

Lecture 07 - B

§8.1 LDL and Cholesky Factorization

T. Takahashi

Department of Mathematics and Statistics
University of New Mexico



Goals

We will answer these questions:

- Can we do better than LU factorization?
- What is the *LDL* factorization? What are its advantages?
- What is the Cholesky factorization? What are its advantages?



Is LU Always the Right Tool?

- LU works for a general matrix
- But LU:
 - Stores both L and U
 - Computes redundant information if A has structure
 - Requires pivoting for stability
- **Key question:** What if A has extra structure?



Two Other Factorizations

LDL Factorization

- Form $A = LDL^T$
- L = unit lower triangular matrix
- D = diagonal matrix
- Special Requirement: A must be symmetric

Cholesky Factorization

- Form $A = \tilde{L}\tilde{L}^T$
- $\tilde{L}$ = lower triangular matrix
- Special Requirement: A must be symmetric and positive definite



LDL Factorization



Symmetric Matrices

Definition: Symmetry

A square matrix A is considered **symmetric** if $A = A^T$.

- A matrix is symmetric if it is invariant with respect to its transpose:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 4 & 5 & 6 \\ 7 & 6 & 9 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 4 & 7 \\ 4 & 5 & 6 \\ 7 & 6 & 9 \end{bmatrix}$$

- Many symmetric matrices appear naturally in applications

$$A = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 1 & -2 \\ 0 & -2 & 1 \end{bmatrix}$$



Symmetric Matrices

Definition: Symmetry

A square matrix A is considered **symmetric** if $A = A^T$.

- A matrix is symmetric if it is invariant with respect to its transpose:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 4 & 5 & 6 \\ 7 & 6 & 9 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 4 & 7 \\ 4 & 5 & 6 \\ 7 & 6 & 9 \end{bmatrix}$$

- Many symmetric matrices appear naturally in applications

$$A = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 1 & -2 \\ 0 & -2 & 1 \end{bmatrix}$$

Question

What happens to the LU factorization if A happens to be symmetric?

Factorizing a Symmetric Matrix

Part 1

Begin

We assume that $A = LU$ and $A = A^T$. Let's see what happens.

- First, write things in terms of L and U

$$A = A^T$$

$$(LU) = (LU)^T$$

$$LU = U^T L^T$$

- With a little manipulation we derive

$$LU = U^T L^T$$

$$L^{-1}LU(L^T)^{-1} = L^{-1}U^T L^T (L^T)^{-1}$$

$$UL^{-T} = L^{-1}U^T$$

- Let $B = UL^{-T}$, then the equation above can be written

$$B = B^T$$



Factorizing a Symmetric Matrix

Part 2

Question

What is so special about $B = B^T$?

- **Lower triangular $\times$ lower triangular is lower triangular.**

$$\begin{bmatrix} \bullet & & & \\ \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & & & \\ \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & & & \\ \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix}$$

- **Upper triangular $\times$ upper triangular is upper triangular.**

$$\begin{bmatrix} \bullet & & & \\ & \bullet & & \\ & & \bullet & \\ & & & \bullet \end{bmatrix} \begin{bmatrix} \bullet & & & \\ & \bullet & & \\ & & \bullet & \\ & & & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & & & \\ & \bullet & & \\ & & \bullet & \\ & & & \bullet \end{bmatrix}$$

- Therefore,

$$B = UL^{-T} = \text{upper} \times \text{upper} = \text{upper}$$



Factorizing a Symmetric Matrix

Part 3

- B is upper triangular
- So B^T must be lower triangular
- But we found that $B = B^T$

Question

How is it possible for a matrix to be both upper triangular and lower triangular?

$$\begin{bmatrix} \bullet & & & \\ \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & & & \\ & \bullet & & \\ & & \bullet & \\ & & & \bullet \end{bmatrix}$$



Factorizing a Symmetric Matrix

Part 4

- Matrix B must be **diagonal**

$$B = B^T \implies B = D$$

- Solve for U :

$$B = D = UL^{-T} \implies U = DL^T \quad (1)$$

- Finally, we substitute into the LU decomposition

$$A = LU = LDL^T$$

Theorem

Suppose a symmetric, nonsingular matrix A admits an LU factorization without pivoting. Then

$$A = LDL^T,$$

where L is unit lower triangular and D is diagonal.

