QR-decomposition

The **QR-decomposition** of an $n \times k$ matrix $\mathbf{A}$, $k \leq n$, is an $n \times n$ unitary matrix $\mathbf{Q}$ and an $n \times k$ upper triangular matrix $\mathbf{R}$ for which

$$\mathbf{A} = \mathbf{QR}$$

In **Matlab**

```
[Q,R]=qr(A);
```

Note. The QR-decomposition is unique up to a change of signs of the columns of $\mathbf{Q}$:

$$\mathbf{A} = (\mathbf{QD})(\bar{\mathbf{D}}\mathbf{R})$$

with $|\mathbf{D}| = \mathbf{I}$

QR-decomposition

The **QR-decomposition** of an $n \times k$ matrix **A**, $k \leq n$, is an $n \times n$ unitary matrix **Q** and an $n \times k$ upper triangular matrix **R** for which

$$\mathbf{A} = \mathbf{QR}$$

If **A** is $n \times k$ with column rank ℓ and $\ell \leq k \leq n$, then the **'ecomical'** QR-decomposition is an $n \times \ell$ orthonormal matrix **Q** and an $\ell \times k$ upper triangular matrix R for which

$$\mathbf{A} = \mathbf{Q}R$$

In **Matlab**

```
[Q,R]=qr(A,'0');
```

QR-decomposition

The **QR-decomposition** of an $n \times k$ matrix $\mathbf{A}$, $k \leq n$, is an $n \times n$ unitary matrix $\mathbf{Q}$ and an $n \times k$ upper triangular matrix $\mathbf{R}$ for which

$$\mathbf{A} = \mathbf{QR}$$

If $\mathbf{A}$ is $n \times k$ with column rank ℓ and $\ell \leq k \leq n$, then the **'ecomical'** QR-decomposition is an $n \times \ell$ orthonormal matrix $\mathbf{Q}$ and an $\ell \times k$ upper triangular matrix R for which

$$\mathbf{A} = \mathbf{Q}R$$

Note. The columns of $\mathbf{Q}$ form an orthonormal basis of the space spanned by the columns of $\mathbf{A}$: the QR-decomp. represents the results of the Gram-Schmidt process.

QR-decomposition

The **QR-decomposition** of an $n \times k$ matrix $\mathbf{A}$, $k \leq n$, is an $n \times n$ unitary matrix $\mathbf{Q}$ and an $n \times k$ upper triangular matrix $\mathbf{R}$ for which

$$\mathbf{A} = \mathbf{QR}$$

Theorem. The QR-decomposition can be stably computed with Householder reflections.

QR-decomposition

The **QR-decomposition** of an $n \times k$ matrix $\mathbf{A}$, $k \leq n$, is an $n \times n$ unitary matrix $\mathbf{Q}$ and an $n \times k$ upper triangular matrix $\mathbf{R}$ for which

$$\mathbf{A} = \mathbf{Q}\mathbf{R}$$

Theorem. The QR-decomposition can be stably computed with Householder reflections:

Let $\widetilde{\mathbf{R}}$ be the computed $\mathbf{R}$ and $\mathbf{Q} = (\mathbf{H}_{v_k} \cdot \ldots \cdot \mathbf{H}_{v_1})^*$ with $\mathbf{H}_{v_j}$ the Householder reflection as actually used in step j. Then $\mathbf{A} + \Delta_A = \mathbf{Q}\widetilde{\mathbf{R}}$ for some Δ_A with $\|\Delta_A\|_{\mathsf{F}} \leq nk\mathbf{u}\|\mathbf{A}\|_{\mathsf{F}}$.

Note. The claim is **not** that $\mathbf{Q}$ is close to the $\mathbf{Q}$ that we would have been obtained in exact arithmetic, but that $\mathbf{Q}$ is unitary (product of Householder reflections).

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is

an $n \times n$ unitary matrix $\mathbf{U}$ and

an $n \times u$ upper triangular matrix $\mathbf{S}$

such that

$$\mathbf{A} = \mathbf{USU}^* \quad \text{or, equivalently,} \quad \mathbf{AU} = \mathbf{US}$$

In **Matlab**

```
[U,S]=schur(A);
```

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is

an $n \times n$ unitary matrix $\mathbf{U}$ and

an $n \times u$ upper triangular matrix $\mathbf{S}$

such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

Theorem. If $\mathbf{S}\mathbf{T} = \mathbf{T}\Lambda$ is the eigenvalue decomposition of $\mathbf{S}$, i.e., $\mathbf{T}$ is non-singular and Λ is diagonal, then $\mathbf{A}(\mathbf{U}\mathbf{T}) = (\mathbf{U}\mathbf{T})\Lambda$ is the eigenvalue decomposition of $\mathbf{A}$.

