Math 6610: Analysis of Numerical Methods, I
The LU and Cholesky decompositions

Department of Mathematics, University of Utah

Fall 2025

Resources: Trefethen and Bau 1997, Lectures 20, 21, 23
Atkinson 1989, Chapter 1
Salgado and Wise 2022, Chapter 3

Gaussian elimination

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be an invertible matrix, and let $\boldsymbol{b} \in \mathbb{C}^n$ be any vector.

Our goal is to compute the solution $\boldsymbol{x} \in \mathbb{C}^n$ to the linear system,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Gaussian elimination

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be an invertible matrix, and let $\boldsymbol{b} \in \mathbb{C}^n$ be any vector.

Our goal is to compute the solution $\boldsymbol{x} \in \mathbb{C}^n$ to the linear system,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

One "standard" way to do this starts by forming the *augmented* rectangular matrix

$$(\boldsymbol{A} \;\; \boldsymbol{b}) \in \mathbb{C}^{n \times (n+1)},$$

and proceeds to perform elimination steps to transform the left $n \times n$ block into the identity matrix.

$$\begin{pmatrix} \underline{\underline{A}} & \underline{b} \end{pmatrix} \xrightarrow{\text{row ops}} \begin{pmatrix} \quad & \tilde{\underline{b}} \end{pmatrix} \xrightarrow{\text{back-sub.}} \begin{pmatrix} \underline{\underline{I}} & \underline{x} \end{pmatrix}$$

Row operations, I

If we record the row operations needed to perform Gaussian elimination, then we can work *only* on the matrix $\boldsymbol{A}$.

Consider a matrix $\boldsymbol{A}$ with columns $(\boldsymbol{a}_j)_{j=1}^n$:

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \\ | & | & & | \end{pmatrix}, \qquad \boldsymbol{a}_j = \begin{pmatrix} a_{j,1} \\ a_{j,2} \\ \vdots \\ a_{j,n} \end{pmatrix}$$

$$\begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{n,1} \\ a_{1,2} & a_{2,2} & & \\ \vdots & & \ddots & \\ a_{1,n} & & & a_{n,n} \end{pmatrix} \xrightarrow[\substack{\vdots \\ r_j \leftarrow r_j - \frac{a_{1,j}}{a_{1,1}} r_1}]{\substack{r1 \leftarrow r1 \\ r2 \leftarrow r2 - \frac{a_{1,2}}{a_{1,1}} r1}} \begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{n,1} \\ 0 & \text{---} & r_2^* & \text{---} \\ 0 & & \vdots & \\ \vdots & & & \\ 0 & \text{---} & r_n^* & \text{---} \end{pmatrix}$$

Row operations, I

If we record the row operations needed to perform Gaussian elimination, then we can work *only* on the matrix $\boldsymbol{A}$.

Consider a matrix $\boldsymbol{A}$ with columns $(\boldsymbol{a}_j)_{j=1}^n$:

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \\ | & | & & | \end{pmatrix}, \qquad \boldsymbol{a}_j = \begin{pmatrix} a_{j,1} \\ a_{j,2} \\ \vdots \\ a_{j,n} \end{pmatrix}$$

If $a_{1,1} \neq 0$, then by standard Gaussian elimination, we replace row j with itself minus a scaled version of row 1 to eliminate entries in column 1.

I.e., if $\boldsymbol{r}_j^*$ is row j of $\boldsymbol{A}$, then for $j > 1$, replace row j with,

$$r_1^* = \tilde{r}_1^*$$

$$\tilde{\boldsymbol{r}}_j^* = \boldsymbol{r}_j^* - \frac{\overset{a_{1,j}}{\cancel{a_{j,1}}}}{a_{1,1}} \boldsymbol{r}_1^* \qquad r_j^* = \tilde{r}_j^* + \frac{a_{1,j}}{a_{1,1}} \tilde{r}_1^*$$

