\mathtt{k}-1,\mathtt{k}) := 0 \qquad \text{\% accept } \mathtt{A(k,k)} \text{ as eigenvalue} \\
\mathtt{end}
\end{array}
$$

Cubic convergence of tridiagonal QR algorithm

Let A_m be the result of m steps of QR iteration with Wilkinson shift applied to A. The cubic rate of convergence $\Rightarrow$ the subdiagonal entry $b_{n-1}^{(m)}$ of A_m converges to zero cubically, that is, $|b_{n-1}^{(m)}| = \mathcal{O}|b_{n-1}^{(m-1)}|^3)$.

Cubic convergence of tridiagonal QR algorithm

Let A_m be the result of m steps of QR iteration with Wilkinson shift applied to A. The cubic rate of convergence $\Rightarrow$ the subdiagonal entry $b_{n-1}^{(m)}$ of A_m converges to zero cubically, that is, $|b_{n-1}^{(m)}| = \mathcal{O}|b_{n-1}^{(m-1)}|^3)$.

Thus $a_n^{(m)}$ converges to an eigenvalue of A and in each iteration the number of correct digits is increased by a multiple of 3.

Cubic convergence of tridiagonal QR algorithm

Let A_m be the result of m steps of QR iteration with Wilkinson shift applied to A. The cubic rate of convergence $\Rightarrow$ the subdiagonal entry $b_{n-1}^{(m)}$ of A_m converges to zero cubically, that is, $|b_{n-1}^{(m)}| = \mathcal{O}|b_{n-1}^{(m-1)}|^3)$.

Thus $a_n^{(m)}$ converges to an eigenvalue of A and in each iteration the number of correct digits is increased by a multiple of 3.

Example: Let $A := \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix}$. The eigenvalues are given by $\Lambda(A) := \{-3.4142, -2.0000, -0.58579\}$. The QR algorithm with Wilkinson shift takes 5 iterations to compute the eigenvalue $\lambda := -0.58579$.

Cubic convergence of tridiagonal QR algorithm

Let A_m be the result of m steps of QR iteration with Wilkinson shift applied to A. The cubic rate of convergence $\Rightarrow$ the subdiagonal entry $b_{n-1}^{(m)}$ of A_m converges to zero cubically, that is, $|b_{n-1}^{(m)}| = \mathcal{O}|b_{n-1}^{(m-1)}|^3)$.

Thus $a_n^{(m)}$ converges to an eigenvalue of A and in each iteration the number of correct digits is increased by a multiple of 3.

Example: Let $A := \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix}$. The eigenvalues are given by $\Lambda(A) := \{-3.4142, -2.0000, -0.58579\}$. The QR algorithm with Wilkinson shift takes 5 iterations to compute the eigenvalue $\lambda := -0.58579$.

The cubic rate of convergence of $b_{n-1}^{(m)}$ to zero and the convergence of $a_n^{(m)}$ to λ as the iteration progresses is evident from Table 1.

Example

```
A(n,n-1)       A(n,n)     |A(n,n)-lam|
-------- --------- ------------------
1.0000e+00  -2.0000e+00  -1.4142e+00
7.0711e-01  -1.0000e+00  -4.1421e-01
3.0397e-02  -5.8642e-01  -6.2863e-04
4.4798e-07  -5.8579e-01  -1.4044e-13
4.4225e-22  -5.8579e-01    0

Table 1:  Convergence rate
```

Example

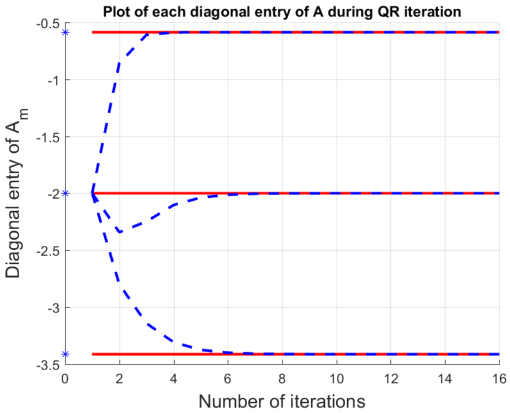
A(n,n-1)	A(n,n)	|A(n,n)-lam|
1.0000e+00	-2.0000e+00	-1.4142e+00
7.0711e-01	-1.0000e+00	-4.1421e-01
3.0397e-02	-5.8642e-01	-6.2863e-04
4.4798e-07	-5.8579e-01	-1.4044e-13
4.4225e-22	-5.8579e-01	0

Table 1: Convergence rate

After 5 steps of QR iteratioin, the matrix A_5 is given by

$$A_5 := \begin{bmatrix} -3.4142 & 1.59 \times 10^{-3} & 0 \\ 1.59 \times 10^{-3} & -2.0 & 0 \\ 0 & 0 & -0.58579 \end{bmatrix}.$$

Example



The blue stars are the eigenvalues of A and the red lines are the plots of the eigenvalues of A against the iteration numbers. The dotted curves are the plots of the diagonal entries of A_m (QR algorithm without shift).