MDI210 Numerical analysis and continuous optimisation

Week 1 - Linear systems

Peter Brown

Télécom Paris
peter.brown@telecom-paris.fr

September 13, 2022

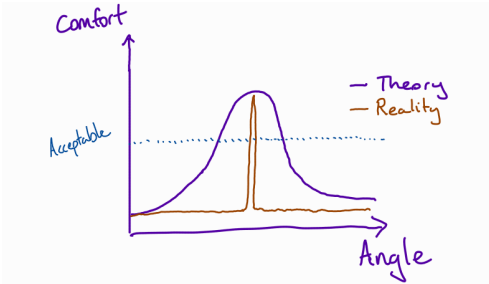
Introduction

We optimise all the time…

Introduction

We optimise all the time…

E.g., shower temperature:

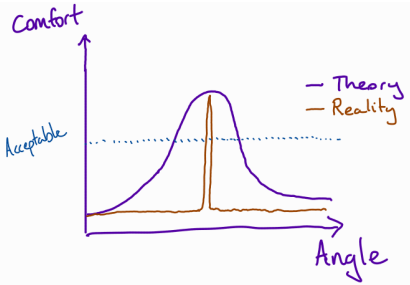


Introduction

We optimise all the time…

E.g., shower temperature:

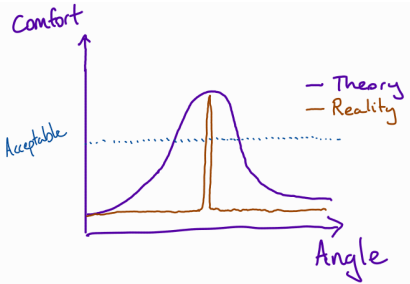
- How do we optimize?
- Efficient algorithms?
- Guaranteed optima?



Introduction

We optimise all the time…

E.g., shower temperature:



- How do we optimize?
- Efficient algorithms?
- Guaranteed optima?

Course outline:

- Linear systems / eigenvalues / linear programming
- Optimization of continuous functions / constraints / convex functions

Notation

Given an $m \times n$ matrix $A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ we define

- *Transpose*: $A^t = (a_{ji})$.
- *Adjoint*: $A^* = (\overline{a_{ji}})$ (where $\overline{x}$ is the complex conjugate of x)

Notation

Given an $m \times n$ matrix $A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ we define

- *Transpose*: $A^t = (a_{ji})$.
- *Adjoint*: $A^* = (\overline{a_{ji}})$ (where $\overline{x}$ is the complex conjugate of x)

A real/complex square matrix A is:

- *symmetric* if $A^t = A$
- *normal* if $A^t A = A A^t$
- *orthogonal* if $A^t A = A A^t = \mathbb{I}$

Hermitian if $A^* = A$

normal if $A^* A = A A^*$

unitary if $A^* A = A A^* = \mathbb{I}$

Notation

Given an $m \times n$ matrix $A = (a_{ij})_{\substack{1 \le i \le m \\ 1 \le j \le n}}$ we define

- *Transpose*: $A^t = (a_{ji})$.
- *Adjoint*: $A^* = (\overline{a_{ji}})$ (where $\overline{x}$ is the complex conjugate of x)

A real/complex square matrix A is:

- *symmetric* if $A^t = A$ — *Hermitian* if $A^* = A$
- *normal* if $A^t A = A A^t$ — *normal* if $A^* A = A A^*$
- *orthogonal* if $A^t A = A A^t = \mathbb{I}$ — *unitary* if $A^* A = A A^* = \mathbb{I}$

The *spectrum* of a matrix A is the set of its eigenvalues. The *spectral radius* is then

$$\rho(A) := \max\{|\lambda| \ : \ \lambda \in \text{spectrum}(A)\} \tag{1}$$

A motivating example

We're interested in solving linear systems given A, b such that $Ax = b$ find x.

Example

We have a linear system

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \qquad \text{Solution:} \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

A motivating example

We're interested in solving linear systems given A, b such that $Ax = b$ find x.

Example

We have a linear system

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \qquad \text{Solution:} \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

Potential numerical problems:

- Floating point errors – finite precision / real data is noisy
- Truncation errors – a useful algorithm must stop at a finite time

A motivating example

We're interested in solving linear systems given A, b such that $Ax = b$ find x.

Example

We have a linear system

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \qquad \text{Solution:} \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

Question

Suppose $Ax = b$. If $A' \approx A$, $b' \approx b$ and $A'x' = b'$ then is $x' \approx x$?

Norms and distance

Definition (Norm)

Let V be a vector space. Then $\|\cdot\| : V \to \mathbb{R}$ is a norm if

1. $\|x\| \geq 0 \qquad \forall x \in V$ (non-negative)
2. $\|x\| = 0 \iff x = 0$ (positive definite)
3. $\|\alpha x\| = |\alpha|\|x\| \quad \forall x \in V$ and scalars α. (absolute homogeneity)
4. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in V$ (triangle inequality)

Norms and distance

Definition (Norm)

Let V be a vector space. Then $\|\cdot\| : V \to \mathbb{R}$ is a norm if

1. $\|x\| \geq 0 \qquad \forall x \in V$ (non-negative)
2. $\|x\| = 0 \iff x = 0$ (positive definite)
3. $\|\alpha x\| = |\alpha|\|x\| \quad \forall x \in V$ and scalars α. (absolute homogeneity)
4. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in V$ (triangle inequality)

Example

Let $x = (x_i)_{1 \leq i \leq n}$ be a vector then

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (1-norm)
- $\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$ (2-norm / Euclidean norm)
- $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ (∞-norm / max-norm)

Norms and distance

Definition (Norm)

Let V be a vector space. Then $\|\cdot\| : V \to \mathbb{R}$ is a norm if

1. $\|x\| \geq 0 \qquad \forall x \in V$ (non-negative)
2. $\|x\| = 0 \iff x = 0$ (positive definite)
3. $\|\alpha x\| = |\alpha|\|x\| \quad \forall x \in V$ and scalars α. (absolute homogeneity)
4. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in V$ (triangle inequality)

Example

Let $x = (x_i)_{1 \leq i \leq n}$ be a vector then

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (1-norm)
- $\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$ (2-norm / Euclidean norm)
- $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ (∞-norm / max-norm)

More generally we have a p-norm for $p \in [1, \infty)$ defined as $\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$.

