# SI231b: Matrix Computations

## Lecture 8: Special LU Factorization and Computational Complexity

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Oct. 12, 2021

For $\mathrm{A} \in \mathbb{R}^{n \times n}$, and a permutation matrix $\mathrm{P}_1$

$$\mathrm{P}_1\mathrm{A} = \left[\begin{array}{c|c} a_{11}^{(0)} & \mathrm{v}^T \\ \hline \mathrm{u} & \mathrm{A}_1' \end{array}\right] = \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{11}^{(0)}\mathrm{u} & \mathrm{I}_{n-1} \end{array}\right]}_{\mathrm{L}_1} \underbrace{\left[\begin{array}{c|c} a_{11}^{(0)} & \mathrm{v}^T \\ \hline 0 & \mathrm{A}_1' - 1/a_{11}^{(0)}\mathrm{u}\mathrm{v}^T \end{array}\right]}_{\mathrm{U}_1}$$

Then repeat the above procedure to $\mathrm{A}_1' - 1/a_{11}^{(0)}\mathrm{u}\mathrm{v}^T$, i.e.,

$$\mathrm{P}_2'\left(\mathrm{A}_1' - 1/a_{11}^{(0)}\mathrm{u}\mathrm{v}^T\right) = \left[\begin{array}{c|c} a_{22}^{(1)} & \mathrm{w}^T \\ \hline \mathrm{s} & \mathrm{A}_2' \end{array}\right]$$
$$= \left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{22}^{(1)}\mathrm{s} & \mathrm{I}_{n-2} \end{array}\right] \left[\begin{array}{c|c} a_{22}^{(1)} & \mathrm{w}^T \\ \hline 0 & \mathrm{A}_2' - 1/a_{22}^{(1)}\mathrm{s}\mathrm{w}^T \end{array}\right]$$

Denote $\mathrm{P}_2 = \begin{bmatrix} 1 & \\ & \mathrm{P}_2' \end{bmatrix}$, we obtain (next page)

$$
P_2P_1A = \underbrace{\begin{bmatrix} 1 & & \\ & 1 & \\ \frac{1}{a_{11}^{(0)}}P_2'u & \frac{1}{a_{22}^{(1)}}s & I_{n-2} \end{bmatrix}}_{L_2} \underbrace{\begin{bmatrix} a_{11}^{(0)} & & v^T \\ & a_{22}^{(1)} & w^T \\ & & A_2' - \frac{1}{a_{22}^{(1)}}sw^T \end{bmatrix}}_{U_2}
$$

- following the above notations, $L = L_{n-1}$, $U = U_{n-1}$
- $P_k$ only acts on the first $(k-1)$ columns of $L_k$
- algorithm style, suitable for computer implementation

**Remark**:

- Gaussian elimination tells why you can perform an LU factorization, and when does it exist
- the recursive approach tells how you can compute the LU factorization on a modern computer

# Example

Please compute an LU factorization with partial pivoting using the method introduced in the last page for

$$\begin{bmatrix} 2 & 4 & 5 \\ -3 & 1 & 4 \\ 4 & 2 & 3 \end{bmatrix}$$

Answer:

$$\begin{bmatrix} & & 1 \\ 1 & & \\ & 1 & \end{bmatrix} \begin{bmatrix} 2 & 4 & 5 \\ -3 & 1 & 4 \\ 4 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & & \\ \frac{1}{2} & 1 & \\ -\frac{3}{4} & \frac{5}{6} & 1 \end{bmatrix} \begin{bmatrix} 4 & 2 & 3 \\ & 3 & \frac{7}{2} \\ & & \frac{10}{3} \end{bmatrix}$$

**LU Factorization without Pivoting**:

```
U = A, L = I;
for k = 1 :  n-1
     for j = k+1 :  n
          ℓ_jk = u_jk / u_kk
          u_{j,k:n} = u_{j,k:n} − ℓ_jk u_{k,k:n}
     end
end
U = triu(U)
```

