SI231b: Matrix Computations

Lecture 6: Solution of Special Linear Systems

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Sept. 24, 2020

Outline

- Computing LU Factorization via Recursion
- LDL^T Factorization for Symmetric Systems
- LDL^T Factorization with Symmetric Pivoting
- Cholesky Factorization for SPD Systems
- Banded Matrices Factorization
- Floating Point Arithmetic
- Condition of Systems of Equations

An Alternative Approach

For $A \in \mathbb{R}^{n\times n}$, and a permutation matrix P_1

$$P_1 A = \left[\begin{array}{c|c} a_{11}^{(0)} & v^T \\ \hline u & A_1' \end{array}\right] = \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{11}^{(0)} u & I_{n-1} \end{array}\right]}_{L_1} \underbrace{\left[\begin{array}{c|c} a_{11}^{(0)} & v^T \\ \hline 0 & A_1' - 1/a_{11}^{(0)} uv^T \end{array}\right]}_{U_1}$$

Then repeat the above procedure to $A_1' - 1/a_{11}^{(0)} uv^T$, i.e.,

$$\begin{aligned} P_2' \left(A_1' - 1/a_{11}^{(0)} uv^T\right) &= \left[\begin{array}{c|c} a_{22}^{(1)} & w^T \\ \hline s & A_2' \end{array}\right] \\ &= \left[\begin{array}{c|c} 1 & 0 \\ \hline 1/a_{22}^{(1)} s & I_{n-2} \end{array}\right] \left[\begin{array}{c|c} a_{22}^{(1)} & w^T \\ \hline 0 & A_2' - 1/a_{22}^{(1)} sw^T \end{array}\right] \end{aligned}$$

Denote $P_2 = \begin{bmatrix} 1 & \\ & P_2' \end{bmatrix}$, 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

LDL^T Factorization

LDL^T: LDU Factorization for Symmetric Matrices

Theorem

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

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

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

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

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

Proof: making use of the LU factorization

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

LDL^T Factorization with Symmetric Pivoting

Symmetry is preferred

If A is symmetric, and P_1 is a permutation matrix

- P_1A is not symmetric
- $P_1AP_1^T$ is symmetric

Consider the following

$$\begin{aligned} P_1AP_1^T &= \begin{bmatrix} \alpha & v^T \\ v & A_1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \\ 1/\alpha v & I_{n-1} \end{bmatrix} \begin{bmatrix} \alpha & \\ & \tilde{A}_1 \end{bmatrix} \begin{bmatrix} 1 & 1/\alpha v^T \\ & I_{n-1} \end{bmatrix}, \end{aligned}$$

with $\tilde{A}_1 = A_1 - 1/\alpha vv^T$ also symmetric.

Note: with symmetric pivoting, α is some diagonal entry a_{ii}, why?

When the procedure terminates, $PAP^T = LDL^T$ where

$$P = P_{n-1} \cdots P_2P_1$$

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 $\mathsf{A} \in \mathbb{R}^{n\times n}$,

$$\mathsf{A} = \begin{bmatrix} a_{11} & \mathsf{w}^T \\ \mathsf{w} & \mathsf{A}_1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \sqrt{a_{11}} & \\ 1/\sqrt{a_{11}}\mathsf{w} & \mathsf{I}_{n-1} \end{bmatrix}}_{\mathsf{L}_1} \underbrace{\begin{bmatrix} 1 & \\ & \mathsf{A}_1 - 1/a_{11}\mathsf{w}\mathsf{w}^T \end{bmatrix}}_{\mathsf{D}_1} \underbrace{\begin{bmatrix} \sqrt{a_{11}} & 1/\sqrt{a_{11}}\mathsf{w}^T \\ & \mathsf{I}_{n-1} \end{bmatrix}}_{\mathsf{L}_1^T}$$

Require: the (1, 1) entry of $\left(\mathsf{A}_1 - 1/a_{11}\mathsf{w}\mathsf{w}^T\right)$ should be positive to continue.

Note: $\left(\mathsf{A}_1 - 1/a_{11}\mathsf{w}\mathsf{w}^T\right)$ is a principle sub-matrix of $\mathsf{L}_1^{-1}\mathsf{A}\mathsf{L}_1^{-T}$.

Following the same principle, when the procedure terminates,

- $\mathsf{L}_n = \mathsf{L}$, $\mathsf{D}_n = \mathsf{I}_n$
- $\mathsf{A} = \mathsf{L}\mathsf{L}^T$: Cholesky factorization
- $\mathcal{O}\left(\frac{1}{3}n^3\right)$ flops, half of LU factorization

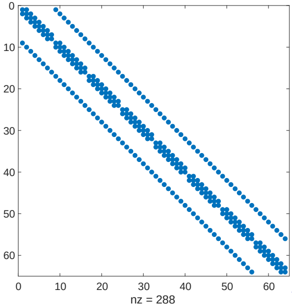
Banded Matrix

For matrix $A \in \mathbb{R}^{n\times n}$, A is called to have

- upper bandwidth q if $a_{ij} = 0$ whenever $j > i + q$;
- lower bandwidth p if $a_{ij} = 0$ whenever $i > j + p$.

Example: discretization of the Laplace operator in $\mathbb{R}^2$

$$\Delta = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$



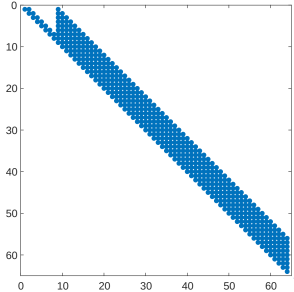
- L inheritates the lower bandwidth of A
- U inheritates the upper bandwidth of A

Theorem

Suppose $A \in \mathbb{R}^{n\times n}$ has an LU factorization $A = LU$. If A has upper bandwidth q and lower bandwidth p, then U has upper bandwidth q and L has lower bandwidth p.

Proof: cf. Theorem 4.3.1 in [Golub and van Loan]

Cholesky factor of the discretized Laplace operator



Banded LU Factorization with Partial Pivoting

For a nonsingular banded matrix $A \in \mathbb{R}^{n \times n}$ with upper bandwidth q and lower bandwidth p, after performing the LU factorization with partial pivoting using Gaussian elimination,

- the upper bandwidth of U is $p + q$
- the lower bandwidth of L is complicated to analyze

Cf. Theorem 4.3.2 in [Golub and van Loan] for details.

Note:

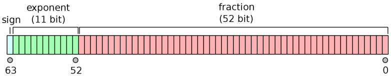
- computational cost of LU factorization for banded matrices is huge
  - banded matrices are often huge (big n)
  - LU factorization costs $\mathcal{O}\left(\frac{2n^3}{3}\right)$ flops, not affordable for large n

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., ε_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 ε_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

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

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 σA leads to small σx (small $\kappa(A)$)
- ill-conditioned if small σA leads to large σx (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