In particular, $\Lambda(\mathbf{A}) = \Lambda(\mathbf{S}) = \text{diag}(\mathbf{S})$ and $\mathcal{C}_2(\mathbf{T}) = \mathcal{C}_2(\mathbf{U}\mathbf{T})$.

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is

an $n \times n$ unitary matrix $\mathbf{U}$ and

an $n \times u$ upper triangular matrix $\mathbf{S}$

such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

The columns $\mathbf{u}_j \equiv \mathbf{U}\mathbf{e}_j$ of $\mathbf{U}$ are called **Schur vectors**.

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is
an $n \times n$ unitary matrix $\mathbf{U}$ and
an $n \times u$ upper triangular matrix $\mathbf{S}$
such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

Observation. Many techniques, where the eigenvalue decomposition of $\mathbf{A}$ is exploited, can be based on the Schur decomposition as well. For practical computations, the Schur decomposition is preferable, since it is stable: $\mathcal{C}_2(\mathbf{U}) = 1$, while $\mathcal{C}_2(\mathbf{U}\mathbf{T})$ can be huge.

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix **A** is
an $n \times n$ unitary matrix **U** and
an $n \times u$ upper triangular matrix **S**
such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

Note. The first column $\mathbf{u}_1 \equiv \mathbf{U}\mathbf{e}_1$ of **U** is an eigenvector of **A** with eigenvalue $\lambda_1 \equiv \mathbf{e}_1^*\mathbf{S}\mathbf{e}_1$.

Proof. **S** is upper triangular: $\mathbf{S}\mathbf{e}_1 = \lambda_1 \mathbf{e}_1$.

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is

an $n \times n$ unitary matrix $\mathbf{U}$ and

an $n \times u$ upper triangular matrix $\mathbf{S}$

such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

Note. The last column $\mathbf{u}_n \equiv \mathbf{U}\mathbf{e}_n$ of $\mathbf{U}$ is an eigenvector of $\mathbf{A}^*$ with eigenvalue $\overline{\lambda}_n$, where $\lambda_n \equiv \mathbf{e}_n^* \mathbf{S} \mathbf{e}_n$.

Proof. $\mathbf{S}^*$ is lower triangular: $\mathbf{S}^* \mathbf{e}_n = \overline{\lambda}_n \mathbf{e}_n$.

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is

an $n \times n$ unitary matrix $\mathbf{U}$ and

an $n \times u$ upper triangular matrix $\mathbf{S}$

such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

Note. The second column $\mathbf{u}_2$ of $\mathbf{U}$ is an eigenvector of $\mathbf{A}' \equiv (\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^*)\mathbf{A}(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^*)$ with eigenvalue $\lambda_2 \equiv \mathbf{e}_2^*\mathbf{S}\mathbf{e}_2$.

Proof. $\mathbf{S}$ is upper triangular: $\mathbf{S}\mathbf{e}_2 = \alpha\mathbf{e}_1 + \lambda_1\mathbf{e}_2$ for $\alpha = \mathbf{S}_{1,2}$. Hence, $\mathbf{U}\mathbf{S}\mathbf{U}^*\mathbf{u}_2 = \alpha\mathbf{u}_1 + \lambda_1\mathbf{u}_2$.

Schur decomposition

The **Schur decomposition** or of an $n \times n$ matrix $\mathbf{A}$ is
an $n \times n$ unitary matrix $\mathbf{U}$ and
an $n \times u$ upper triangular matrix $\mathbf{S}$
such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^* \quad \text{or, equivalently,} \quad \mathbf{A}\mathbf{U} = \mathbf{U}\mathbf{S}$$

In **Matlab**

```
[U,S]=schur(A);
```

Note. The second column $\mathbf{u}_2$ of $\mathbf{U}$ is an eigenvector of $\mathbf{A}' \equiv (\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^*)\mathbf{A}(\mathbf{I} - \mathbf{u}_1\mathbf{u}_1^*)$ with eigenvalue $\lambda_2 \equiv \mathbf{e}_2^*\mathbf{S}\mathbf{e}_2$.

In $\mathbf{A}'$, the eigenvector $\mathbf{u}_1$ is **deflated** from $\mathbf{A}$.

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} - σ_k I = Q_k R_k
    3) Compute   S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Theorem. With a proper shift strategy:
$\mathbf{U}_k \rightarrow \mathbf{U}$, $\quad \mathbf{U}$ is unitary
$\mathbf{S}_k \rightarrow \mathbf{S}$, $\quad \mathbf{S}$ is upper triangular, $\quad \mathbf{AU} = \mathbf{US}$:

The QR-algorithm converges to the Schur decomposition.