LDL Factorization

Method: LDL Factorization

- **Given:** Symmetric matrix A
- **Output:** Unit lower triangular matrix L and Diagonal matrix D , such that $A = LDL^T$
- **Idea:** Similar to LU except storage and calculation of U is not needed

Comments:

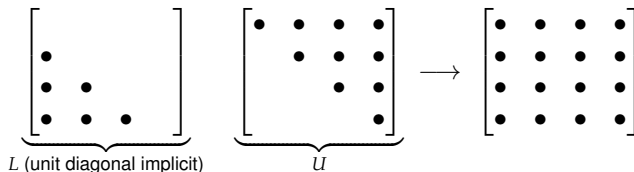
- For a general matrix, you must do $A = LU$
- If A is symmetric, you may do $A = LDL^T$
- Method is stable¹



¹Requires a special form of pivoting called Bunch-Kaufman

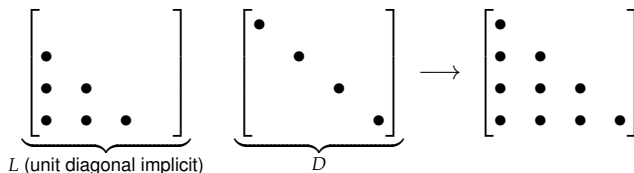
Memory

- **LU**: Store the strict lower triangle of L and the upper triangle of U .



$$\frac{n(n-1)}{2} + \frac{n(n+1)}{2} = \boxed{n^2}.$$

- **LDL^T**: Store the strict lower triangle of L and the diagonal of D .



$$\frac{n(n-1)}{2} + n = \boxed{\frac{n(n+1)}{2}}.$$



LU versus LDL

When does each factorization apply?

- **LU**: generally works for any nonsingular A
- **LDL^T**: requires symmetric A

Storage flop counts (dense $n \times n$)

	Memory (floats)	Flops (factor)
LU	n^2	$\frac{2}{3}n^3$
LDL^T	$\frac{n(n+1)}{2}$	$\frac{1}{3}n^3$

(LDL stores no upper part and replaces U 's diagonal by D .)

Key Idea

LDL is faster and requires less memory than LU, but we require A to be symmetric which is not always the case.

LDL Factorization

```
1 given A
2 output L, D
3 for j = 1...n
4      $\ell_{jj} = 1$ 
5      $d_j = a_{jj} - \sum_{v=1}^{j-1} d_v \ell_{jv}^2$ 
6     for i = j + 1...n
7          $\ell_{ji} = 0$ 
8          $\ell_{ij} = (a_{ij} - \sum_{v=1}^{j-1} \ell_{iv} d_v \ell_{jv}) / d_j$ 
9     end
10 end
```

Listing 1: LDL' Factorization

- It is a triple loop so LDL is still $\mathcal{O}(n^3)$ cost
- The LDL factorization requires slightly above half the amount of memory as LU factorization



Calculating LDL by Hand

Steps:

- 1 Calculate the usual $A = LU$ factorization
- 2 Set $U = DL^T$ and solve for D

Key Idea

- The L in LDL^T factorization is the same matrix
- We may solve for D from the equation $U = DL^T$



Multiplying by D

Useful Knowledge

Multiplying DA is the same as multiplying each row of A by the diagonal elements.

- Example: if

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 10 \\ 1 & 1 & 0 \end{bmatrix}$$

- Then DA is

$$DA = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 2 & 5 \\ 3 & 3 & 0 \end{bmatrix}$$



Example

Part 1

Example

Calculate the LDL factorization of the matrix

$$A = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 11 & 5 \\ -2 & 5 & 33 \end{bmatrix} \quad (\text{symmetric})$$

- First find the LU factorization. Eliminate elements under $a_{11} = 2$

$$M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boxed{2} & 4 & -2 \\ 4 & 11 & 5 \\ -2 & 5 & 33 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 3 & 9 \\ 0 & 9 & 31 \end{bmatrix}$$

- Eliminate the elements under $a_{22} = 3$. We obtain U .

$$M_2 M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & \boxed{3} & 9 \\ 0 & 9 & 31 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 3 & 9 \\ 0 & 0 & 4 \end{bmatrix} = U$$



Example

Part 2

- Next, find L using the elementary matrices

$$M_2 M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \implies L = M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{bmatrix}$$

- Find D using the equation $DL^T = U$

$$DL^T = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 3 & 9 \\ 0 & 0 & 4 \end{bmatrix} = U$$

- It is clear that

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

- Done. We found L and D .