Norms and distance II

Norms on square matrices can be defined a vector norm (subordinate norms)

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{x:\|x\|=1} \|Ax\| = \sup_{x:0<\|x\|\leq 1} \frac{\|Ax\|}{\|x\|} \tag{3}$$

These norms satisfy additional property $\|AB\| \leq \|A\|\|B\|$ (*submultiplicative*) [Exercise: proof]

Norms and distance II

Norms on square matrices can be defined a vector norm (subordinate norms)

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{x:\|x\|=1} \|Ax\| = \sup_{x:0<\|x\|\leq 1} \frac{\|Ax\|}{\|x\|} \tag{3}$$

These norms satisfy additional property $\|AB\| \leq \|A\|\|B\|$ (*submultiplicative*) [Exercise: proof]

Example

$$\|A\|_1 = \sup_{x \neq 0} \frac{\|Ax\|_1}{\|x\|_1} = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$$

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sqrt{\rho(A^T A)} \qquad \text{(largest singular value of } A\text{)}$$

$$\|A\|_\infty = \sup_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$$

Norms and distance III

Not all matrix norms are subordinate norms

Example

A common matrix norm that is not subordinate is the *Frobenius* norm

$$\|A\|_F = \sqrt{\mathrm{Tr}\left[A^T A\right]}$$

Other examples include the *Schatten* norms $p \geq 1$

$$\|A\|_{(p)} := \left(\mathrm{Tr}\left[\left(A^T A\right)^{p/2}\right]\right)^{1/p}$$

(matrix generalization of the p-norms).

Norms and distance III

Given a norm $\|\cdot\|$ we can then define a distance between x and y (or A and B) as

$$\mathrm{dist}(x, y) = \|x - y\| \tag{4}$$

this distance measure is called a metric.

Norms and distance III

Given a norm $\|\cdot\|$ we can then define a distance between x and y (or A and B) as

$$\text{dist}(x, y) = \|x - y\| \tag{4}$$

this distance measure is called a metric.

We can use this notion of distance to talk about *convergence*, a sequence of matrices A_k is said to converge to A (denoted $\lim_{k\to\infty} A_k = A$) if

$$\lim_k \|A_k - A\| = 0 \tag{5}$$

Norms and distance III

Given a norm $\|\cdot\|$ we can then define a distance between x and y (or A and B) as

$$\text{dist}(x, y) = \|x - y\| \tag{4}$$

this distance measure is called a metric.

We can use this notion of distance to talk about *convergence*, a sequence of matrices A_k is said to converge to A (denoted $\lim_{k\to\infty} A_k = A$) if

$$\lim_k \|A_k - A\| = 0 \tag{5}$$

All norms on finite dimensional spaces are equivalent! For any two norms $\|\cdot\|_a$ and $\|\cdot\|_b$ there exist constants $c_1, c_2 > 0$ such that

$$c_1\|x\|_a \leq \|x\|_b \leq c_2\|x\|_a \qquad \forall x \tag{6}$$

Convergence doesn't depend on choice of norm!

Norms and distance IV

Example

Let A be a matrix such that $\|A\| < 1$ for a submultiplicative norm $\|\cdot\|$ then

$$\lim_k A^k = 0\,.$$

Norms and distance IV

Example

Let A be a matrix such that $\|A\| < 1$ for a submultiplicative norm $\|\cdot\|$ then

$$\lim_k A^k = 0\,.$$

This follows from

$$\|A^k\| \leq \|A\|^k \to 0 \qquad \text{as } k \to \infty$$

therefore $\lim_k A^k = 0$.

Norms and distance IV

Example

Let A be a matrix such that $\|A\| < 1$ for a submultiplicative norm $\|\cdot\|$ then

$$\lim_k A^k = 0\,.$$

This follows from

$$\|A^k\| \leq \|A\|^k \to 0 \qquad \text{as } k \to \infty$$

therefore $\lim_k A^k = 0$.

Take $A = \begin{pmatrix} a & a \\ a & a \end{pmatrix}$ for $a \in \mathbb{R}$. Then $\|A\|_1 = \|A\|_2 = \|A\|_\infty = 2|a|$ and so $A^k \to 0$ when $-1/2 < a < 1/2$.

Norms and distance IV

Example

Let A be a matrix such that $\|A\| < 1$ for a submultiplicative norm $\|\cdot\|$ then

$$\lim_k A^k = 0\,.$$

This follows from

$$\|A^k\| \leq \|A\|^k \to 0 \qquad \text{as } k \to \infty$$

therefore $\lim_k A^k = 0$.

Take $A = \begin{pmatrix} a & a \\ a & a \end{pmatrix}$ for $a \in \mathbb{R}$. Then $\|A\|_1 = \|A\|_2 = \|A\|_\infty = 2|a|$ and so $A^k \to 0$ when $-1/2 < a < 1/2$.

This agrees with

$$A^n = 2^n \begin{pmatrix} a^n & a^n \\ a^n & a^n \end{pmatrix}$$

Conditioning of linear systems

Back to the example

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \qquad \text{Solution:} \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{7}$$

Conditioning of linear systems

Back to the example

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \qquad \text{Solution:} \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{7}$$

Perturbing b we see

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{pmatrix} = \begin{pmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{pmatrix} \qquad \text{Solution:} \quad x' = \begin{pmatrix} 9.2 \\ -12.6 \\ 4.5 \\ -1.1 \end{pmatrix} \tag{8}$$

The relative error on b is $\frac{\|b'-b\|}{\|b\|}$ of order $1/100$ but the relative error on x is of order 10! (Order 1000 increase)

Conditioning of linear systems II

This large error increase is due to the *condition number* of A.

Conditioning of linear systems II

This large error increase is due to the *condition number* of A.

Let $\delta x = x' - x$ and $\delta b = b' - b$, then

$$\delta x = A^{-1}\delta b \tag{9}$$

Then for any *subordinate norm*

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\| \tag{10}$$

Conditioning of linear systems II

This large error increase is due to the *condition number* of A.