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, ... do
   1) Select a shift σ_k
   2) Compute Q_k unitary and R_k upper triangular
      such that S_{k-1} − σ_k I = Q_k R_k
   3) Compute  S_k = R_k Q_k + σ_k I
   4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

Proof.

$$\begin{aligned}\mathbf{S}_0\mathbf{Q}_1 &= (\mathbf{S}_0 - \sigma_1\mathbf{I} + \sigma_1\mathbf{I})\mathbf{Q}_1 = (\mathbf{Q}_1\mathbf{R}_1 + \sigma_1\mathbf{I})\mathbf{Q}_1 \\ &= \mathbf{Q}_1(\mathbf{R}_1\mathbf{Q}_1 + \sigma_1\mathbf{I}) = \mathbf{Q}_1\mathbf{S}_1\end{aligned}$$

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute   S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

Proof. $$\mathbf{S}_{k-1}\mathbf{Q}_k = \mathbf{Q}_k\mathbf{S}_k$$

$$\mathbf{A}\mathbf{U}_k = \mathbf{S}_0\mathbf{Q}_1\mathbf{Q}_2\ldots\mathbf{Q}_k = \mathbf{Q}_1\mathbf{S}_1\mathbf{Q}_2\ldots\mathbf{Q}_k = \mathbf{U}_k\mathbf{S}_k$$

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
   1) Select a shift σ_k
   2) Compute Q_k unitary and R_k upper triangular
      such that S_{k-1} − σ_k I = Q_k R_k
   3) Compute   S_k = R_k Q_k + σ_k I
   4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$,

Proof. $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_{k-1}(\mathbf{S}_{k-1} - \sigma_k\mathbf{I}) = \mathbf{U}_{k-1}\mathbf{Q}_k\mathbf{R}_k$

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute   S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Proof. $\mathbf{U}_{k-1}^*(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I}) = \mathbf{R}_k^*\mathbf{U}_k^*$

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , ... do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute  S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Corollary. With $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_1$ and $\tau_k \equiv \mathbf{e}_1^*\mathbf{R}_k\mathbf{e}_1$,
we have $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{x}_{k-1} = \tau_k\,\mathbf{x}_k$ (the shifted power method).

Proof. $\mathbf{R}_k$ upper triangular $\Rightarrow$ $\mathbf{R}_k\mathbf{e}_1 = \tau_k\mathbf{e}_1$.

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute  S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Corollary. With $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_1$ and $\tau_k \equiv \mathbf{e}_1^*\mathbf{R}_k\mathbf{e}_1$,
we have $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{x}_{k-1} = \tau_k\,\mathbf{x}_k$ (the shifted power method).

With $p(\lambda) \equiv (\lambda - \sigma_k) \cdot \ldots \cdot (\lambda - \sigma_1)$,

$$\mathbf{x}_k = \tau p(\mathbf{A})\mathbf{x}_0 \text{ for some } \tau \in \mathbb{C}.$$

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute   S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Corollary. With $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_1$ and $\tau_k \equiv \mathbf{e}_1^*\mathbf{R}_k\mathbf{e}_1$,
we have $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{x}_{k-1} = \tau_k\,\mathbf{x}_k$ (the shifted power method).

Note that $\lambda^{(k)} \equiv \mathbf{x}_k^*\mathbf{A}\mathbf{x}_k = \mathbf{e}_1^*\mathbf{U}_k^*\mathbf{A}\mathbf{U}_k\mathbf{e}_1 = \mathbf{e}_1\mathbf{S}_k\mathbf{e}_1$.

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute  S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Suppose $\mathbf{v}$ is the dominant eigenvector for $\mathbf{A} - \sigma\mathbf{I}$.
With $\sigma_k = \sigma$ and $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_1$, for $k \to \infty$, we have that

$$\angle(\mathbf{x}_k, \mathbf{v}) \to 0, \quad \lambda^{(k)} \equiv \mathbf{e}_1^*\mathbf{S}_k\mathbf{e}_1 \to \lambda, \quad \mathbf{S}_k\mathbf{e}_1 - \lambda^{(k)}\mathbf{e}_1 \to 0,$$

where λ is the eigenvalue of $\mathbf{A}$ associated $\mathbf{v}$.

QR-algorithm

```
Select U_0 unitary.   Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
   1) Select a shift σ_k
   2) Compute Q_k unitary and R_k upper triangular
      such that S_{k-1} - σ_k I = Q_k R_k
   3) Compute   S_k = R_k Q_k + σ_k I
   4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Corollary. With $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_n$ and $\tau_k \equiv \mathbf{e}_n^*\mathbf{R}_k\mathbf{e}_n$,
we have $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{x}_k = \bar{\tau}_k\,\mathbf{x}_{k-1}$ (Shift & Invert).

Proof. $\mathbf{R}_k^*$ lower triangular $\Rightarrow$ $\mathbf{R}_k^*\mathbf{e}_n = \bar{\tau}_k\mathbf{e}_n$.

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute   S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Corollary. With $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_n$ and $\tau_k \equiv \mathbf{e}_n^*\mathbf{R}_k\mathbf{e}_n$,
we have $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{x}_k = \bar{\tau}_k\,\mathbf{x}_{k-1}$ (Shift & Invert).

Note that $\lambda^{(k)} \equiv \mathbf{x}_k^*\mathbf{A}\mathbf{x}_k = \mathbf{e}_n^*\mathbf{U}_k^*\mathbf{A}\mathbf{U}_k\mathbf{e}_n = \mathbf{e}_n\mathbf{S}_k\mathbf{e}_n$.

QR-algorithm