In particular, this shows that $\boldsymbol{r}_j$ can be reconstructed in terms of $\tilde{\boldsymbol{r}}_j$ and $\boldsymbol{r}_1$. or $\tilde{r}_1$

Row operations, II

After row operations that transform the first column to a multiple of e_1, we have

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{A}_2, \qquad \boldsymbol{L}_1 = \begin{pmatrix} | & - & \boldsymbol{0}_{1\times(n-1)} & - \\ \boldsymbol{\ell} & & \boldsymbol{I}_{(n-1)\times(n-1)} & \\ | & & & \end{pmatrix},$$

with $\boldsymbol{A}_2$ the matrix

$$\boldsymbol{A}_2 = \begin{pmatrix} a_{1,1} & | & & | \\ 0 & & & \\ \vdots & \boldsymbol{a}_2^{(2)} & \cdots & \boldsymbol{a}_n^{(2)} \\ 0 & | & & | \end{pmatrix}.$$

Often: $\underline{\underline{L}}_1$ is called an "elementary matrix"

after 1 iteration of GE.

$$\underline{\ell} = \begin{pmatrix} 1 \\ \frac{a_{1,2}}{a_{1,1}} \\ \frac{a_{1,3}}{a_{1,1}} \\ \vdots \\ \frac{a_{1,n}}{a_{1,1}} \end{pmatrix}$$

If we continue triangular elimination from $\boldsymbol{A}_2$, until the last column we obtain,

$$\boldsymbol{A} = \boldsymbol{L}_1 \cdots \boldsymbol{L}_{n-1} \boldsymbol{A}_n,$$

A_n: upper triangular.

where $\boldsymbol{A}_n$ is an upper-triangular matrix, and each $\boldsymbol{L}_j$ has the form,

$$\boldsymbol{L}_j = \begin{pmatrix} | & & | & | & | & & | \\ \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_{j-1} & \boldsymbol{\ell}_j & \boldsymbol{e}_{j+1} & \cdots & \boldsymbol{e}_n \\ | & & | & | & | & & | \end{pmatrix},$$

$$\underline{\ell} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \times \\ \times \\ \vdots \\ \times \end{pmatrix} \begin{array}{l} \left.\vphantom{\begin{matrix}0\\0\\\vdots\\0\end{matrix}}\right\} j \text{ entries} \\ \left.\vphantom{\begin{matrix}\times\\\times\\\vdots\\\times\end{matrix}}\right\} \begin{matrix} n-j \\ \text{entries} \\ \text{non-zero} \end{matrix} \end{array}$$

where $\boldsymbol{\ell}_j$ is a vector with jth component $\ell_{j,j} = 1$, and $\ell_{j,k} = 0$ for $k < j$.

For step j of elimination: leave rows $1, 2 \ldots j$ alone
rows $j+1 \ldots n$ suffer elimination

Row operations, III

If we continue triangular elimination from $\boldsymbol{A}_2$, until the last column we obtain,

$$\boldsymbol{A} = \boldsymbol{L}_1 \cdots \boldsymbol{L}_{n-1} \boldsymbol{A}_n,$$

where $\boldsymbol{A}_n$ is an upper-triangular matrix, and each $\boldsymbol{L}_j$ has the form,

$$\boldsymbol{L}_j = \begin{pmatrix} | & & | & | & | & & | \\ \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_{j-1} & \boldsymbol{\ell}_j & \boldsymbol{e}_{j+1} & \cdots & \boldsymbol{e}_n \\ | & & | & | & | & & | \end{pmatrix},$$

where $\boldsymbol{\ell}_j$ is a vector with jth component $\ell_{j,j} = 1$, and $\ell_{j,k} = 0$ for $k < j$. Note that each $\boldsymbol{L}_j$ is lower triangular, and one can show that

$$\boldsymbol{L}_j \boldsymbol{L}_{j+1} = \begin{pmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_{j-1} & \boldsymbol{\ell}_j & \boldsymbol{\ell}_{j+1} & \boldsymbol{e}_{j+2} & \cdots & \boldsymbol{e}_n \end{pmatrix},$$

so that $\boldsymbol{L} := \prod_{j=1}^{n-1} \boldsymbol{L}_j$ is also ~~upper~~-triangular.

lower

The LU factorization

We have just shown that, if all our elimination steps successfully complete, then

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U},$$

where $\boldsymbol{L}$ is lower-triangular, and $\boldsymbol{U}$ is upper-triangular.

has pivot entries on diagonal.