Operations count:

- $\mathcal{O}\left(\frac{2}{3}n^3\right)$ flops

Please give your own explanation

**LU Factorization with Partial Pivoting**:

```
U = A, L = I, P = I;
for k = 1 :  n-1
     select i ≥ k to maximize |u_ik|
     u_{k,k:m} ↔ u_{i,k:m} (exchange of rows)
     ℓ_{k,1:k-1} ↔ ℓ_{i,1:k-1}
     p_{k,:} ↔ p_{i,:}
     for j = k+1 :  n
          ℓ_jk = u_jk / u_kk
          u_{j,k:n} = u_{j,k:n} - ℓ_jk u_{k,k:n}
     end
end
U = triu(U)
```

Operations count:

- $\mathcal{O}\left(\frac{2}{3}n^3\right)$ flops, flops count of partial pivoting?

# $\mathrm{LDL}^T$ Factorization for Symmetric Matrices

**Theorem**

If $\mathrm{A} \in \mathbb{R}^{n \times n}$ is symmetric and nonsingular, and every leading principal sub-matrix $\mathrm{A}_{\{1,\ldots,k\}}$ satisfies

$$\det(\mathrm{A}_{\{1,\ldots,k\}}) \neq 0,$$

for $k = 1, 2, \cdots, n-1$, then there exists a lower-triangular matrix L with unit entries and a diagonal matrix

$$\mathrm{D} = \mathrm{diag}(d_1, d_2, \cdots, d_n),$$

where $d_i \neq 0$ for $i = 1, 2, \cdots, n$, such that $\mathrm{A} = \mathrm{LDL}^T$. The factorization is unique.

Proof: making use of the LU factorization

Computational complexity: not surprisingly $\mathcal{O}\left(\frac{n^3}{3}\right)$

# $\mathrm{LDL}^T$ Factorization with Symmetric Pivoting

## Symmetry is preferred

If A is symmetric, and $\mathrm{P}_1$ is a permutation matrix

- $\mathrm{P}_1\mathrm{A}$ is not symmetric
- $\mathrm{P}_1\mathrm{A}\mathrm{P}_1^T$ is symmetric

Consider the following

$$\begin{aligned}\mathrm{P}_1\mathrm{A}\mathrm{P}_1^T &= \begin{bmatrix} \alpha & \mathrm{v}^T \\ \mathrm{v} & \mathrm{A}_1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \\ 1/\alpha\mathrm{v} & \mathrm{I}_{n-1} \end{bmatrix} \begin{bmatrix} \alpha & \\ & \tilde{\mathrm{A}}_1 \end{bmatrix} \begin{bmatrix} 1 & 1/\alpha\mathrm{v}^T \\ & \mathrm{I}_{n-1} \end{bmatrix},\end{aligned}$$

with $\tilde{\mathrm{A}}_1 = \mathrm{A}_1 - 1/\alpha\mathrm{v}\mathrm{v}^T$ also symmetric.

**Note**: with symmetric pivoting, $\alpha$ is some diagonal entry $a_{ii}$, why?

When the procedure terminates, $\mathrm{PAP}^T = \mathrm{LDL}^T$ where

$$\mathrm{P} = \mathrm{P}_{n-1} \cdots \mathrm{P}_2\mathrm{P}_1$$

**Symmetric Positive Definite (SPD)**

$\mathrm{M} = \mathrm{M}^T \in \mathbb{R}^{n\times n}$ is SPD iff (if and only if)

$$\mathrm{x}^T \mathrm{M} \mathrm{x} > 0, \quad \forall \mathrm{x} \in \mathbb{R}^n \backslash 0$$

**Properties of SPD Matrices**:

- real positive eigenvalues
- positive diagonal entries
- all principle sub-matrices are SPD
- $\mathrm{A} \in \mathbb{R}^{n\times n}$ is SPD and $\mathrm{X} \in \mathbb{R}^{n\times r}$ has full rank, then $\mathrm{X}^T\mathrm{A}\mathrm{X}$ is also SPD