Cholesky Factorization



Positive Definite

Definition: Positive Definite

A square matrix A is **positive definite** if for all $x \neq 0$

$$x^T A x > 0.$$

When A is both symmetric and positive definite, we abbreviate it as **SPD**.
Some mathematicians write

$$A > 0$$

to mean SPD.

- For a real symmetric matrix, positive definiteness is equivalent to having all positive eigenvalues. In particular, an SPD matrix is nonsingular.
- In the special case that A is SPD, we have an even better factorization

the Cholesky factorization!



Cholesky Factorization Derivation

Part 1

Begin

We assume that $A = LDL^T$ and that A is positive definite.

- Since A is positive definite, for every $x \neq 0$,

$$x^T A x > 0$$

$$x^T L D L^T x > 0$$

- Since we are free to choose x , let $x = L^{-T} e_i$ where $e_i = i^{th}$ column of I

$$(L^{-T} e_i)^T L D L^T (L^{-T} e_i) > 0$$

$$e_i^T (L^{-1} L) D (L^T L^{-T}) e_i > 0$$

$$e_i^T D e_i > 0$$

$$d_{ii} > 0$$

- Therefore, all **diagonal entries** of D are positive.



Cholesky Factorization Derivation

Part 2

- Since $d_{ii} > 0$ for all i , we can decompose $D = D^{1/2}D^{1/2}$.
- For instance,

$$D = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 9 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & \sqrt{7} & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & \sqrt{7} & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

- We can further decompose the LDL

$$\begin{aligned} A &= LDL^T \\ &= LD^{1/2}D^{1/2}L^T \\ &= (LD^{1/2})(D^{1/2}L^T) \\ &= (LD^{1/2})(LD^{1/2})^T \end{aligned}$$

- Let $\tilde{L} = LD^{1/2}$. Then the factorization becomes $A = \tilde{L}\tilde{L}^T$



Cholesky Factorization

Method: Cholesky Factorization

- **Given:** SPD matrix A
- **Output:** Lower triangular matrix $\tilde{L}$ such that $\tilde{L}\tilde{L}^T = A$
- **Idea:** This requires half the FLOPs and memory compared to LU

Comments:

- The upper triangular U from $A = LU$ is not needed
- The diagonal D from $A = LDL^T$ is not needed
- This is the most compact elimination-based decomposition
- Calculate $A = LDL^T$ first. Then $\tilde{L} = LD^{1/2}$.



Cholesky Factorization

```
1 given  $A$ 
2 output  $L$ 
3
4 for  $k = 1 \dots n$ 
5      $\ell_{kk} = \left( a_{kk} - \sum_{s=1}^{k-1} \ell_{ks}^2 \right)^{1/2}$ 
6     for  $i = k+1 \dots n$ 
7          $\ell_{ik} = \left( a_{ik} - \sum_{s=1}^{k-1} \ell_{is} \ell_{ks} \right) / \ell_{kk}$ 
8     end
9 end
```

- This algorithm is also $\mathcal{O}(n^3)$
- Cholesky and LDL have similar operation counts and the same factor storage: $\frac{n(n+1)}{2}$ numbers.

$$\ell_{kk} = \sqrt{a_{kk} - \sum_{s=1}^{k-1} \ell_{ks}^2}, \quad a_{kk} - \sum_{s=1}^{k-1} \ell_{ks}^2 > 0.$$



Pivoting



Existence of a Factorization

- $A = LDL^T$ only exists when A is symmetric
- $A = \tilde{L}\tilde{L}^T$ exists if and only if A is SPD.

Question

When does LU factorization exist? Does it exist for any square matrix?



Why Pivoting Is Needed

- Recall that Naive GE finds multipliers by dividing the pivot entries a_{kk}
- This can cause issues if the pivot is zero or very small, e.g.

$$A = \begin{bmatrix} \boxed{0} & 1 \\ 1 & 1 \end{bmatrix}$$

- **LU factorization is the same as GE so it needs pivoting**

Key Idea

Pivoting is needed in LU factorization for stability and avoiding zero pivots



Permutation Matrix

- A **permutation matrix** P is the identity matrix with some rows swapped
- Example:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Multiplying another matrix by a permutation matrix causes rows to swap

$$PA = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & x & x \\ y & y & y \\ z & z & z \end{bmatrix} = \begin{bmatrix} y & y & y \\ x & x & x \\ z & z & z \end{bmatrix}$$

- P matrix is very easy to invert

$$P^{-1} = P^T$$



Pivoting with LU Factorization

- If pivoting is necessary, multiply by P

$$PA = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \\ 2 & 5 & 8 \\ 1 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 2 & 5 & 8 \\ 0 & 1 & 2 \\ 1 & 6 & 9 \end{bmatrix} \text{ swap } R_1 \text{ and } R_2$$

- Thus, partial pivoting fits seamlessly into the factorization algorithm, e.g.

$$M_2 P_2 M_1 P_1 A = U$$

- In practice, LU factorization with pivoting matrix P has the form

$$PA = LU$$

Key Idea

Not all matrices have an LU factorization, but all matrices have a pivoted LU factorization.