L has 1's on the diagonal

The LU factorization

We have just shown that, if all our elimination steps successfully complete, then

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U},$$

where $\boldsymbol{L}$ is lower-triangular, and $\boldsymbol{U}$ is upper-triangular.

How can the steps fail?

Theorem

$\boldsymbol{A}$ has an LU decomposition if and only if $\det \boldsymbol{A}_j \neq 0$ for all $j = 1, \ldots, n$, where $\boldsymbol{A}_j$ is the principal (upper-left) $j \times j$ submatrix of $\boldsymbol{A}$.

$$\begin{pmatrix} x & x & x & x \\ x & x & d & x \\ x & x & x & x \\ x & x & a & x \end{pmatrix}$$

E.g. $j=3$: $\begin{pmatrix} x & x & x \\ & x & x \\ & & x \end{pmatrix}$ (row-reduced A_3)

$=0$ iff A_3 isn't invertible

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

At step j of GE, consider $j \times j$ principal submatrix of $\underline{\underline{A}}$:

$$\underline{\underline{A}} \xrightarrow{\text{step } j} \overbrace{\begin{pmatrix} x & x & x & \cdots & x \\ 0 & x & x & \cdots & x \\ 0 & 0 & x & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & x \end{pmatrix}}^{U_j}$$

col j

$$\underline{\underline{A}} = \begin{pmatrix} \boxed{A_j} & \\ & \end{pmatrix}$$

$A_j = L_j U_j$

L_j invertible.

Note: $(U_j)_{k,k} \neq 0$ for $k < j$ because GE was successful previously.

$\det U_j = 0$ iff $(U_j)_{j,j} = 0$

Also $\det U_j = 0$ iff $\det A_j = 0$.

LU factorization utility

The LU factorization/decomposition has several uses;

- It's how we solve linear systems (even on computers!)
- If an LU factorization for $\boldsymbol{A}$ is available, then solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ requires only $\mathcal{O}(n^2)$ operations.
- $\det \boldsymbol{A} = (\det \boldsymbol{L})(\det \boldsymbol{U})$.

$LUx = b$

$x = \underbrace{U^{-1}\underbrace{L^{-1}b}_{\text{forward subst., } O(n^2)}}_{\text{back subst. } O(n^2)}$

Generic computation of $A = LU$ requires $O(n^3)$ effort.

LU factorization utility

The LU factorization/decomposition has several uses;

- It's how we solve linear systems
- If an LU factorization for $\boldsymbol{A}$ is available, then solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ requires only $\mathcal{O}(n^2)$ operations.
- $\det \boldsymbol{A} = \det \boldsymbol{L} \det \boldsymbol{U}$.

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be an invertible matrix. If Gaussian elimination succeeds, then

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U},$$

where $\boldsymbol{L}$ and $\boldsymbol{U}$ are lower- and upper-triangular, respectively.

LU factorization utility

The LU factorization/decomposition has several uses;

- It's how we solve linear systems
- If an LU factorization for $\boldsymbol{A}$ is available, then solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ requires only $\mathcal{O}(n^2)$ operations.
- $\det \boldsymbol{A} = \det \boldsymbol{L} \det \boldsymbol{U}$.

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be an invertible matrix. If Gaussian elimination succeeds, then

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U},$$

where $\boldsymbol{L}$ and $\boldsymbol{U}$ are lower- and upper-triangular, respectively.

"Standard" Gaussian elimination fails in some cases, e.g., with

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

"Solution": interchange rows

Pivoting (interchanging rows)

The standard approach to "fixing" this problem is pivoting, which interchanges rows and/or columns.

