Householder Reduction (III)

- After $n-1$ Householder Reflections,

$$H_{n-1} \cdots H_1 \, A \, H_1^T \cdots H_{n-1}^T = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix} \stackrel{def}{=} T.$$

- Householder's Method,

$$H^T \, A \, H = T, \quad \text{or} \quad A = H \, T \, H^T,$$

where $H \stackrel{def}{=} H_1^T \cdots H_{n-1}^T$.

Reduction cost?

Householder Reduction (IV)

Partition symmetric matrix $A = \begin{pmatrix} \alpha_1 & \mathbf{a}_1^T \\ \mathbf{a}_1 & \widehat{A}_1 \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad \widehat{A}_1^T = \widehat{A}_1.$

- Let $\mathbf{v}_1 \in \mathbb{R}^{n-1}$ be the unit vector in the Householder Reflection matrix $\widehat{H}_1 = I - 2\,\mathbf{v}_1\,\mathbf{v}_1^T$ so that

$$\widehat{H}_1\,\mathbf{a}_1 = \begin{pmatrix} \pm\,\|\mathbf{a}_1\|_2 \\ \mathbf{0} \end{pmatrix} \stackrel{def}{=} \begin{pmatrix} \beta_1 \\ \mathbf{0} \end{pmatrix}. \quad \text{Let} \quad H_1 = \begin{pmatrix} 1 & \\ & \widehat{H}_1 \end{pmatrix}.$$

- Similarity transform on A:

$$H_1\,A\,H_1^T \;=\; \begin{pmatrix} \alpha_1 & \left(\widehat{H}_1\,\mathbf{a}_1\right)^T \\ \widehat{H}_1\,\mathbf{a}_1 & \widehat{H}_1\,\widehat{A}_1\,\widehat{H}_1^T \end{pmatrix} \stackrel{def}{=} \begin{pmatrix} \alpha_1 & \beta_1 & \mathbf{0}^T \\ \beta_1 & \alpha_2 & \mathbf{a}_2^T \\ \mathbf{0} & \mathbf{a}_2 & \widehat{A}_2 \end{pmatrix}.$$

- Main cost of this reduction:

$$\widehat{H}_1\,\widehat{A}_1\,\widehat{H}_1^T = \left(I - 2\,\mathbf{v}_1\,\mathbf{v}_1^T\right)\,\widehat{A}_1\,\left(I - 2\,\mathbf{v}_1\,\mathbf{v}_1^T\right).$$

Householder Reduction (V)

- Main cost of first Householder reduction:

$$\begin{aligned}\widehat{H}_1\,\widehat{A}_1\,\widehat{H}_1^T &= \left(I - 2\,\mathbf{v}_1\,\mathbf{v}_1^T\right)\,\widehat{A}_1\,\left(I - 2\,\mathbf{v}_1\,\mathbf{v}_1^T\right)\\ &= \widehat{A}_1 - 2\,\mathbf{v}_1\,\mathbf{v}_1^T\,\widehat{A}_1 - 2\,\widehat{A}_1\,\mathbf{v}_1\,\mathbf{v}_1^T + 4\left(\mathbf{v}_1^T\,\widehat{A}_1\,\mathbf{v}_1\right)\mathbf{v}_1\,\mathbf{v}_1^T\\ &= \widehat{A}_1 - \mathbf{u}_1\,\mathbf{v}_1^T + \mathbf{v}_1\,\mathbf{w}_1^T,\end{aligned}$$

  with $\mathbf{u}_1 = 2\,\widehat{A}_1\,\mathbf{v}_1$, $\tau = 2\,\mathbf{u}_1^T\,\mathbf{v}_1$, and $\mathbf{w}_1 = \tau\,\mathbf{v}_1 - 2\,\mathbf{u}_1$.
- Only half of entries in $\widehat{H}_1\,\widehat{A}_1\,\widehat{H}_1^T$ need to be computed.
- This adds to about $4(n-1)^2$ multiplications and additions.