Pivoting with LU Factorization

- Matlab's `lu()` function uses partial pivoting

```
1 [L,U,P]=lu(A)
2 L =
3     1.0000         0         0         0
4     0.9917     1.0000         0         0
5     0.8920    -0.3250     1.0000         0
6     0.1390    -0.4552     0.2567     1.0000
7
8 U =
9     0.9134     0.5469     0.9706     0.1419
10         0    -0.4448     0.0024     0.3447
11         0         0     0.0925     0.9426
12         0         0         0     0.6955
13
14 P =
15         0         0         0         1
16         0         1         0         0
17         1         0         0         0
18         0         0         1         0
```

Pivoting with LU Factorization

- This is how you would explicitly use the factorizations

$$P^{-1}LUx = b \implies x = U^{-1}L^{-1}Pb$$

```
1 >> x=U\(L\(P*b))  
2 x = 0.5326 0.5416 -1.2765 1.1315  
3 >> A\b  
4      0.5326 0.5416 -1.2765 1.1315
```

- Here is another way, use the function `decomposition()`

```
1 >> dA = decomposition(A,'lu');  
2 >> x = dA\b
```



LU cost

LU factorization requires partial pivoting but the extra cost is only $\mathcal{O}(n^2)$. The total cost is still

$$\frac{2}{3}n^3 + \mathcal{O}(n^2)$$

- Memory Cost: although P is a matrix it can be compressed to a vector
- Example:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

- Permutation just adds $\mathcal{O}(n)$ extra memory



Pivoting for LDL^T Factorization

Question

What is the cost of pivoting for LDL^T ?

Symmetric Pivoting

Instead of swapping only rows, we apply the same permutation to **both rows and columns**:

$$PAP^T = LDL^T$$

- Symmetry is preserved: $(PAP^T)^T = PAP^T$
- Symmetric pivoting costs $\mathcal{O}(n^2)$

Key Idea

LDL requires a slightly more expensive form of pivoting to maintain symmetry of the matrix but it is still $\mathcal{O}(n^2)$.

Pivoting for Cholesky Factorization

- Cholesky applies only to **symmetric positive definite (SPD)** matrices
- Recall the factorization:

$$A = \tilde{L}\tilde{L}^T$$

- Unlike LU and LDL, **no pivoting is required**

$$a_{kk} - \sum_{j=1}^{k-1} \ell_{kj}^2 > 0$$

- For SPD matrices, every Cholesky pivot is positive in exact arithmetic.
- The SPD structure allows Cholesky to operate stably without pivoting.
- An ill-conditioned matrix can still lead to an inaccurate computed solution.

Key Idea

Cholesky factorization does not require pivoting due to the structure of SPD matrices.

LU vs Cholesky

```
1 given A, b
2
3 for k = 1...n-1
4     for i = k+1...n
5         xmult = aik/akk
6         aik = xmult
7         for j = k+1...n
8             aij = aij - (xmult)akj
9         end
10    end
11 end
```

LU (no pivoting)

```
1 given A
2 output L
3
4 for k = 1...n
5      $\ell_{kk} = \left( a_{kk} - \sum_{s=1}^{k-1} \ell_{ks}^2 \right)^{1/2}$ 
6     for i = k+1...n
7          $\ell_{ik} = \left( a_{ik} - \sum_{s=1}^{k-1} \ell_{is} \ell_{ks} \right) / \ell_{kk}$ 
8     end
9 end
```

Cholesky

- LU costs $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ and needs pivoting
- Cholesky costs $\frac{1}{3}n^3 + \mathcal{O}(n^2)$ and does **not** need pivoting



- Use Matlab's `chol` function
- Cholesky has two main advantages:
 - Save 50% of the flops and many memory operations with Cholesky
 - No pivoting required

See `time_LU_vs_Cholesky.m`



LU, LDL, Cholesky

**MATRIX IS
INVERTIBLE**



**AND IT'S
SYMMETRIC**



IT'S SPD