Let $\delta x = x' - x$ and $\delta b = b' - b$, then

$$\delta x = A^{-1}\delta b \tag{9}$$

Then for any *subordinate norm*

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\| \tag{10}$$

Moreover $\|b\| \leq \|A\|\|x\|$, putting these together we get

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\|\frac{\|\delta b\|}{\|b\|} \tag{11}$$

Conditioning of linear systems II

This large error increase is due to the *condition number* of A.

Let $\delta x = x' - x$ and $\delta b = b' - b$, then

$$\delta x = A^{-1}\delta b \tag{9}$$

Then for any *subordinate norm*

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\| \tag{10}$$

Moreover $\|b\| \leq \|A\|\|x\|$, putting these together we get

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\|\frac{\|\delta b\|}{\|b\|} \tag{11}$$

We call $\mathrm{cond}(A) := \|A\|\|A^{-1}\|$ the *conditioning of A* – relative error amplification depends on the conditioning!

For the example A we have $\mathrm{cond}_2(A) \approx 2974$.

Properties of the conditioning

Recall $\mathrm{cond}(A) := \|A\|\|A^{-1}\|$ for some *subordinate norm* $\|\cdot\|$.

Theorem

Let A be an invertible matrix. Then

1. $\mathrm{cond}(A) \geq 1$
2. $\mathrm{cond}(A) = \mathrm{cond}(A^{-1})$
3. *For $\alpha \neq 0$ we have* $\mathrm{cond}(\alpha A) = \mathrm{cond}(A)$.
4. *If U, V are orthogonal/unitary then* $\mathrm{cond}_2(UAV) = \mathrm{cond}(A)$
5. *If A is orthogonal/unitary then* $\mathrm{cond}_2(A) = 1$.

Properties of the conditioning

Recall $\mathrm{cond}(A) := \|A\|\|A^{-1}\|$ for some *subordinate norm* $\|\cdot\|$.

Theorem

Let A be an invertible matrix. Then

1. $\mathrm{cond}(A) \geq 1$
2. $\mathrm{cond}(A) = \mathrm{cond}(A^{-1})$
3. *For $\alpha \neq 0$ we have* $\mathrm{cond}(\alpha A) = \mathrm{cond}(A)$.
4. *If U, V are orthogonal/unitary then* $\mathrm{cond}_2(UAV) = \mathrm{cond}(A)$
5. *If A is orthogonal/unitary then* $\mathrm{cond}_2(A) = 1$.

Remark (Geometric characterization of cond_2)

Let $\theta(A)$ be the smallest angle between vectors Ax and Ay where x and y are orthonormal. Then

$$\mathrm{cond}_2(A) = \frac{1}{\tan(\theta(A)/2)} \tag{12}$$

Dealing with bad conditioning

We can improve the conditioning of our problem by *rebalancing* the matrix A.

Dealing with bad conditioning

We can improve the conditioning of our problem by *rebalancing* the matrix A.
Let D_1, D_2 be diagonal invertible matrices satisfying

$$\mathrm{cond}(D_1 A D_2) = \inf_{\Delta_1, \Delta_2 \text{invertible/diagonal}} \mathrm{cond}(\Delta_1 A \Delta_2) \tag{13}$$

Dealing with bad conditioning

We can improve the conditioning of our problem by *rebalancing* the matrix A.
Let D_1, D_2 be diagonal invertible matrices satisfying

$$\mathrm{cond}(D_1 A D_2) = \inf_{\Delta_1, \Delta_2 \text{invertible/diagonal}} \mathrm{cond}(\Delta_1 A \Delta_2) \tag{13}$$

Then solve $Ax = b$ in two steps:

1. Solve $D_1 A D_2 y = D_1 b$ (receiving y)
2. Then compute $x = D_2 y$

Dealing with bad conditioning

We can improve the conditioning of our problem by *rebalancing* the matrix A.
Let D_1, D_2 be diagonal invertible matrices satisfying

$$\mathrm{cond}(D_1 A D_2) = \inf_{\Delta_1, \Delta_2 \text{invertible/diagonal}} \mathrm{cond}(\Delta_1 A \Delta_2) \tag{13}$$

Then solve $Ax = b$ in two steps:

1. Solve $D_1 A D_2 y = D_1 b$ (receiving y)
2. Then compute $x = D_2 y$

Remark

In practice it is difficult to optimize the above so one can try to optimize the ratio

$$\frac{\max_{ij} |a'_{ij}|}{\min_{ij: a'_{ij} \neq 0} |a'_{ij}|} \qquad A' = \Delta_1 A \Delta_2$$

which can be treat as a linear program (see notes once studied LP).

Conditioning problem for eigenvalues

Question

How do the eigenvalues of a matrix A change if we perturb its entries?

Conditioning problem for eigenvalues

Question

How do the eigenvalues of a matrix A change if we perturb its entries?

Theorem

Let A be diagonalizable ($P^{-1}AP = \mathrm{diag}(\lambda_i)$). Let $\|\cdot\|$ be a matrix norm so that $\|\mathrm{diag}(\delta_i)\| = \max_i |\delta_i|$.
Then for any matrix δA we have

$$\mathrm{spectrum}(A + \delta A) \subset \bigcup_{i=1}^{n} D_i \tag{14}$$

where

$$D_i = \{z \in \mathbb{C} : |z - \lambda_i| \leq \mathrm{cond}(P)\|\delta A\|\}. \tag{15}$$

Conditioning problem for eigenvalues

Question

How do the eigenvalues of a matrix A change if we perturb its entries?

Theorem

Let A be diagonalizable ($P^{-1}AP = \mathrm{diag}(\lambda_i)$). Let $\|\cdot\|$ be a matrix norm so that $\|\mathrm{diag}(\delta_i)\| = \max_i |\delta_i|$.
Then for any matrix δA we have

$$\mathrm{spectrum}(A + \delta A) \subset \bigcup_{i=1}^{n} D_i \tag{14}$$

where

$$D_i = \{z \in \mathbb{C} : |z - \lambda_i| \leq \mathrm{cond}(P)\|\delta A\|\}\,. \tag{15}$$

We define the conditioning of A for the eigenvalue problem to be

$$\Gamma(A) = \min\{\mathrm{cond}(P) : P^{-1}AP \text{ is diagonal}\} \tag{16}$$

Solving linear systems

Problem

Let $A = (a_{ij})$ be an invertible square matrix and let x and b be column vectors such that

$$Ax = b \tag{17}$$

Solve for x.