Householder Reduction (VI)

- $n-1$ Householder Reflections,

$$H_{n-1} \cdots H_1 \, A \, H_1^T \cdots H_{n-1}^T = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix}.$$

- For each $1 \leq j \leq n-1$, main cost of j-th Householder reduction

$$\widehat{H}_j \, \widehat{A}_j \, \widehat{H}_j^T = \left(I - 2\,\mathbf{v}_j\,\mathbf{v}_j^T\right) \widehat{A}_j \left(I - 2\,\mathbf{v}_j\,\mathbf{v}_j^T\right),$$

namely about $4(n-j)^2$ multiplications and additions.

- Grand total is about $\sum_{j=1}^{n-1} 4(n-j)^2 \approx \frac{4}{3}\, n^3$ multiplications and additions.

Non-symmetric Reduction (I)

Partition non-symmetric matrix $A = \begin{pmatrix} \alpha_1 & \mathbf{b}_1^T \\ \mathbf{a}_1 & \widehat{A}_1 \end{pmatrix} \in \mathbb{R}^{n \times n}$.

- Let $\mathbf{v}_1 \in \mathbb{R}^{n-1}$ be the unit vector in the Householder Reflection matrix $\widehat{H}_1 = I - 2\,\mathbf{v}_1\,\mathbf{v}_1^T$ so that

$$\widehat{H}_1\,\mathbf{a}_1 = \begin{pmatrix} \beta_1 \\ \mathbf{0} \end{pmatrix}, \quad \text{and} \quad H_1 = \begin{pmatrix} 1 & \\ & \widehat{H}_1 \end{pmatrix}.$$

- Similarity transform on A:

$$H_1\,A\,H_1^T \quad = \quad \begin{pmatrix} \alpha_1 & \left(\widehat{H}_1\,\mathbf{b}_1\right)^T \\ \widehat{H}_1\,\mathbf{a}_1 & \widehat{H}_1\,\widehat{A}_1\,\widehat{H}_1^T \end{pmatrix} \stackrel{def}{=} \begin{pmatrix} \alpha_1 & x & \mathbf{x}_1^T \\ \beta_1 & \alpha_2 & \mathbf{b}_2^T \\ \mathbf{0} & \mathbf{a}_2 & \widehat{A}_2 \end{pmatrix}.$$

- Repeat similarity transform procedure on $\mathbf{a}_2$ and $\widehat{A}_2$.

Non-symmetric Reduction (II)

- Let $\mathbf{v}_2 \in \mathbb{R}^{n-2}$ be the unit vector in the Householder Reflection matrix $\widehat{H}_2 = I - 2\,\mathbf{v}_2\,\mathbf{v}_2^T$ so that

$$\widehat{H}_2\,\mathbf{a}_2 = \begin{pmatrix} \pm\,\|\mathbf{a}_2\|_2 \\ \mathbf{0} \end{pmatrix} \stackrel{def}{=} \begin{pmatrix} \beta_2 \\ \mathbf{0} \end{pmatrix}, \quad \text{and} \quad H_2 \stackrel{def}{=} \begin{pmatrix} I_2 & \\ & \widehat{H}_2 \end{pmatrix}$$

- Similarity transform on A with H_2,

$$H_2\,H_1\,A\,H_1^T\,H_2^T = \begin{pmatrix} \alpha_1 & x & \mathbf{x}_1^T\,\widehat{H}_2^T \\ \beta_1 & \alpha_2 & \left(\widehat{H}_2\,\mathbf{b}_2\right)^T \\ \mathbf{0} & \widehat{H}_2\,\mathbf{a}_2 & \widehat{H}_2\,\widehat{A}_2\,\widehat{H}_2^T \end{pmatrix} \stackrel{def}{=} \begin{pmatrix} \alpha_1 & x & x & \widehat{\mathbf{x}}_1^T \\ \beta_1 & \alpha_2 & x & \mathbf{x}_2^T \\ 0 & \beta_2 & \alpha_3 & \mathbf{b}_3^T \\ \mathbf{0} & \mathbf{0} & \mathbf{a}_3 & \widehat{A}_3 \end{pmatrix}.$$

- Repeat similarity transform procedure on $\mathbf{a}_3$ and $\widehat{A}_3$.
- Matrix becomes Upper Hessenberg at end of procedure.