```
Select U_0 unitary.  Compute S_0 = U_0^* A U_0
for k = 1, 2, , . . . do
    1) Select a shift σ_k
    2) Compute Q_k unitary and R_k upper triangular
       such that S_{k-1} − σ_k I = Q_k R_k
    3) Compute   S_k = R_k Q_k + σ_k I
    4) U_k = U_{k-1} Q_k
end for
```

Lemma. a) $\mathbf{U}_k$ unitary, b) $\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{S}_k$.

c) $(\mathbf{A} - \sigma_k\mathbf{I})\mathbf{U}_{k-1} = \mathbf{U}_k\mathbf{R}_k$, d) $(\mathbf{A}^* - \bar{\sigma}_k\mathbf{I})\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{R}_k^*$.

Suppose $\mathbf{v}$ is the dominant eigenvector for $(\mathbf{A}^* - \bar{\sigma}\mathbf{I})^{-1}$. With $\sigma_k = \sigma$ and $\mathbf{x}_k \equiv \mathbf{U}_k\mathbf{e}_n$, for $k \to \infty$, we have that

$$\angle(\mathbf{x}_k, \mathbf{v}) \to 0, \quad \lambda^{(k)} \equiv \mathbf{e}_n^*\mathbf{S}_k\mathbf{e}_n \to \lambda, \quad \mathbf{S}_k^*\mathbf{e}_n - \bar{\lambda}^{(k)}\mathbf{e}_n \to 0,$$

where $\bar{\lambda}$ is the eigenvalue of $\mathbf{A}^*$ associated $\mathbf{v}$.

Selecting shifts

The QR-agorithm incorporates the Shift and Invert power method (for $\mathbf{A}^*$).

Rayleigh Quotient Iteration is Shift and Invert with shifts equal to the the Rayleigh quotients, $\bar{\sigma}_k = \mathbf{x}_{k-1}^* \mathbf{A}^* \mathbf{x}_{k-1}$.

Theorem. The asymptotic convergence of RQI is quadratic.

In this case, with $\mathbf{x}_{k-1} = \mathbf{U}_{k-1}\mathbf{e}_n, \quad \sigma_k = \mathbf{e}_n^* \mathbf{S}_{k-1}\mathbf{e}_n$.

With *"The asymptotic convergence of this method is quadratic"*, we mean: the method produces sequences $(\mathbf{x}_k)$ that converge provided $\mathbf{x}_0$ is close enough to some (limit) eigenvector, and for k large, the error reduces quadratically.

Selecting shifts

The QR-agorithm incorporates the Shift and Invert power method (for $\mathbf{A}^*$).

Rayleigh Quotient Iteration is Shift and Invert with shifts equal to the the Rayleigh quotients, $\bar{\sigma}_k = \mathbf{x}_{k-1}^* \mathbf{A}^* \mathbf{x}_{k-1}$.

Theorem. The asymptotic convergence of RQI is quadratic.

In this case, with $\mathbf{x}_{k-1} = \mathbf{U}_{k-1}\mathbf{e}_n, \quad \sigma_k = \mathbf{e}_n^* \mathbf{S}_{k-1} \mathbf{e}_n$.

RQI need converge, as the following example shows

Example. With $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\mathbf{x}_0 = \mathbf{e}_1$.

RQI produces the sequence $(\mathbf{x}_k) = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_1, \mathbf{e}_2, \ldots)$.
Note that $\mathbf{x}_k^* \mathbf{A} \mathbf{x}_k = 0$.

Observation.
The shifts $\sigma_k = \mathbf{e}_n^* \mathbf{S}_{k-1} \mathbf{e}_n$ may lead to stagnation.

Selecting shifts

The QR-agorithm incorporates the Shift and Invert power method (for $\mathbf{A}^*$).

The Wilkinson shift is the absolute smallest eigenvalue of the 2×2 right lower block of $\mathbf{S}_k$.

Selecting shifts

The QR-agorithm incorporates the Shift and Invert power method (for $\mathbf{A}^*$).

The Wilkinson shift is the absolute smallest eigenvalue of the 2×2 right lower block of $\mathbf{S}_k$.

Theorem.