Solving linear systems

Problem

Let $A = (a_{ij})$ be an invertible square matrix and let x and b be column vectors such that

$$Ax = b \tag{17}$$

Solve for x.

Numerical methods typically do not compute A^{-1} (we can be more efficient!)

Solving linear systems

Problem

Let $A = (a_{ij})$ be an invertible square matrix and let x and b be column vectors such that

$$Ax = b \tag{17}$$

Solve for x.

Numerical methods typically do not compute A^{-1} (we can be more efficient!)

First let's think about a special case – *triangular matrices*.

Upper triangular A

Suppose A is an upper triangular matrix. Then we have the linear equations

$$\begin{aligned} a_{1,1}x_1 + \cdots + a_{1,n-1}x_{n-1} + a_{1,n}x_n &= b_1 \\ &\cdots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\ a_{n,n}x_n &= b_n \end{aligned}$$

Upper triangular A

Suppose A is an upper triangular matrix. Then we have the linear equations

$$\begin{aligned} a_{1,1}x_1 + \cdots + a_{1,n-1}x_{n-1} + a_{1,n}x_n &= b_1 \\ &\cdots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\ a_{n,n}x_n &= b_n \end{aligned}$$

We can then use *backwards substitution*:

$$\begin{aligned} x_n &= \frac{b_n}{a_{n,n}} \\ x_{n-1} &= \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} \\ &\cdots \end{aligned}$$

Upper triangular A – efficiency

Question

How many operations needed to solve with backwards substitution?

Upper triangular A – efficiency

Question

How many operations needed to solve with backwards substitution?

Solution

$$
\begin{aligned}
a_{1,1}x_1 + \cdots + a_{1,n-1}x_{n-1} + a_{1,n}x_n &= b_1 \\
&\cdots \\
a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\
a_{n,n}x_n &= b_n
\end{aligned}
$$

We have $n(n-1)/2$ additions, $n(n-1)/2$ multiplications and n divisions.

Upper triangular A – efficiency

Question

How many operations needed to solve with backwards substitution?

Solution

$$
\begin{aligned}
a_{1,1}x_1 + \cdots + a_{1,n-1}x_{n-1} + a_{1,n}x_n &= b_1 \\
&\cdots \\
a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\
a_{n,n}x_n &= b_n
\end{aligned}
$$

We have $n(n-1)/2$ additions, $n(n-1)/2$ multiplications and n divisions.

For comparison, computing the inverse is of order $O(n^3)$ operations.

Reducing to upper triangular matrices

Triangular matrices are easy to solve. *Can we reduce everything to triangular matrices?*

Reducing to upper triangular matrices

Triangular matrices are easy to solve. *Can we reduce everything to triangular matrices?*

Yes! Using *invertible* operations.

Theorem

Let $P \in \mathbb{R}^{n \times n}$ be an invertible matrix. Then

$$\{x : Ax = b\} = \{x : PAx = Pb\} \tag{18}$$

Reducing to upper triangular matrices

Triangular matrices are easy to solve. *Can we reduce everything to triangular matrices?*

Yes! Using *invertible* operations.

Theorem

Let $P \in \mathbb{R}^{n \times n}$ be an invertible matrix. Then

$$\{x : Ax = b\} = \{x : PAx = Pb\} \tag{18}$$

Gaussian Elimination: Use a sequence of invertible operators $P_1, \ldots, P_k$ such that

$$P_n \ldots P_2 P_1 A = U \tag{19}$$

where U is upper triangular. Then solve

$$Ux = P_n \ldots P_2 P_1 b\,. \tag{20}$$

Gaussian elimination

The *elementary row operations* (all invertible) are

- Interchange any two rows.
- Multiply a row by a constant.
- Add a multiple of one row to another row.

Gaussian elimination

The *elementary row operations* (all invertible) are

- Interchange any two rows.
- Multiply a row by a constant.
- Add a multiple of one row to another row.

The algorithm:

1. For $j \in \{1, \ldots n\}$ repeat steps 2-4:
2. Choose a non-zero element (*pivot*) from column j below and including $a_{j,j}$.
3. If not row j then swap chosen elements row with row j.
4. Use linear combinations of rows to set all $a_{ij} = 0$ for $i > j$.

Gauss method – example

Consider the linear system

$$\begin{aligned} 2x_1 &+ x_2 - 3x_3 = 5 \\ 4x_1 &+ x_2 + 5x_3 = -1 \\ 10x_1 &- 7x_2 + 13x_3 = -3 \end{aligned}$$

Gauss method – example

Consider the linear system

$$\begin{aligned} 2x_1 &+ x_2 - 3x_3 = 5 \\ 4x_1 &+ x_2 + 5x_3 = -1 \\ 10x_1 &- 7x_2 + 13x_3 = -3 \end{aligned}$$

Step 1.2: Choose pivot.

Gauss method – example

Consider the linear system

$$\begin{aligned} 2x_1 &+ x_2 - 3x_3 = 5 \\ 4x_1 &+ x_2 + 5x_3 = -1 \\ 10x_1 &- 7x_2 + 13x_3 = -3 \end{aligned}$$

Step 1.3: Swap chosen pivot row so pivot on diagonal. (done)

Gauss method – example

Consider the linear system

$$\begin{array}{rcrcrcl} 2x_1 & + & x_2 & - & 3x_3 & = & 5 \\ 4x_1 & + & x_2 & + & 5x_3 & = & -1 \\ 10x_1 & - & 7x_2 & + & 13x_3 & = & -3 \end{array}$$

Step 1.4: Use row sums to remove below elements.

$$R_2 \mapsto R_2 - 2R_1 \qquad \text{and} \qquad R_3 \mapsto R_3 - 5R_1$$

Gauss method – example

Consider the linear system

$$\begin{array}{rcrcrcl} 2x_1 & + & x_2 & - & 3x_3 & = & 5 \\ 4x_1 & + & x_2 & + & 5x_3 & = & -1 \\ 10x_1 & - & 7x_2 & + & 13x_3 & = & -3 \end{array}$$