Non-symmetric Reduction (III)

- After $n-1$ Householder Reflections,

$$H_{n-1} \cdots H_1 \, A \, H_1^T \cdots H_{n-1}^T = \begin{pmatrix} x & x & x & \cdots & x \\ x & x & x & \cdots & x \\ & x & x & \cdots & x \\ & & \ddots & \ddots & \vdots \\ & & & x & x \end{pmatrix}, \quad \text{or}$$

$$A = H \begin{pmatrix} x & x & x & \cdots & x \\ x & x & x & \cdots & x \\ & x & x & \cdots & x \\ & & \ddots & \ddots & \vdots \\ & & & x & x \end{pmatrix} H^T,$$

where $H = H_1^T \cdots H_{n-1}^T$.

Non-symmetric Reduction (IV)

- $n-1$ Householder Reflections,

$$H_{n-1}\,\cdots\,H_1\,A\,H_1^T\,\cdots\,H_{n-1}^T = \begin{pmatrix} x & x & x & \cdots & x \\ x & x & x & \cdots & x \\ & x & x & \cdots & x \\ & & \ddots & \ddots & \vdots \\ & & & x & x \end{pmatrix}.$$

- For each $1 \leq j \leq n-1$, one of the main costs of j-th Householder reduction

$$\widehat{H}_j\,\widehat{A}_j\,\widehat{H}_j^T \quad = \quad \left(I - 2\,\mathbf{v}_j\,\mathbf{v}_j^T\right)\,\widehat{A}_j\,\left(I - 2\,\mathbf{v}_j\,\mathbf{v}_j^T\right),$$

  namely about $8(n-j)^2$ multiplications and additions.
- another one of the main costs

$$\overline{A}_j\,\widehat{H}_j^T = \overline{A}_j\,\left(I - 2\,\mathbf{v}_j\,\mathbf{v}_j^T\right),$$

  where $\overline{A}_j \in \mathbb{R}^{j\times(n-j)}$ is submatrix on top of $\widehat{A}_j$. This costs about $4j(n-j)$ multiplications and additions.
- Grand total is about $\sum_{j=1}^{n-1} 8(n-j)^2 + 4j(n-j) \approx \frac{10}{3}\,n^3$ multiplications and additions.

Basic Tool: Givens rotation $G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \; c^2 + s^2 = 1$

- G is orthogonal: $G^T G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
- Given any non-zero $\mathbf{a} \in \mathbb{R}^2$,

$$G\,\mathbf{a} = \begin{pmatrix} \|\mathbf{a}\|_2 \\ 0 \end{pmatrix}, \quad \text{for} \begin{pmatrix} c \\ s \end{pmatrix} \stackrel{def}{=} \mathbf{a}\,/\|\mathbf{a}\|_2 \quad (G = \mathbf{I} \text{ for } \mathbf{a} = \mathbf{0}.)$$

- Givens (1910 – 1993)

§9.5 The QR Algorithm

Given: *Symmetric tridiagonal*

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Want: *All eigenvalues of A.*

Method: *The QR Algorithm.*

(One of the best algorithms invented last century.)

Idea: *Repeatedly compute* QR FACTORIZATION

$$A = Q\,R = Q \begin{pmatrix} x & x & x & 0 & \cdots & 0 \\ & x & x & x & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & x & x & x \\ & & & & x & x \\ & & & & & x \end{pmatrix}.$$

$O(n)$ Algorithm to eliminate sub-diagonal entries (I)

Notation: $$G_j = \begin{pmatrix} I_{j-1} & & \\ & \begin{pmatrix} c_j & s_j \\ -s_j & c_j \end{pmatrix} & \\ & & I_{n-j} \end{pmatrix}, \quad j = 1, \cdots, n.$$

Eliminate first sub-diagonal entry with G_1

$$G_1 \begin{pmatrix} x & x & 0 & \cdots & 0 & 0 \\ x & x & x & \cdots & 0 & 0 \\ & x & x & \cdots & 0 & 0 \\ & & x & \cdots & x & x \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix} = \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ & x & x & \cdots & 0 & 0 \\ & & x & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix},$$