For σ_k take the Wilkinson shift and take $\mathbf{x}_k \equiv \mathbf{U}_k \mathbf{e}_n$.
Then, for some eigenpair $(\mathbf{v}, \bar{\lambda})$ of $\mathbf{A}^*$, we have that

$$\angle(\mathbf{x}_k, \mathbf{v}) \to 0, \quad \lambda^{(k)} \equiv \mathbf{e}_n^* \mathbf{S}_k \mathbf{e}_n \to \lambda, \quad \mathbf{S}_k^* \mathbf{e}_n - \bar{\lambda}^{(k)} \mathbf{e}_n \to 0.$$

The convergence is quadratic (and cubic if $\mathbf{A}$ is Hermitian).

Selecting shifts

The QR-agorithm incorporates the Shift and Invert power method (for $\mathbf{A}^*$).

The Wilkinson shift is the absolute smallest eigenvalue of the 2×2 right lower block of $\mathbf{S}_k$.

Theorem.
For σ_k take the Wilkinson shift and take $\mathbf{x}_k \equiv \mathbf{U}_k \mathbf{e}_n$.
Then, for some eigenpair $(\mathbf{v}, \bar{\lambda})$ of $\mathbf{A}^*$, we have that

$$\angle(\mathbf{x}_k, \mathbf{v}) \to 0, \quad \lambda^{(k)} \equiv \mathbf{e}_n^* \mathbf{S}_k \mathbf{e}_n \to \lambda, \quad \mathbf{S}_k^* \mathbf{e}_n - \bar{\lambda}^{(k)} \mathbf{e}_n \to 0.$$

The convergence is quadratic (and cubic if $\mathbf{A}$ is Hermitian).

The QR-algorithm: if $\|\mathbf{e}_n^* \mathbf{S}_k - \lambda^{(k)} \mathbf{e}_n^*\|_2 \leq \epsilon$, then

- accept $\mathbf{U}_k \mathbf{e}_n$ as an eigenvector of $\mathbf{A}^*$
- **deflate**: delete the last row and column of $\mathbf{S}_k$ and continu (the search for an eigenpair of the lower dimensional matrix).

Selecting shifts

The QR-agorithm incorporates the Shift and Invert power method (for $\mathbf{A}^*$).

The Wilkinson shift is the absolute smallest eigenvalue of the 2×2 right lower block of $\mathbf{S}_k$.

Theorem.
For σ_k take the Wilkinson shift and take $\mathbf{x}_k \equiv \mathbf{U}_k \mathbf{e}_n$.
Then, for some eigenpair $(\mathbf{v}, \bar{\lambda})$ of $\mathbf{A}^*$, we have that

$$\angle(\mathbf{x}_k, \mathbf{v}) \to 0, \quad \lambda^{(k)} \equiv \mathbf{e}_n^* \mathbf{S}_k \mathbf{e}_n \to \lambda, \quad \mathbf{S}_k^* \mathbf{e}_n - \bar{\lambda}^{(k)} \mathbf{e}_n \to 0.$$

The convergence is quadratic (and cubic if $\mathbf{A}$ is Hermitian).

The QR-algorithm: if $\|\mathbf{e}_n^* \mathbf{S}_k - \lambda^{(k)} \mathbf{e}_n^*\|_2 \leq \epsilon$, then

- accept $\mathbf{U}_k \mathbf{e}_n$ as the nth Schur vector of $\mathbf{A}$
- **deflate**: delete the last row and column of $\mathbf{S}_k$ and continu (the search for an eigenpair of the lower dimensional matrix).

Deflation

Consider the kth step of the QR-algorithm.

Put $\mathbf{u}_n \equiv \mathbf{U}_k \mathbf{e}_n$.
Note that

$$(\mathbf{I} - \mathbf{u}_n \mathbf{u}_n^*)\mathbf{U}_k = \mathbf{U}_k(\mathbf{I} - \mathbf{e}_n \mathbf{e}_n^*)$$

Hence

$$(\mathbf{I} - \mathbf{u}_n \mathbf{u}_n^*)\mathbf{A}(\mathbf{I} - \mathbf{u}_n \mathbf{u}_n^*)\mathbf{U}_k = \mathbf{U}_k(\mathbf{I} - \mathbf{e}_n \mathbf{e}_n^*)\mathbf{S}_k(\mathbf{I} - \mathbf{e}_n \mathbf{e}_n^*).$$

Deflating the nth Schur vector from $\mathbf{A}$ can easily be performed in the QR-algorithm: simply delete the last row and last column of the "active" matrix $\mathbf{S}_k$ (assumming that $\mathbf{U}_k \mathbf{e}_n$ is the nth Schur vector to required accuracy).

QR-algorithm