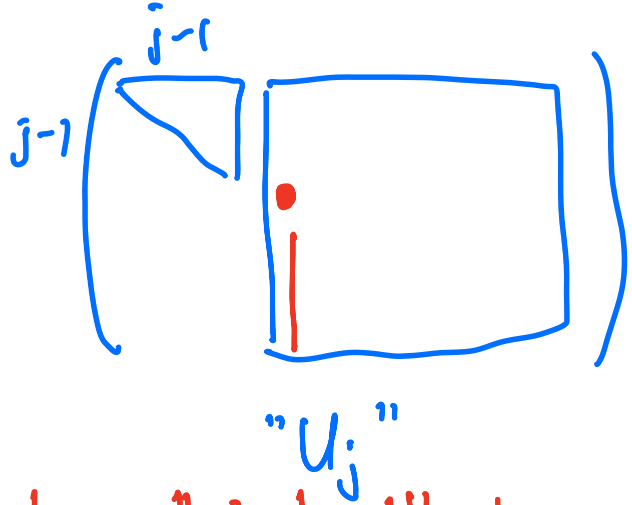
We know pivoting by another name: permutations.

The standard approach to "fixing" this problem is pivoting, which interchanges rows and/or columns.

We know pivoting by another name: permutations.

General pivoting strategy: permute rows so that diagonal elements during elimination are non-zero.
(For stability, pivot so that diagonal elements have maximum magnitude.)

Iteration j:

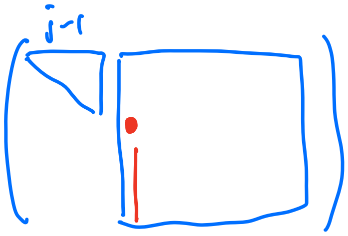


•: (j,j) entry, "standard" choice of pivot.

During elimination: divide by pivot entry.

Goal: Divide by a number that's as large as possible

Iteration j:

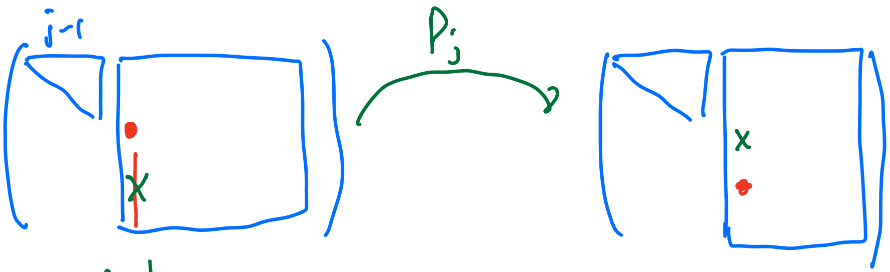


Interchange rows so (j,j) entry has max. magnitude.
(Don't destroy triangular structure!)

$$\text{column } j: \quad \begin{pmatrix} x\\ x\\ \vdots\\ x \\ y \\ y \\ \vdots \\ y\end{pmatrix}\begin{matrix}\left.\vphantom{\begin{matrix}x\\x\\\vdots\\x\end{matrix}}\right\} j-1 \text{ entries}\\ \left.\vphantom{\begin{matrix}y\\y\\\vdots\\y\end{matrix}}\right\}(n-j+1)\text{ entries}\end{matrix}$$

Find element in last $n-j+1$ entries w/ max magnitude, and permute it to row j.

Iteration j:



x: max magnitude
P_j: permutation matrix interchanges row w/ green x and row j

Pivoting

The standard approach to "fixing" this problem is pivoting, which interchanges rows and/or columns.

We know pivoting by another name: permutations.

General pivoting strategy: permute rows so that diagonal elements during elimination are non-zero.
(For stability, pivot so that diagonal elements have maximum magnitude.)