$O(n)$ Algorithm to eliminate sub-diagonal entries (II)

Notation: $$G_j = \begin{pmatrix} I_{j-1} & & \\ & \begin{pmatrix} c_j & s_j \\ -s_j & c_j \end{pmatrix} & \\ & & I_{n-j} \end{pmatrix}, \quad j = 1, \cdots, n.$$

Eliminate second sub-diagonal entry with G_2

$$G_2 \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ & x & x & \cdots & 0 & 0 \\ & & x & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix} = \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \widehat{x} & \widehat{x} & \cdots & 0 & 0 \\ & 0 & \widehat{x} & \cdots & 0 & 0 \\ & & x & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix},$$

$O(n)$ Algorithm to eliminate sub-diagonal entries (III)

$$\text{Notation:} \quad G_j = \begin{pmatrix} I_{j-1} & & \\ & \begin{pmatrix} c_j & s_j \\ -s_j & c_j \end{pmatrix} & \\ & & I_{n-j} \end{pmatrix}, \quad j = 1, \cdots, n.$$

Eliminate three sub-diagonal entry with G_3

$$G_3 \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \widehat{x} & \widehat{x} & \cdots & 0 & 0 \\ & 0 & \widehat{x} & \cdots & 0 & 0 \\ & & x & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix} = \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \widehat{x} & \widehat{x} & \cdots & 0 & 0 \\ & 0 & \check{x} & \cdots & 0 & 0 \\ & & 0 & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix},$$

$O(n)$ Algorithm to eliminate sub-diagonal entries (IV)

Eliminate first three sub-diagonal entries with $G_3\, G_2\, G_1$

$$G_3\, G_2\, G_1 \begin{pmatrix} x & x & 0 & \cdots & 0 & 0 \\ x & x & x & \cdots & 0 & 0 \\ & x & x & \cdots & 0 & 0 \\ & & x & \cdots & x & x \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix} = \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \widehat{x} & \widehat{x} & \cdots & 0 & 0 \\ & 0 & \check{x} & \cdots & 0 & 0 \\ & & 0 & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix},$$

$O(n)$ Algorithm to eliminate sub-diagonal entries (V)

Eliminate all sub-diagonal entries with $G_{n-1} \cdots G_2 \, G_1$

$$G_{n-1} \cdots G_2 \, G_1 \begin{pmatrix} x & x & 0 & \cdots & 0 & 0 \\ x & x & x & \cdots & 0 & 0 \\ & x & x & \cdots & 0 & 0 \\ & & x & \cdots & x & x \\ & & & \ddots & \vdots & \vdots \\ & & & & x & x \end{pmatrix} = \begin{pmatrix} \overline{x} & \overline{x} & \overline{x} & \cdots & 0 & 0 \\ 0 & \widehat{x} & \widehat{x} & \cdots & 0 & 0 \\ & 0 & \check{x} & \cdots & 0 & 0 \\ & & 0 & \cdots & 0 & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & 0 & \underline{x} \end{pmatrix}$$

$$\stackrel{def}{=} R.$$

$O(n)$ cost for QR factorization with Givens rotation (II)

$$A \;=\; (G_{n-1} \cdots G_2 \, G_1)^T \; R \stackrel{def}{=} Q \, R.$$

Now define $\widehat{A} = R \, Q$.

- $\widehat{A} = Q^T A \, Q$. $\widehat{A}$ and A have the same eigenvalues.
- Assume A non-singular. $Q = A \, R^{-1}$ upper Hessenberg.
- $\widehat{A} = R \, Q$. must be upper Hessenberg as well.
- $\widehat{A} = Q^T A \, Q$ is symmetric. Hence $\widehat{A}$ must be itself symmetric tridiagonal.

QR Algorithm

Algorithm 1 The QR Algorithm

Initialize: $A^{(0)} = A$, $k = 0$, $Q^{(0)} = I$.

while NOT YET CONVERGED **do**

  Compute QR factorization $A^{(k)} = Q\,R$.

  Compute $A^{(k+1)} = R\,Q$, $Q^{(k+1)} = Q^{(k)}\,Q$

  $k = k + 1$

end while