```
Select U unitary.   S = U*AU,
m = size(A,1), N = [1:m], I = I_m.

repeat until m = 1
   1) Select the Wilskinson shift σ
   2) [Q,R] = qr(S − σI)
   3) S ← RQ + σI
   4) U(:,N) ← U(:,N)Q
   5) if |S(m,m−1)| ≤ ε|S(m,m)|
         %% Deflate
         m ← m−1, N ← [1:m], I ← I_m
         S ← S(N,N)
      end if
end repeat
```

Theorem. $\mathbf{U}_k \to \mathbf{U}$, $\quad \mathbf{U}$ is unitary
$\mathbf{S}_k \to \mathbf{S}$, $\quad \mathbf{S}$ is upper triangular, $\quad \mathbf{AU} = \mathbf{US}$.

Observations.

• The QR-algorithm quickly converges towards to the eigenvalue as 'targeted' by the Wilkinson shift (on average 8 steps of the QR algorithm seems to be required for accurate detection of the first eigenvalue).

• While converging to a 'target' eigenvalue, other eigenvalues are also approximated. Therefore, the next eigenvalues are detected more quickly (from the 5th eigenvalue on, 2 steps appear to be sufficient).

• All eigenvalues are being computed (according to multiplicity). Computation of all eigenvalues (actually of the Schur decomposition of $\mathbf{A}$) requires approximately

$$2n \text{ steps of the QR-algorithm.}$$

• The order in which the eigenvalues are being computed can not be controled.

Initiation of the QR-algorithm

Theorem. There is a unitary matrix $\mathbf{U}_0$ (product of Householder reflections) such that

$$\mathbf{S}_0 \equiv \mathbf{U}_0^* \mathbf{A} \mathbf{U}_0$$

is upper Hessenberg.

Start QR-alg. Bring $\mathbf{A}$ to upper Hessenberg form (i.e., $\mathbf{S} = \mathbf{S}_0$, $\mathbf{U} = \mathbf{U}_0$). Computation requires $\frac{4}{3}n^3$ flop.

Theorem. If $\mathbf{S}_{k-1}$ is upper Hessenberg, then $\mathbf{S}_k$ is upper Hessenberg. Moreover, if $\mathbf{S}_k$ is $m \times m$, then $\mathbf{Q}_k$ can be obtained as a product of $m-1$ Givens rotations, i.e., rotations in the $(j, j+1)$ plane. The steps 2 and 3 in the QR algorithm can be performed in (together) $3m^2$ flop.

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations: with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$, where $\mathbf{G}_j$ rotates in the $(j, j{+}1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{R}_1 = \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} = \begin{bmatrix} c & -s & & & \\ s & c & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} = \mathbf{G}_1 \mathbf{R}_0$$

Here, $\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \cos(\phi_1) & -\sin(\phi_1) \\ \sin(\phi_1) & \cos(\phi_1) \end{bmatrix}$. Empty matrix entries are 0.

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j\,\mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{R}_2 = \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & c & -s & & \\ & s & c & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} = \mathbf{G}_2\mathbf{R}_1$$

Here, $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \cos(\phi_2) & -\sin(\phi_2) \\ \sin(\phi_2) & \cos(\phi_2) \end{bmatrix}$$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma \mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations: with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$, where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{R}_3 = \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \\ & & & \star & \star \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c & -s & \\ & & s & c & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} = \mathbf{G}_3 \mathbf{R}_2$$

Here, $\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \cos(\phi_3) & -\sin(\phi_3) \\ \sin(\phi_3) & \cos(\phi_3) \end{bmatrix}$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations: with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$, where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{R}_4 = \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \\ & & & & \star \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c & -s \\ & & & s & c \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \\ & & & \star & \star \end{bmatrix} = \mathbf{G}_4 \mathbf{R}_3$$

Here, $\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \cos(\phi_4) & -\sin(\phi_4) \\ \sin(\phi_4) & \cos(\phi_4) \end{bmatrix}$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations: with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$, where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{R} = \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \\ & & & & \star \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c & -s \\ & & & s & c \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \\ & & & \star & \star \end{bmatrix} = \mathbf{G}_4 \mathbf{R}_3$$