This results in the decomposition,

$$\boldsymbol{A} = \boldsymbol{P}_1 \boldsymbol{L}_1 \ \boldsymbol{P}_2 \boldsymbol{L}_2 \ \cdots \ \boldsymbol{P}_{n-1} \boldsymbol{L}_{n-1} \boldsymbol{U},$$

where $\boldsymbol{P}_j$ is a permutation matrix that permutes row j with row k for some $k \geqslant j$.

The standard approach to "fixing" this problem is pivoting, which interchanges rows and/or columns.

We know pivoting by another name: permutations.

General pivoting strategy: permute rows so that diagonal elements during elimination are non-zero.
(For stability, pivot so that diagonal elements have maximum magnitude.)

This results in the decomposition,

$$\boldsymbol{A} = \boldsymbol{P}_1 \boldsymbol{L}_1 \, \boldsymbol{P}_2 \boldsymbol{L}_2 \, \cdots \, \boldsymbol{P}_{n-1} \boldsymbol{L}_{n-1} \boldsymbol{U},$$

where $\boldsymbol{P}_j$ is a permutation matrix that permutes row j with row k for some $k \geqslant j$. One can show that $\boldsymbol{L}_j \boldsymbol{P}_k = \boldsymbol{P}_k \widetilde{\boldsymbol{L}}_j$ if $j < k$ for some other lower-triangular matrix $\widetilde{\boldsymbol{L}}_j$, so that

$$A = \underbrace{\left(\prod_{j=1}^{n-1} \boldsymbol{P}_j \right)}_{\text{a permutation}} \underbrace{\left(\prod_{j=1}^{n-1} \widetilde{\boldsymbol{L}}_j \right)}_{\text{lower triangular.}} \boldsymbol{U}.$$

In fact, we can show that this *row pivoting* strategy always works.

Theorem

If $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is invertible, then there exists

- *a permutation matrix $\boldsymbol{P}$,*
- *a lower-triangular matrix $\boldsymbol{L}$,*
- *an upper-triangular matrix $\boldsymbol{U}$,*

such that

$$\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$$

$$\begin{pmatrix} x \\ x \\ \vdots \\ x \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}x\\x\\ \vdots\\x\end{matrix}}\right\} j-1 \\ \left.\vphantom{\begin{matrix}0\\0\\ \vdots\\0\end{matrix}}\right\} n-j+1 \end{matrix}$$

This particular pivoting strategy is "partial" or "row" pivoting.

More pivoting

Row pivoting is not the only option.

For example, *full* pivoting permutes *both* lower rows and rightmost columns in search of a maximum-magnitude pivot.

$$\boldsymbol{A} = \boldsymbol{P}_1 \boldsymbol{L}_1 \, \boldsymbol{P}_2 \boldsymbol{L}_2 \, \cdots \, \boldsymbol{P}_{n-1} \boldsymbol{L}_{n-1} \, \boldsymbol{U} \, \boldsymbol{Q}_{n-1} \boldsymbol{Q}_{n-2} \cdots \boldsymbol{Q}_1,$$

where both $\boldsymbol{P}_j$ and $\boldsymbol{Q}_j$ are permutation matrices.

More pivoting

Row pivoting is not the only option.

For example, *full* pivoting permutes *both* lower rows and rightmost columns in search of a maximum-magnitude pivot.

$$\boldsymbol{A} = \boldsymbol{P}_1 \boldsymbol{L}_1 \, \boldsymbol{P}_2 \boldsymbol{L}_2 \, \cdots \, \boldsymbol{P}_{n-1} \boldsymbol{L}_{n-1} \, \boldsymbol{U} \, \boldsymbol{Q}_{n-1} \boldsymbol{Q}_{n-2} \cdots \boldsymbol{Q}_1,$$

where both $\boldsymbol{P}_j$ and $\boldsymbol{Q}_j$ are permutation matrices.

This achieves the full-pivoted LU decomposition,

$$\boldsymbol{P} \boldsymbol{A} \boldsymbol{Q} = \boldsymbol{L} \boldsymbol{U}.$$

More pivoting

Row pivoting is not the only option.