Step 1.4: Use row sums to remove below elements.

$$R_2 \mapsto R_2 - 2R_1 \qquad \text{and} \qquad R_3 \mapsto R_3 - 5R_1$$

$$\begin{array}{rcrcrcl} 2x_1 & + & x_2 & - & 3x_3 & = & 5 \\ 0 & - & x_2 & + & 11x_3 & = & -11 \\ 0 & - & 12x_2 & + & 28x_3 & = & -28 \end{array}$$

Gauss method – example

Consider the linear system

$$\begin{array}{rcrcrl} 2x_1 & + & x_2 & - & 3x_3 & = 5 \\ 0 & - & 1x_2 & + & 11x_3 & = -11 \\ 0 & - & 12x_2 & + & 28x_3 & = -28 \end{array}$$

Step 2.1: Choose pivot

Gauss method – example

Consider the linear system

$$\begin{array}{rcrcrl} 2x_1 & + & x_2 & - & 3x_3 & = 5 \\ 0 & - & 1x_2 & + & 11x_3 & = -11 \\ 0 & - & 12x_2 & + & 28x_3 & = -28 \end{array}$$

Step 2.2: Move to diagonal (done)

Gauss method – example

Consider the linear system

$$\begin{array}{rcrcrl} 2x_1 & + & x_2 & - & 3x_3 & = 5 \\ 0 & - & 1x_2 & + & 11x_3 & = -11 \\ 0 & - & 12x_2 & + & 28x_3 & = -28 \end{array}$$

Step 2.3: Remove zeros below pivot

$$R_3 \mapsto R_3 - 12R_2$$

Gauss method – example

Consider the linear system

$$\begin{array}{rcrcrcl} 2x_1 & + & x_2 & - & 3x_3 & = & 5 \\ 0 & - & 1x_2 & + & 11x_3 & = & -11 \\ 0 & - & 12x_2 & + & 28x_3 & = & -28 \end{array}$$

Step 2.3: Remove zeros below pivot

$$R_3 \mapsto R_3 - 12R_2$$

$$\begin{array}{rcrcrcl} 2x_1 & + & x_2 & - & 3x_3 & = & 5 \\ 0 & - & 1x_2 & + & 11x_3 & = & -11 \\ 0 & - & 0 & - & 104x_3 & = & 104 \end{array}$$

Gauss method – example

Now we're left with an upper triangular linear system

$$\begin{aligned} 2x_1 &+ x_2 - 3x_3 = 5 \\ 0 &- x_2 + 11x_3 = -11 \\ 0 &- 0 - 104x_3 = 104 \end{aligned}$$

which we can compute like before!

Applying backward substitution we get

$$x_3 = -1, \qquad x_2 = 0, \qquad x_1 = 1$$

Gauss method – example

Now we're left with an upper triangular linear system

$$\begin{aligned} 2x_1 &+ x_2 - 3x_3 = 5 \\ 0 &- x_2 + 11x_3 = -11 \\ 0 &- 0 - 104x_3 = 104 \end{aligned}$$

which we can compute like before!

Applying backward substitution we get

$$x_3 = -1, \qquad x_2 = 0, \qquad x_1 = 1$$

Remark

Whenever you exchange rows you flip the sign of the determinant. We can compute $\det(A)$ as

$$\det(A) = (-1)^q \times \Pi_i p_i$$

where p_i is the value of the i-th pivot.

Gauss method - choice of pivot

- *Default strategy*: Choose the term on the diagonal (or next available one).
- *Partial pivot*: Choose pivot with largest modulus in the current column.
- *Total pivot*: Choose pivot with largest modulus in the remaining residual matrix.

Choosing larger modulus reduces numerical instability due to divisions but is more expensive.

Gauss method - choice of pivot

- *Default strategy*: Choose the term on the diagonal (or next available one).
- *Partial pivot*: Choose pivot with largest modulus in the current column.
- *Total pivot*: Choose pivot with largest modulus in the remaining residual matrix.

Choosing larger modulus reduces numerical instability due to divisions but is more expensive.

Example (Choice of pivot)

Consider the system (solution $(x_1, x_2) = (1.0001, 0.9999)$)

$$\begin{aligned} 10^{-4}x_1 + \quad x_2 &= \quad 1 \\ x_1 + \quad x_2 &= \quad 2 \end{aligned}$$

After *default pivoting* we have

$$x_2 = \frac{2 - 10^4}{1 - 10^4} = \frac{9998}{9999} \approx_{\text{lose 1 digit}} 1 \implies x_1 = 0$$

Gauss method - choice of pivot

- *Default strategy*: Choose the term on the diagonal (or next available one).
- *Partial pivot*: Choose pivot with largest modulus in the current column.
- *Total pivot*: Choose pivot with largest modulus in the remaining residual matrix.

Choosing larger modulus reduces numerical instability due to divisions but is more expensive.

Example (Choice of pivot)

Consider the system (solution $(x_1, x_2) = (1.0001, 0.9999)$)

$$\begin{aligned} x_1 + \quad x_2 &= \quad 2 \\ 10^{-4} x_1 + \quad x_2 &= \quad 1 \end{aligned}$$

After *partial pivoting* we have

$$x_2 = \frac{1 - 2 \times 10^{-4}}{1 - 10^{-4}} = \frac{0.9998}{0.9999} \approx_{\text{lose 1 digit}} 1 \implies x_1 = 1$$

Gauss-Jordan method – computing inverses

We don't have to solve a single linear system

We can solve n systems at the same time and compute the inverse!

Gauss-Jordan method – computing inverses

We don't have to solve a single linear system

We can solve n systems at the same time and compute the inverse!

Example: Compute the inverse of

$$A = \begin{pmatrix} 1 & -3 & 14 \\ 1 & -2 & 10 \\ -2 & 4 & -19 \end{pmatrix}$$

Gauss-Jordan method – computing inverses

We don't have to solve a single linear system

We can solve n systems at the same time and compute the inverse!