### Recursive Factorization

For an SPD matrix $A \in \mathbb{R}^{n\times n}$,

$$A = \begin{bmatrix} a_{11} & w^T \\ w & A_1 \end{bmatrix}$$
$$= \underbrace{\begin{bmatrix} \sqrt{a_{11}} & \\ 1/\sqrt{a_{11}}w & I_{n-1} \end{bmatrix}}_{L_1} \underbrace{\begin{bmatrix} 1 & \\ & A_1 - 1/a_{11}ww^T \end{bmatrix}}_{D_1} \underbrace{\begin{bmatrix} \sqrt{a_{11}} & 1/\sqrt{a_{11}}w^T \\ & I_{n-1} \end{bmatrix}}_{L_1^T}$$

**Require**: the (1, 1) entry of $\left(A_1 - 1/a_{11}ww^T\right)$ should be positive to continue.

**Note**: $\left(A_1 - 1/a_{11}ww^T\right)$ is a principle sub-matrix of $L_1^{-1}AL_1^{-T}$.

Following the same principle, when the procedure terminates,

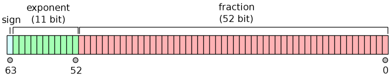
- $L_n = L$, $D_n = I_n$
- $A = LL^T$: Cholesky factorization
- $\mathcal{O}\left(\frac{1}{3}n^3\right)$ flops, half of LU factorization

## IEEE Standard for Floating-Point Arithmetic (IEEE 754)

- single format, 32 bit
- double format, 64 bit

Take the double format for example,

- 1 bit for sign;
- 52 bits for the mantissa;
- 11 bits for the exponent;




IEEE standard stipulates that each arithmetic operation be correctly rounded, meaning that the computed result is the rounded version of the exact result.

## Machine Precision

Resolution is traditionally summarized by a number known as machine epsilon, i.e., $\varepsilon_m$

$$\varepsilon_m = \frac{1}{2} \times (\text{gap between 1 and next largest floating point number})$$

- $\varepsilon_m \approx 5.96 \times 10^{-8}$ for single format
- $\varepsilon_m \approx 1.11 \times 10^{-16}$ for double format

Try the eps command in Matlab to get $\varepsilon_m$

### Property

$$\forall x \in \mathbb{R}, \text{there exists } x' \in \mathbb{F}, \text{ such that } |x - x'| < \varepsilon_m |x|$$

where $\mathbb{F}$ represents the set of floating point numbers. Or equivalently,

$$\forall x \in \mathbb{R}, \text{ there exists } \varepsilon \text{ with } |\varepsilon| \leq \varepsilon_m, \text{ such that } fl(x) = x(1+\varepsilon)$$

## Matrix Condition Number

Consider solving the linear equation $Ax = b$ using direct methods, such as LUP/Cholesky factorization, which can be represented by

$$(A + \sigma A)(x + \sigma x) = b.$$

Making use of $Ax = b$ and dropping out the product $\sigma A \sigma x$, we obtain

$$\frac{\|\sigma x\|}{\|x\|} \Bigg/ \frac{\|\sigma A\|}{\|A\|} \leq \|A\|\|A^{-1}\|$$

where $\|A\|\|A^{-1}\|$ defines the condition number of the matrix A and is often denoted by $\kappa(A)$.

The linear equation $Ax = b$ is

- well-conditioned if small $\sigma A$ leads to small $\sigma x$ (small $\kappa(A)$)
- ill-conditioned if small $\sigma A$ leads to large $\sigma x$ (large $\kappa(A)$)

**Note**: here the meaning of ''small" and''large" depends on the application.

# Readings

You are supposed to read

- Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, 2013.

  Chapter 2.6 – 2.7, Chapter 4.1 – 4.4

- Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*, SIAM, 1997.

  Lecture 12 – 13, 23