For example, *full* pivoting permutes *both* lower rows and rightmost columns in search of a maximum-magnitude pivot.

$$\boldsymbol{A} = \boldsymbol{P}_1 \boldsymbol{L}_1 \, \boldsymbol{P}_2 \boldsymbol{L}_2 \, \cdots \, \boldsymbol{P}_{n-1} \boldsymbol{L}_{n-1} \, \boldsymbol{U} \, \boldsymbol{Q}_{n-1} \boldsymbol{Q}_{n-2} \cdots \boldsymbol{Q}_1,$$

where both $\boldsymbol{P}_j$ and $\boldsymbol{Q}_j$ are permutation matrices.

This achieves the full-pivoted LU decomposition,

$$\boldsymbol{P} \boldsymbol{A} \boldsymbol{Q} = \boldsymbol{L} \boldsymbol{U}.$$

An alternative is *rook pivoting*, which performs a permutation similar to the above, except that at elimination step j, the maximum is sought *only* over row j and column j.

More pivoting

Row pivoting is not the only option.

For example, *full* pivoting permutes *both* lower rows and rightmost columns in search of a maximum-magnitude pivot.

$$\boldsymbol{A} = \boldsymbol{P}_1 \boldsymbol{L}_1 \, \boldsymbol{P}_2 \boldsymbol{L}_2 \, \cdots \, \boldsymbol{P}_{n-1} \boldsymbol{L}_{n-1} \, \boldsymbol{U} \, \boldsymbol{Q}_{n-1} \boldsymbol{Q}_{n-2} \cdots \boldsymbol{Q}_1,$$

where both $\boldsymbol{P}_j$ and $\boldsymbol{Q}_j$ are permutation matrices.

This achieves the full-pivoted LU decomposition,

$$\boldsymbol{P}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{U}.$$

An alternative is *rook pivoting*, which performs a permutation similar to the above, except that at elimination step j, the maximum is sought *only* over row j and column j. All flavors of LU factorizations require $\mathcal{O}(n^3)$ complexity with explicit, small multiplying constant.

But the choice of pivoting can substantially affect the actual runtime (the constant).

Hermitian positive-definite matrices

Assume $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is Hermitian positive definite.

Our investigation of LU decompositions specializes considerably in this case.

Hermitian positive-definite matrices

Assume $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is Hermitian positive definite.

Our investigation of LU decompositions specializes considerably in this case.

First we note some properties of $\boldsymbol{A}$:

- $\boldsymbol{A}$ is invertible
- The diagonal entries of $\boldsymbol{A}$ are real and strictly positive
- If $\boldsymbol{B} \in \mathbb{C}^{m \times n}$ with $m \leqslant n$ is of full rank, then $\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^*$ is positive-definite

LU on positive-definite matrices

A general positive-definite matrix $\boldsymbol{A}$ has the form

$$\boldsymbol{A} = \begin{pmatrix} a & - & \boldsymbol{v}^* & - \\ | & & & \\ \boldsymbol{v} & & \boldsymbol{A}_2 & \\ | & & & \end{pmatrix}.$$

Consider performing elimination on $\boldsymbol{A}$:

LU on positive-definite matrices

A general positive-definite matrix $\boldsymbol{A}$ has the form

$$\boldsymbol{A} = \begin{pmatrix} a & - & \boldsymbol{v}^* & - \\ | & & & \\ \boldsymbol{v} & & \boldsymbol{A}_2 & \\ | & & & \end{pmatrix}.$$

Consider performing elimination on $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{B}^* = \begin{pmatrix} 1 & - & 0 & - \\ | & & & \\ \frac{\boldsymbol{v}}{a} & I & & \\ | & & & \end{pmatrix} \begin{pmatrix} a & - & \boldsymbol{v}^* & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix}$$

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{B}^*$$

We can perform a single step of Gaussian elimination on $\boldsymbol{B}$:

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{B}^*$$

We can perform a single step of Gaussian elimination on $\boldsymbol{B}$:

$$\boldsymbol{B} = \boldsymbol{L}_1 \begin{pmatrix} a & - & 0 & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix},$$

i.e.,

$$\boldsymbol{A} = \boldsymbol{L}_1 \begin{pmatrix} a & - & 0 & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix} L_1^* = \widetilde{\boldsymbol{L}}_1 \begin{pmatrix} 1 & - & 0 & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix} \widetilde{\boldsymbol{L}}_1^*.$$

$$\boldsymbol{A} = \widetilde{\boldsymbol{L}}_1 \begin{pmatrix} 1 & - & 0 & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix} \widetilde{\boldsymbol{L}}_1^*.$$

Note that $\boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a}$ must be positive definite since $\widetilde{\boldsymbol{L}}_1$ is invertible.

$$\boldsymbol{A} = \widetilde{\boldsymbol{L}}_1 \begin{pmatrix} 1 & - & 0 & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix} \widetilde{\boldsymbol{L}}_1^*.$$

Note that $\boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a}$ must be positive definite since $\widetilde{\boldsymbol{L}}_1$ is invertible.

Thus, we can repeat this process:

$$\begin{aligned} \boldsymbol{A} &= \left(\widetilde{\boldsymbol{L}}_1 \widetilde{\boldsymbol{L}}_2 \cdots \widetilde{\boldsymbol{L}}_{n-1}\right) \left(\widetilde{\boldsymbol{L}}_1 \widetilde{\boldsymbol{L}}_2 \cdots \widetilde{\boldsymbol{L}}_{n-1}\right)^* \\ &=: \boldsymbol{L}\boldsymbol{L}^*. \end{aligned}$$

$$\boldsymbol{A} = \widetilde{\boldsymbol{L}}_1 \begin{pmatrix} 1 & - & 0 & - \\ | & & & \\ 0 & & \boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a} & \\ | & & & \end{pmatrix} \widetilde{\boldsymbol{L}}_1^*.$$

Note that $\boldsymbol{A}_2 - \frac{\boldsymbol{v}\boldsymbol{v}^*}{a}$ must be positive definite since $\widetilde{\boldsymbol{L}}_1$ is invertible.

Thus, we can repeat this process:

$$\begin{aligned} \boldsymbol{A} &= \left(\widetilde{\boldsymbol{L}}_1 \widetilde{\boldsymbol{L}}_2 \cdots \widetilde{\boldsymbol{L}}_{n-1}\right)\left(\widetilde{\boldsymbol{L}}_1 \widetilde{\boldsymbol{L}}_2 \cdots \widetilde{\boldsymbol{L}}_{n-1}\right)^* \\ &=: \boldsymbol{L}\boldsymbol{L}^*. \end{aligned}$$

Theorem

Every Hermitian positive definite matrix $\boldsymbol{A}$ has a unique symmetric LU, or Cholesky, decomposition: $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^*$, where $\boldsymbol{L}$ is lower-triangular and invertible.

Pivoted Cholesky

One can perform symmetric pivoting on a Hermitian positive-definite matrix $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{L}^*\boldsymbol{P}^*$.

This could be used to pivot maximum-magnitude diagonal entries to the front.

Pivoted Cholesky

One can perform symmetric pivoting on a Hermitian positive-definite matrix $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{L}^*\boldsymbol{P}^*$.

This could be used to pivot maximum-magnitude diagonal entries to the front.

However, pivoted Cholesky decompositions have another use:

Theorem

Every Hermitian positive semi-definite matrix $\boldsymbol{A}$ has a pivoted Cholesky decomposition: $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{L}^*\boldsymbol{P}^*$, where $\boldsymbol{L}$ is lower-triangular but need not invertible. This decomposition is in general not unique.

Pivoted Cholesky

One can perform symmetric pivoting on a Hermitian positive-definite matrix $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{L}^*\boldsymbol{P}^*$.

This could be used to pivot maximum-magnitude diagonal entries to the front.

