QR-Iteration mit Hessenbergmatrizen

$$A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

Gegeben sei eine $n \times n$-Hessenbergmatrix A.

Dann lässt sich mit Hilfe von Givens-Rotationen eine QR-Iteration in nur $\mathcal{O}(n^2)$ flops ausführen.

QR-Iteration mit Hessenbergmatrizen

$$A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

Dazu betrachte die **rote** 2×2-Submatrix und bestimme eine Givensrotation $\hat{G}_1$, die das $(2,1)$-Element eliminiert:

$$\hat{G}_1 \begin{bmatrix} * & * \\ * & * \end{bmatrix} = \begin{bmatrix} * & * \\ 0 & * \end{bmatrix}$$

QR-Iteration mit Hessenbergmatrizen

$$G_1 A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

Bette $\hat{G}_1$ in eine $n \times n$-Matrix ein:

$$G_1 = \begin{bmatrix} \hat{G}_1 & 0 \\ 0 & I_{n-2} \end{bmatrix}$$

Multiplikation mit G_1 beeinflusst nur den **roten** Teil der Matrix.

QR-Iteration mit Hessenbergmatrizen

$$G_1 A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & {\color{red}*} & {\color{red}*} & * & * & * \\ 0 & {\color{red}*} & {\color{red}*} & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

und so weiter...

$$G_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & {\color{red}\hat{G}_2} & 0 \\ 0 & 0 & I_{n-3} \end{bmatrix}$$

mit einer Givens-Rotation $\hat{G}_2$, die den $(2, 1)$-Eintrag der **roten** Matrix eliminiert.

QR-Iteration mit Hessenbergmatrizen

$$G_2 G_1 A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$G_3 G_2 G_1 A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$G_4 G_3 G_2 G_1 A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$G_5 G_4 G_3 G_2 G_1 A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$\underbrace{G_5 G_4 G_3 G_2 G_1}_{Q^*} A = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

Damit haben wir eine QR-Zerlegung von A berechnet mit

$$Q = G_1^{-1} \cdots G_{n-1}^{-1}, \quad R = Q^* A.$$

Jetzt berechnen wir das Produkt RQ, d.h. wir bilden das Produkt

$$R G_1^{-1} \cdots G_{n-1}^{-1} = R G_1^* \cdots G_{n-1}^*$$

QR-Iteration mit Hessenbergmatrizen

$$R = \begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

Multiplikation mit

$$G_1^* = \begin{bmatrix} \hat{G_1}^* & 0 \\ 0 & I_{n-2} \end{bmatrix}$$

von rechts beeinflusst nur den **roten** Teil der Matrix.

QR-Iteration mit Hessenbergmatrizen

$$RG_1^* = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

QR-Iteration mit Hessenbergmatrizen

$$RG_1^* = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

und so weiter...

Multiplikation mit G_2^* von rechts beeinflusst nur **diesen roten** Bereich der Matrix.

QR-Iteration mit Hessenbergmatrizen

$$RG_1^*G_2^* = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$RG_1^*G_2^*G_3^* = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$RG_1^*G_2^*G_3^*G_4^* = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

und so weiter...

QR-Iteration mit Hessenbergmatrizen

$$RG_1^*G_2^*G_3^*G_4^*G_5^* = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

FERTIG!

Fazit: Eine QR-Iteration kostet nur $\mathcal{O}(n^2)$ flops und die resultierende Matrix ist wieder in Hessenbergform.