Lecture 5

3.4.2 Cholesky factorization

- $A \in \mathbb{R}_{n \times n}$ is symmetric if $a_{ij} = a_{ji}$ or $A = A^T$.

- A symmetric matrix A is positive definite if $x^T A x > 0$ for all $x \neq 0$.

- Positive definite matrix has positive eigenvalues.

- Arise in many applications, for example, covariance matrices in Statistics.

- If A is positive definite, then A is nonsingular and thus the linear system $Ax = b$ has exactly one solution.

- A symmetric matrix A is positive definite if and only if its principal submatrices are nonsingular.

- Theorem 3.6 If $A \in \mathbb{R}_{n \times n}$ is positive definite, then there exists a unique upper triangular matrix H with positive diagonal entries such that
$$A = H^T H$$
Proof: See p.84-85

The algorithm

$A = H^T H$. Let $H^T = (h_{ij})$ (lower triangular). Then the (i,j) entry of H is h_{ji}.

- Consider the lower triangular entries of A:

  $$h_{11} = \sqrt{a_{11}}$$

  For $i = 2, \ldots, n$,

  $$a_{ij} = \sum_{k=1}^{j} h_{ik} h_{jk}, \qquad j = 1, \ldots, i$$

  Then

  $$a_{ij} = \sum_{k=1}^{j-1} h_{ik} h_{jk} + h_{ij} h_{jj}, \qquad j = 1, \ldots, i$$

  Thus

  $$h_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} h_{ik} h_{jk} \right) / h_{jj}, \qquad j = 1, \ldots, i-1$$

  and

  $$h_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} h_{ik}^2 \right)^{1/2}$$

  Solving the equations in the order (1,1), (2,1), (2,2), (3,1), (3,2), (3,3), ... (n,n).

- Cholesky's method serves a test of positive definiteness. If A is not positive definite, the algorithm must fail. The algorithm fails if and only if at some step the number under the square root sign is negative or zero. It is the best general test of positive definiteness known.

Implementation of Cholesky factorization

- Consider the following reduction algorithm: Set

$$H^T = \begin{bmatrix} h_{11} & 0 \\ \hat{h}^T & \hat{H}^T \end{bmatrix} = (h_{ij}), \qquad A = \begin{bmatrix} a_{11} & \hat{a} \\ \hat{a}^T & \hat{A} \end{bmatrix} = (a_{ij})$$

  Then

$$\begin{bmatrix} h_{11} & 0 \\ \hat{h}^T & \hat{H}^T \end{bmatrix} \begin{bmatrix} h_{11} & \hat{h} \\ 0 & \hat{H} \end{bmatrix} = \begin{bmatrix} a_{11} & \hat{a} \\ \hat{a}^T & \hat{A} \end{bmatrix}$$

  This leads to

$$\begin{aligned} h_{11} &= \sqrt{a_{11}} \\ \hat{H}^T \hat{H} &= \hat{A} - \hat{h}^T \hat{h} \end{aligned}$$

- `chol2`: Cholesky factorization

```
function [H] = chol2(A)
% Find upper triangular matrix H s.t. A = H'H
% Overwrite the matrix H' in the lower triangular part of A
[n,n]=size(A);
for k=1:n-1
% Compute main diagonal elt. and then scale the k-th column
   A(k,k)=sqrt(A(k,k));
   A(k+1:n,k)=A(k+1:n,k)/A(k,k);
% Update lower triangle of the trailing (n-k) by (n-k) block
%
   for j=k+1:n
      A(j:n,j)=A(j:n,j)-A(j:n,k)*A(j,k);
   end
end
A(n,n)=sqrt(A(n,n));
H = (tril(A))';
return
```

  The lower triangular part `tril(A)` is H^T such that $A = H^T H$.

- cost $\approx n^3/3$.

- MATLAB's command is `chol(A)`

Error analysis:

- $$\tilde{H}^T\tilde{H} = A + \delta A$$
  $$\frac{\|\delta A\|_2}{\|A\|_2} \le 8n(n+1)u$$
  with the assumption that
  $$2n(n+1)u \le 1-(n+1)u$$
  Thus backward stable.

- But the forward errors in $\tilde{H}$ might be large (like QR Householder)
  $$\frac{\|\tilde{H}-H\|}{\|H\|} = O(K(T)u)$$

- Backward stable algorithm:
  $$(A+\delta A)\hat{x} = b$$
  $$\frac{\|\delta A\|}{\|A\|} = O(u)$$