Here, $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \cos(\phi_4) & -\sin(\phi_4) \\ \sin(\phi_4) & \cos(\phi_4) \end{bmatrix}$$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

Then $\quad \mathbf{R} = \mathbf{R}_{n-1} \quad$ and $\quad \mathbf{Q}^* = \mathbf{G}_{n-1} \cdot \ldots \cdot \mathbf{G}_1$.

Note. $\mathbf{Q}$ need not be formed explicitly:
it suffices to store the sequences of cosines and sines.

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

Then $\quad \mathbf{R} = \mathbf{R}_{n-1} \quad$ and $\quad \mathbf{Q}^* = \mathbf{G}_{n-1} \cdot \ldots \cdot \mathbf{G}_1$,

and

$$\widetilde{\mathbf{S}} = \mathbf{R}\mathbf{G}_1 \cdot \ldots \cdot \mathbf{G}_{n-1}.$$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$(\mathbf{G}_3\ \mathbf{R}_2)\mathbf{G}_1 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c & -s & \\ & & s & c & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} \begin{bmatrix} c & -s & & & \\ s & c & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{G}_3(\mathbf{R}_2\ \mathbf{G}_1) = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c & -s & \\ & & s & c & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix} \begin{bmatrix} c & -s & & & \\ s & c & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

$$\mathbf{G}_3(\mathbf{R}_2\ \mathbf{G}_1) = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c & -s & \\ & & s & c & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ & & \star & \star & \star \\ & & \star & \star & \star \\ & & & \star & \star \end{bmatrix}$$

Chasing the bulge.

Upper Hessenberg matrices

Theorem. If $\mathbf{S}$ is an upper Hessenberg matrix and $\mathbf{S} = \mathbf{QR}$ is the QR-decomposition, then

- $\mathbf{Q}$ is Hessenberg
- $\widetilde{\mathbf{S}} \equiv \mathbf{RQ}$ and $\widetilde{\mathbf{S}} + \sigma\mathbf{I}$ are upper Hessenberg.

$\mathbf{Q}$ can be obtained as the product of $n-1$ Givens rotations:
with $\mathbf{R}_0 \equiv \mathbf{S}$, $\quad \mathbf{R}_j = \mathbf{G}_j \mathbf{R}_{j-1} \quad (j = 1, \ldots, n-1)$,
where $\mathbf{G}_j$ rotates in the $(j, j+1)$ plane (i.e., in span(e_j, e_{j+1}))

Then $\quad \mathbf{R} = \mathbf{R}_{n-1} \quad$ and $\quad \mathbf{Q}^* = \mathbf{G}_{n-1} \cdot \ldots \cdot \mathbf{G}_1$

and

$$\widetilde{\mathbf{S}} = \mathbf{R}\mathbf{G}_1 \cdot \ldots \cdot \mathbf{G}_{n-1}.$$

Property. $\quad \widetilde{\mathbf{S}} = \ldots \mathbf{G}_4^*(\mathbf{G}_3^*(\mathbf{G}_2^*\mathbf{G}_1^*\mathbf{S})\mathbf{G}_1)\mathbf{G}_2 \ldots$

Only two sines and two cosines have to be stored.

Initiation of the QR-algorithm

Theorem. There is a unitary matrix $\mathbf{U}_0$ (product of Householder reflections) such that

$$\mathbf{S}_0 \equiv \mathbf{U}_0^* \mathbf{A} \mathbf{U}_0$$

is upper Hessenberg.

Start QR-alg. Bring $\mathbf{A}$ to upper Hessenberg form (i.e., $\mathbf{S} = \mathbf{S}_0$, $\mathbf{U} = \mathbf{U}_0$). Computation requires $\frac{4}{3}n^3$ flop.

Theorem. If $\mathbf{S}_{k-1}$ is upper Hessenberg, then $\mathbf{S}_k$ is upper Hessenberg. Moreover, if $\mathbf{S}_k$ is $m \times m$, then $\mathbf{Q}_k$ can be obtained as a product of $m-1$ Givens rotations, i.e., rotations in the $(j, j+1)$ plane. The steps 2 and 3 in the QR algorithm can be performed in (together) $3m^2$ flop.

Initiation of the QR-algorithm

Theorem. There is a unitary matrix $\mathbf{U}_0$ (product of Householder reflections) such that

$$\mathbf{S}_0 \equiv \mathbf{U}_0^* \mathbf{A} \mathbf{U}_0$$

is upper Hessenberg.

Start QR-alg. Bring $\mathbf{A}$ to upper Hessenberg form (i.e., $\mathbf{S} = \mathbf{S}_0$, $\mathbf{U} = \mathbf{U}_0$). Computation requires $\frac{4}{3}n^3$ flop.

Theorem. If $\mathbf{S}_{k-1}$ is upper Hessenberg, then $\mathbf{S}_k$ is upper Hessenberg. Moreover, if $\mathbf{S}_k$ is $m \times m$, then $\mathbf{Q}_k$ can be obtained as a product of $m-1$ Givens rotations, i.e., rotations in the $(j, j+1)$ plane. The steps 2 and 3 in the QR algorithm can be performed in (together) $3m^2$ flop.

Observation. For computing the eigenvalues only, step 4 can be skipped.

