Lecture 9

Hessenberg form

- Section 5.6.2, p.211-213
- Schur triangular form of a matrix
- An attempt to compute Schur factorization $Q^TAQ = T$ assuming that $A \in \mathbb{R}_{n\times n}$ has real eigenvalues.

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \overset{Q_1^T}{\rightarrow} \underset{Q_1^TA}{\begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix}} \overset{Q_1}{\rightarrow} \underset{Q_1^TAQ_1}{\begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix}}$$

- The right multiplication destroys the zeros previously introduced.
- Impossible due to Abel's theorem to obtain Schur triangular form of $n \times n$ A in finitely many steps.
- But there is a finite algorithm to bring A to an 'almost' upper triangular form -Hessenberg form
- A (upper) Hessenberg matrix is of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \ddots & \ddots & \vdots \\ & & a_{n,n-1} & a_{nn} \end{bmatrix}$$

  It is almost upper triangular with a subdiagonal.

- Reduction to Hessenberg matrix.

  Theorem 1. For each $A \in \mathbb{R}_{n\times n}$, there is an orthogonal Q such that Q^TAQ is Hessenberg.

The algorithm via Householder reduction

- $$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \overset{Q_1^T}{\longrightarrow} \underset{Q_1^T A}{\begin{bmatrix} \times & \times & \times & \times \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix}} \overset{Q_1}{\longrightarrow} \underset{Q_1^T A Q_1}{\begin{bmatrix} \times & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \times & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix}}$$

- The established zeros are kept.

- Continue with column 2:

$$\underset{Q_1^T A Q}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}} \overset{Q_2^T}{\longrightarrow} \underset{Q_2^T Q_1^T A Q_1}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix}} \overset{Q_2}{\longrightarrow} \underset{Q_2^T Q_1^T A Q_1 Q_2}{\begin{bmatrix} \times & \times & \mathbf{x} & \mathbf{x} \\ \times & \times & \mathbf{x} & \mathbf{x} \\ & \times & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix}}$$

- After $n-2$ step,

$$Q_{n-2}^T \cdots Q_2^T Q_1^T A Q_1 Q_2 \cdots Q_{n-2} = H = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}$$

- costs $\approx 10n^3/3$ flops. Less than or equal to what is saved during the first QR iteration. Not twice Householder QR.

- If A is symmetric, its Hessenberg form is tridiagonal. Costs $\approx 4n^3/3$ flops.

Implementation: Householder-Hessenberg method

- Program 29 **househess**: Householder-Hessenberg method

  Outputs : `H`, `Q` where $\mathtt{H} = \mathtt{Q}^{\mathtt{T}}\mathtt{AQ}$

```
0001 function [H,Q]=houshess(A)
0002 n=max(size(A));
0003 Q=eye(n);
0004 H=A;
0005 for k=1:(n-2),
0006      [v,beta]=vhouse(H(k+1:n,k));
0007     I=eye(k);
0008     N=zeros(k,n-k);
0009     m=length(v);
0010     R=eye(m)-beta*v*v';
0011     H(k+1:n,k:n)=R*H(k+1:n,k:n);
0012     H(1:n,k+1:n)=H(1:n,k+1:n)*R;
0013     P=[I, N; N', R];
0014     Q=Q*P;
0015 end
0016 return
```

- What can you improve in the above program (see line 0011)?

- Costs $\approx 10n^3/3$ flops. Well conditioned with respect to rounding error

- Stability:

$$\hat{H} = Q^T(A + \delta A)Q, \qquad \|\delta A\|_F \leq cn^2 u \|A\|_F$$

  where $\hat{H}$ is the Hessenberg computed by Program 29.

- Example: `A = hilb(4)`

Picking Householder vector

- Program 32 `vhouse`: Construction of the Householder vector

```
function [v,beta]=vhouse(x)
n=length(x);
x=x/norm(x);
s=x(2:n)'*x(2:n);
v=[1; x(2:n)];
if (s==0), beta=0;
else
   mu=sqrt(x(1)^2+s);
   if (x(1) <= 0)
      v(1)=x(1)-mu;
   else
      v(1)=-s/(x(1)+mu);
   end
   beta=2*v(1)^2/(s+v(1)^2);
   v=v/v(1);
end
return
```

Stability and Accuracy

- The Householder Hessenberg reduction is backward stable

$$\hat{Q}\hat{H}\hat{Q}^T = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

  where $\hat{Q}$ and $\hat{H}$ are the computed matrices.

- MATLAB's `hess` command `[Q,H]=hess(A)`

- Read p.212