Example: Compute the inverse of

$$A = \begin{pmatrix} 1 & -3 & 14 \\ 1 & -2 & 10 \\ -2 & 4 & -19 \end{pmatrix}$$

So we solve 3 systems simultaneously

$$\begin{array}{rcrcrcc|c|c} x_1 & - & 3x_2 & + & 14x_3 & = & 1 & 0 & 0 \\ x_1 & - & 2x_2 & + & 10x_3 & = & 0 & 1 & 0 \\ -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \end{array}$$

Gauss-Jordan method – computing inverses

So we solve 3 systems simultaneously

$$\begin{array}{rcrcrcc|c|c} x_1 & - & 3x_2 & + & 14x_3 & = & 1 & 0 & 0 \\ x_1 & - & 2x_2 & + & 10x_3 & = & 0 & 1 & 0 \\ -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \end{array}$$

Gauss-Jordan method – computing inverses

So we solve 3 systems simultaneously

$$\begin{array}{rcrcrcc|c|c} x_1 & - & 3x_2 & + & 14x_3 & = & 1 & 0 & 0 \\ x_1 & - & 2x_2 & + & 10x_3 & = & 0 & 1 & 0 \\ -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \end{array}$$

Step 1.1: Let's do a partial pivot

$$\begin{array}{rcrcrcc|c|c} -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \\ x_1 & - & 2x_2 & + & 10x_3 & = & 0 & 1 & 0 \\ x_1 & - & 3x_2 & + & 14x_3 & = & 1 & 0 & 0 \end{array}$$

Gauss-Jordan method – computing inverses

So we solve 3 systems simultaneously

$$\begin{array}{rcrcrcc|c|c} x_1 & - & 3x_2 & + & 14x_3 & = & 1 & 0 & 0 \\ x_1 & - & 2x_2 & + & 10x_3 & = & 0 & 1 & 0 \\ -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \end{array}$$

Step 1.1: Let's do a partial pivot

$$\begin{array}{rcrcrcc|c|c} -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \\ x_1 & - & 2x_2 & + & 10x_3 & = & 0 & 1 & 0 \\ x_1 & - & 3x_2 & + & 14x_3 & = & 1 & 0 & 0 \end{array}$$

Step 1.2: Remove the elements below pivot

$$\begin{array}{rcrcrcc|c|c} -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \\ & & & + & \frac{1}{2}x_3 & = & 0 & 1 & \frac{1}{2} \\ & - & x_2 & + & \frac{9}{2}x_3 & = & 1 & 0 & \frac{1}{2} \end{array}$$

$$R_2 \mapsto R_2 + \frac{1}{2}R_1 \qquad R_3 \mapsto R_3 + \frac{1}{2}R_1$$

Gauss-Jordan method – computing inverses

$$\begin{array}{rcrcrcr|c|c}
-2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \\
 & & & + & \frac{1}{2}x_3 & = & 0 & 1 & \frac{1}{2} \\
 & - & x_2 & + & \frac{9}{2}x_3 & = & 1 & 0 & \frac{1}{2}
\end{array}$$

Gauss-Jordan method – computing inverses

$$\begin{array}{rcrcrcl|c|c}
-2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \\
 & & & + & \frac{1}{2}x_3 & = & 0 & 1 & \frac{1}{2} \\
 & - & x_2 & + & \frac{9}{2}x_3 & = & 1 & 0 & \frac{1}{2}
\end{array}$$

Step 2.1: We do a total pivot (to demonstrate)

$$\begin{array}{rcrcrcl|c|c}
-2x_1 & - & 19x_3 & + & 4x_2 & = & 0 & 1 & 0 \\
 & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\
 & + & \frac{1}{2}x_3 & & & = & 0 & \frac{1}{2} & 1
\end{array}$$

Gauss-Jordan method – computing inverses

$$\begin{array}{rcrcrcl|c|c} -2x_1 & + & 4x_2 & - & 19x_3 & = & 0 & 0 & 1 \\ & & & + & \frac{1}{2}x_3 & = & 0 & 1 & \frac{1}{2} \\ & - & x_2 & + & \frac{9}{2}x_3 & = & 1 & 0 & \frac{1}{2} \end{array}$$

Step 2.1: We do a total pivot (to demonstrate)

$$\begin{array}{rcrcrcl|c|c} -2x_1 & - & 19x_3 & + & 4x_2 & = & 0 & 1 & 0 \\ & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\ & + & \frac{1}{2}x_3 & & & = & 0 & \frac{1}{2} & 1 \end{array}$$

Step 2.2: Remove coefficients above and below the pivot

$$\begin{array}{rcrcrcl|c|c} -2x_1 & & & - & \frac{2}{9}x_2 & = & \frac{38}{9} & \frac{28}{9} & 0 \\ & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

$$R_1 \mapsto R_1 + \frac{38}{9}R_2 \qquad R_3 \mapsto R_3 9 - \frac{1}{9}R_2$$

Gauss-Jordan method – computing inverses

$$\begin{array}{rcrcrcl|c|c}
-2x_1 & & & - & \frac{2}{9}x_2 & = & \frac{38}{9} & \frac{28}{9} & 0 \\
 & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\
 & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1
\end{array}$$

Gauss-Jordan method – computing inverses

$$\begin{array}{rcrcrcrcl|c|c}
-2x_1 & & & - & \frac{2}{9}x_2 & = & \frac{38}{9} & \frac{28}{9} & 0 \\
 & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\
 & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1
\end{array}$$

Step 3.1: We choose a default pivot

$$\begin{array}{rcrcrcrcl|c|c}
-2x_1 & & & - & \frac{2}{9}x_2 & = & \frac{38}{9} & \frac{28}{9} & 0 \\
 & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\
 & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1
\end{array}$$

Gauss-Jordan method – computing inverses

$$\begin{array}{rcrcrcr|c|c} -2x_1 & & & - & \frac{2}{9}x_2 & = & \frac{38}{9} & \frac{28}{9} & 0 \\ & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

Step 3.1: We choose a default pivot

$$\begin{array}{rcrcrcr|c|c} -2x_1 & & & - & \frac{2}{9}x_2 & = & \frac{38}{9} & \frac{28}{9} & 0 \\ & + & \frac{9}{2}x_3 & - & x_2 & = & 1 & \frac{1}{2} & 0 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