Initiation of the QR-algorithm

Theorem. There is a unitary matrix $\mathbf{U}_0$ (product of Householder reflections) such that

$$\mathbf{S}_0 \equiv \mathbf{U}_0^* \mathbf{A} \mathbf{U}_0$$

is upper Hessenberg.

Start QR-alg. Bring $\mathbf{A}$ to upper Hessenberg form (i.e., $\mathbf{S} = \mathbf{S}_0$, $\mathbf{U} = \mathbf{U}_0$). Computation requires $\frac{4}{3}n^3$ flop.

Theorem. If $\mathbf{S}_{k-1}$ is upper Hessenberg, then $\mathbf{S}_k$ is upper Hessenberg. Moreover, if $\mathbf{S}_k$ is $m \times m$, then $\mathbf{Q}_k$ can be obtained as a product of $m-1$ Givens rotations, i.e., rotations in the $(j, j+1)$ plane. The steps 2 and 3 in the QR algorithm can be performed in (together) $3m^2$ flop.

If the eigenvalue λ_j is available, then the associated eigenvector can also be computed with Shift & Invert: solve $(\mathbf{A} - \lambda_j \mathbf{I})\mathbf{v}_j = \mathbf{e}_1$ for $\mathbf{v}_j$. Note that the LU-decomposition can cheaply be computed if $\mathbf{A}$ is upper Hessenberg.

Initiation of the QR-algorithm

Theorem. There is a unitary matrix $\mathbf{U}_0$ (product of Householder reflections) such that

$$\mathbf{S}_0 \equiv \mathbf{U}_0^* \mathbf{A} \mathbf{U}_0$$

is upper Hessenberg.

Start QR-alg. Bring $\mathbf{A}$ to upper Hessenberg form (i.e., $\mathbf{S} = \mathbf{S}_0$, $\mathbf{U} = \mathbf{U}_0$). Computation requires $\frac{4}{3}n^3$ flop.

Theorem. If $\mathbf{S}_{k-1}$ is upper Hessenberg, then $\mathbf{S}_k$ is upper Hessenberg. Moreover, if $\mathbf{S}_k$ is $m \times m$, then $\mathbf{Q}_k$ can be obtained as a product of $m-1$ Givens rotations, i.e., rotations in the $(j, j+1)$ plane. The steps 2 and 3 in the QR algorithm can be performed in (together) $3m^2$ flop.

Observation. The QR-algorithm requires approximately

$$8n^3 \text{ flop}$$

to compute the Schur decomposition to full accuracy.

Initiation of the QR-algorithm

Theorem. There is a unitary matrix $\mathbf{U}_0$ (product of Householder reflections) such that

$$\mathbf{S}_0 \equiv \mathbf{U}_0^* \mathbf{A} \mathbf{U}_0$$

is upper Hessenberg.

Start QR-alg. Bring $\mathbf{A}$ to upper Hessenberg form (i.e., $\mathbf{S} = \mathbf{S}_0$, $\mathbf{U} = \mathbf{U}_0$). Computation requires $\frac{4}{3}n^3$ flop.

Theorem. If $\mathbf{S}_{k-1}$ is upper Hessenberg, then $\mathbf{S}_k$ is upper Hessenberg. Moreover, if $\mathbf{S}_k$ is $m \times m$, then $\mathbf{Q}_k$ can be obtained as a product of $m-1$ Givens rotations, i.e., rotations in the $(j, j+1)$ plane. The steps 2 and 3 in the QR algorithm can be performed in (together) $3m^2$ flop.

Observation. The QR-algorithm can not exploit any sparsity structure of $\mathbf{A}$.

Benefits of the QR-RQ steps.

• The Shift & Invert power method is implicitly incorporated for one eigenvalue.

• The power method is implicitly incorporated for all other eigenvalues.

• Easy deflation is allowed.

• The computations are stable (when a stable qr-decomposition is used).

When combined with an upper Hessenberg start:

• Upper Hessenberg structure is preserved, leading to relatively low computational costs per step.

• Simple error controle:
the norm of the residual equals $|\mathbf{S}_k(n, n-1)|$.

• Effective shifts can easily be computed: with an eigenvalue of the 2×2 right lower block of $\mathbf{S}_k$, quadratic convergence is achieved and stagnation avoided.

Excellent performance of the QR algorithm relies on

• **QR-RQ steps.** **(see previous transparant)**

• A good **shift strategy** leading to fast convergence (quadratic and, if $\mathbf{A}$ is Hermitian, cubic) to one eigenvalue. While quickly converging to one eigenvalue, other eigenvalues are also approximated, yielding good starts for quick eigenvalue computation.

• **Deflation** allows a fast search for the next eigenvalue. Deflation is performed simply by deleting the last row and the last column of the active matrix.

• The **upper Hessenberg** structure is preserved, allowing relatively cheap QR steps.

Theorem. The QR-algorithm is stable: for the matrix $\mathbf{U}$ and the upper triangular matrix $\mathbf{S}$ we have that

$$(\mathbf{A} + \Delta_A)\mathbf{U} = \mathbf{US}, \qquad \|\mathbf{U}^*\mathbf{U} - \mathbf{I}\|_2 \leq \mathbf{u}s$$

where Δ_A is an $n \times n$ matrix such that

$$\|\Delta_A\|_2 \leq \mathbf{u}\|\mathbf{A}\|_2$$