However, pivoted Cholesky decompositions have another use:

Theorem

Every Hermitian positive semi-definite matrix $\boldsymbol{A}$ has a pivoted Cholesky decomposition: $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{L}^*\boldsymbol{P}^*$, where $\boldsymbol{L}$ is lower-triangular but need not invertible. This decomposition is in general not unique.

Why do we care about Cholesky decompositions? For positive-definite matrices:

- Not having to deal with pivoting is of considerable computational savings (but doesn't change asymptotic complexity)
- The Cholesky decomposition provides a "whitening" transform, e.g., for $\boldsymbol{x} \mapsto \boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x}$.
- Low-rank updates of Cholesky factors are (very) useful.

$\boldsymbol{LDU}$ and $\boldsymbol{LDL}^*$ decompositions

A minor generalization of LU for a generic matrix: If $\boldsymbol{A}$ is invertible, then we can always write,

$$\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U},$$

where $\boldsymbol{P}$ is a permutation matrix.

By construction:

- The diagonal of $\boldsymbol{L}$ is all ones
- The diagonal of $\boldsymbol{U}$ contains the non-zero pivot entries

$\boldsymbol{LDU}$ and $\boldsymbol{LDL}^*$ decompositions

A minor generalization of LU for a generic matrix: If $\boldsymbol{A}$ is invertible, then we can always write,

$$\boldsymbol{PA} = \boldsymbol{LU},$$

where $\boldsymbol{P}$ is a permutation matrix.

By construction:

- The diagonal of $\boldsymbol{L}$ is all ones
- The diagonal of $\boldsymbol{U}$ contains the non-zero pivot entries

Let $\boldsymbol{D}$ be a diagonal matrix with the pivot entries of $\boldsymbol{U}$, then we can write,

$$\boldsymbol{PA} = \boldsymbol{LD}\widetilde{\boldsymbol{U}},$$

where both $\boldsymbol{L}$ and $\boldsymbol{U}$ have ones on the diagonal. This is the (pivoted) LDU decomposition of $\boldsymbol{A}$. (There are some niche cases when doing this decomposition of $\boldsymbol{A}$ is slightly preferable.)

$\boldsymbol{LDU}$ and $\boldsymbol{LDL^*}$ decompositions

A minor generalization of LU for a generic matrix: If $\boldsymbol{A}$ is invertible, then we can always write,

$$\boldsymbol{PA} = \boldsymbol{LU},$$

where $\boldsymbol{P}$ is a permutation matrix.

By construction:

- The diagonal of $\boldsymbol{L}$ is all ones
- The diagonal of $\boldsymbol{U}$ contains the non-zero pivot entries

Let $\boldsymbol{D}$ be a diagonal matrix with the pivot entries of $\boldsymbol{U}$, then we can write,

$$\boldsymbol{PA} = \boldsymbol{LD}\widetilde{\boldsymbol{U}},$$

where both $\boldsymbol{L}$ and $\boldsymbol{U}$ have ones on the diagonal. This is the (pivoted) LDU decomposition of $\boldsymbol{A}$.
(There are some niche cases when doing this decomposition of $\boldsymbol{A}$ is slightly preferable.)

For Hermitian positive semi-definite matrices, we can write

$$\boldsymbol{PAP}^* = \boldsymbol{LDL}^*$$

(Or without permutations if $\boldsymbol{A}$ is positive definite.) This is the $\boldsymbol{LDL}^T$ decomposition of $\boldsymbol{A}$.

References I

Atkinson, Kendall (1989). *An Introduction to Numerical Analysis*. New York: Wiley. ISBN: 978-0-471-62489-9.

Salgado, Abner J. and Steven M. Wise (2022). *Classical Numerical Analysis: A Comprehensive Course*. Cambridge: Cambridge University Press. ISBN: 978-1-108-83770-5. DOI: 10.1017/9781108942607.

Trefethen, Lloyd N. and David Bau (1997). *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics. ISBN: 0-89871-361-7.