Step 3.2: Remove coefficients above the pivot

$$\begin{array}{rcrcrcr|c|c} -2x_1 & & & & & = & 4 & 4 & 2 \\ & + & \frac{9}{2}x_3 & & & = & 0 & \frac{9}{2} & 9 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

$$R_1 \mapsto R_1 + 2R_3 \qquad R_2 \mapsto R_2 + 9R_3$$

Gauss-Jordan method – computing inverses

Now we're almost done

$$\begin{array}{rcrcrcl|c|c} -2x_1 & & & & & = & 4 & 4 & 2 \\ & + & \frac{9}{2}x_3 & & & = & 0 & \frac{9}{2} & 9 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

Gauss-Jordan method – computing inverses

Now we're almost done

$$\begin{array}{rcrcrcl|c|c} -2x_1 & & & & & = & 4 & 4 & 2 \\ & + & \frac{9}{2}x_3 & & & = & 0 & \frac{9}{2} & 9 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

Step 4: Solution

$$\begin{array}{rcr|c|c} x_1 & = & -2 & -1 & -2 \\ x_2 & = & -1 & 9 & 4 \\ x_3 & = & 0 & 2 & 1 \end{array}$$

Gauss-Jordan method – computing inverses

Now we're almost done

$$\begin{array}{rcrcrcc|c|c} -2x_1 & & & & & = & 4 & 4 & 2 \\ & + & \frac{9}{2}x_3 & & & = & 0 & \frac{9}{2} & 9 \\ & & & + & \frac{1}{9}x_2 & = & -\frac{1}{9} & \frac{4}{9} & 1 \end{array}$$

Step 4: Solution

$$\begin{array}{rcr|r|r} x_1 & = & -2 & -1 & -2 \\ x_2 & = & -1 & 9 & 4 \\ x_3 & = & 0 & 2 & 1 \end{array}$$

and so

$$A^{-1} = \begin{pmatrix} -2 & -1 & -2 \\ -1 & 9 & 4 \\ 0 & 2 & 1 \end{pmatrix}$$

The LU decomposition

Suppose every pivot in the Gauss method was on the diagonal.

The LU decomposition

Suppose every pivot in the Gauss method was on the diagonal.

Consider the matrix P_k that does the linear combinations for column k, it takes the form

$$P_k = \begin{pmatrix} 1 & 0 & & & & & \\ 0 & 1 & 0 & & & & \\ & 0 & \ddots & \ddots & & & \\ & & \ddots & 1 & 0 & & \\ & & & -\frac{a_{k+1}}{a_k} & \ddots & & \\ & & & \vdots & & 1 & 0 \\ & & & -\frac{a_n}{a_k} & & 0 & 1 \end{pmatrix} = I - v_k e_k^T$$

I.e., an identity matrix with coefficients under the k-th diagonal.

The LU decomposition

Therefore

$$P_n \dots P_1 A = U$$

where $P_1, \dots, P_n$ are *lower triangular* and U is *upper triangular*.

The LU decomposition

Therefore

$$P_n \dots P_1 A = U$$

where $P_1, \dots, P_n$ are *lower triangular* and U is *upper triangular*.

One can show P_k^{-1} is also lower triangular , therefore

$$\begin{aligned} A &= P_1^{-1} \dots P_n^{-1} U \\ &= LU \end{aligned} \tag{21}$$

I.e., we can decompose A into the product of a lower triangular and upper triangular matrix!

The LU decomposition

Therefore

$$P_n \dots P_1 A = U$$

where $P_1, \dots, P_n$ are *lower triangular* and U is *upper triangular*.

One can show P_k^{-1} is also lower triangular , therefore

$$\begin{aligned} A &= P_1^{-1} \dots P_n^{-1} U \\ &= LU \end{aligned} \tag{21}$$

I.e., we can decompose A into the product of a lower triangular and upper triangular matrix!

Question

What if the pivots are not on the diagonal?

The LU decomposition

Therefore

$$P_n \dots P_1 A = U$$

where $P_1, \dots, P_n$ are *lower triangular* and U is *upper triangular*.

One can show P_k^{-1} is also lower triangular , therefore

$$\begin{aligned} A &= P_1^{-1} \dots P_n^{-1} U \\ &= LU \end{aligned} \tag{21}$$

I.e., we can decompose A into the product of a lower triangular and upper triangular matrix!

Question

What if the pivots are not on the diagonal?

Can rearrange rows/cols of A so all pivots on diagonal so

$$QAR = LU$$

The LU decomposition

Theorem

Let $A \in \mathbb{R}^{n \times n}$ be an invertible square matrix such that for $1 \leq k \leq n$ the submatrix

$$A_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}$$

is invertible. Then the Gauss method can be performed with all pivots on the diagonal and hence the LU decomposition exists for A.

The LU decomposition

Theorem

Let $A \in \mathbb{R}^{n \times n}$ be an invertible square matrix such that for $1 \leq k \leq n$ the submatrix

$$A_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}$$

is invertible. Then the Gauss method can be performed with all pivots on the diagonal and hence the LU decomposition exists for A.

Remark

The LU decomposition can be computed when trying to solve more than one linear system. Once you have the *LU* decomposition you need to solve two triangular systems per original system

$$Ly = b \qquad \text{then} \qquad Ux = y$$

Cholesky Decomposition

A matrix A is *positive definite* if it is symmetric and

$$x^T Ax > 0, \qquad \forall x \neq 0 \tag{21}$$

For positive definite matrices we have a more efficient LU decomposition with $L = U^T$.

Example

We have $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, then $x^T Ax = x_1^2 + (x_1 + x_2)^2 > 0$ and so A is positive definite.

Cholesky Decomposition

A matrix A is *positive definite* if it is symmetric and

$$x^T Ax > 0, \qquad \forall x \neq 0 \tag{21}$$

For positive definite matrices we have a more efficient LU decomposition with $L = U^T$.

Example

We have $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, then $x^T Ax = x_1^2 + (x_1 + x_2)^2 > 0$ and so A is positive definite.

Theorem (Cholesky decomposition)

Let $A \in \mathbb{R}^{n \times n}$ be positive definite. Then there exists a lower triangular matrix $B \in \mathbb{R}^{n \times n}$ such that $A = BB^T$.

Cholesky Decomposition

A matrix A is *positive definite* if it is symmetric and

$$x^T Ax > 0, \qquad \forall x \neq 0 \tag{21}$$

For positive definite matrices we have a more efficient LU decomposition with $L = U^T$.

Example

We have $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, then $x^T Ax = x_1^2 + (x_1 + x_2)^2 > 0$ and so A is positive definite.

Theorem (Cholesky decomposition)

Let $A \in \mathbb{R}^{n \times n}$ be positive definite. Then there exists a lower triangular matrix $B \in \mathbb{R}^{n \times n}$ such that $A = BB^T$.

Proof.

By induction... □

Cholesky Decomposition – Proof

Base case $n = 1$ is trivial.

Cholesky Decomposition – Proof

Base case $n = 1$ is trivial. Consider general positive definite A and partition as

$$A = \begin{pmatrix} a_{11} & W^T \\ W & C \end{pmatrix}$$

where $W \in \mathbb{R}^{n \times 1}$ and $C \in \mathbb{R}^{n-1 \times n-1}$.

Cholesky Decomposition – Proof

Base case $n = 1$ is trivial. Consider general positive definite A and partition as

$$A = \begin{pmatrix} a_{11} & W^T \\ W & C \end{pmatrix}$$

where $W \in \mathbb{R}^{n \times 1}$ and $C \in \mathbb{R}^{n-1 \times n-1}$. By direct calculation we have

$$A = \begin{pmatrix} a_{11} & W^T \\ W & C \end{pmatrix} = \begin{pmatrix} \sqrt{a_{11}} & 0 \\ W/\sqrt{a_{11}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & C - WW^T/a_{11} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & W^T/\sqrt{a_{11}} \\ 0 & I \end{pmatrix}$$

Cholesky Decomposition – Proof

Base case $n = 1$ is trivial. Consider general positive definite A and partition as

$$A = \begin{pmatrix} a_{11} & W^T \\ W & C \end{pmatrix}$$

where $W \in \mathbb{R}^{n \times 1}$ and $C \in \mathbb{R}^{n-1 \times n-1}$. By direct calculation we have

$$A = \begin{pmatrix} a_{11} & W^T \\ W & C \end{pmatrix} = \begin{pmatrix} \sqrt{a_{11}} & 0 \\ W/\sqrt{a_{11}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & C - WW^T/a_{11} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & W^T/\sqrt{a_{11}} \\ 0 & I \end{pmatrix}$$

The middle matrix is block diagonal and positive definite therefore $C - WW^T/a_{11} = LL^T$ (Cholesky decomposition) by induction assumption. So

$$\begin{aligned} A &= \begin{pmatrix} \sqrt{a_{11}} & 0 \\ W/\sqrt{a_{11}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & LL^T \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & W^T/\sqrt{a_{11}} \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{a_{11}} & 0 \\ W/\sqrt{a_{11}} & L \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & W^T/\sqrt{a_{11}} \\ 0 & L^T \end{pmatrix} \end{aligned}$$

Cholesky Decomposition – Algorithm

$$A = \begin{pmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{21} & \dots & b_{n1} \\ 0 & b_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{nn} \end{pmatrix}$$

Cholesky Decomposition – Algorithm

$$A = \begin{pmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{21} & \dots & b_{n1} \\ 0 & b_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{nn} \end{pmatrix}$$

and so

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} b_{ik} b_{jk} = \langle (b_{ik})_k, (b_{jk})_k \rangle = a_{ji}$$

Cholesky Decomposition – Algorithm

$$A = \begin{pmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{21} & \dots & b_{n1} \\ 0 & b_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{nn} \end{pmatrix}$$

and so

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} b_{ik} b_{jk} = \langle (b_{ik})_k, (b_{jk})_k \rangle = a_{ji}$$

For the first column

$$b_{11} = \sqrt{a_{11}} \qquad \text{and} \qquad b_{i1} = \frac{a_{i1}}{b_{11}} \quad \text{for } 2 \leq i \leq n$$

Cholesky Decomposition – Algorithm

$$A = \begin{pmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{21} & \dots & b_{n1} \\ 0 & b_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{nn} \end{pmatrix}$$

and so

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} b_{ik} b_{jk} = \langle (b_{ik})_k, (b_{jk})_k \rangle = a_{ji}$$

For the first column

$$b_{11} = \sqrt{a_{11}} \qquad \text{and} \qquad b_{i1} = \frac{a_{i1}}{b_{11}} \quad \text{for } 2 \leq i \leq n$$

For column $2 \leq j \leq n$

$$b_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2} \qquad \text{and} \qquad b_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} b_{ik} b_{jk}}{b_{jj}}$$

Cholesky Decomposition – Algorithm II

Remark (Properties)

1. Suppose we solve $Ax = b$ for a positive definite A.
   - Gaussian elimination – approx $\frac{2n^3}{3}$ operations
   - Cholesky decomposition – approx $\frac{n^3}{3}$ operations
2. If $A = BB^T$ then

$$\det(A) = \det(BB^T) = \det(B)\det(B^T) = (b_{11}b_{22}\dots b_{nn})^2$$

Cholesky Decomposition – Algorithm II

Remark (Properties)

1. Suppose we solve $Ax = b$ for a positive definite A.
   - Gaussian elimination – approx $\frac{2n^3}{3}$ operations
   - Cholesky decomposition – approx $\frac{n^3}{3}$ operations
2. If $A = BB^T$ then

$$\det(A) = \det(BB^T) = \det(B)\det(B^T) = (b_{11}b_{22}\dots b_{nn})^2$$

Example

We have

$$A = \begin{pmatrix} 4 & -2 & 0 \\ -2 & 2 & 3 \\ 0 & 3 & 10 \end{pmatrix}$$

$$x^T Ax = (2x_1 - x_2)^2 + (x_2 + 3x_3)^2 + x_3^2 > 0$$

Cholesky Decomposition – Algorithm II

Remark (Properties)

1. Suppose we solve $Ax = b$ for a positive definite A.
   - Gaussian elimination – approx $\frac{2n^3}{3}$ operations
   - Cholesky decomposition – approx $\frac{n^3}{3}$ operations
2. If $A = BB^T$ then

$$\det(A) = \det(BB^T) = \det(B)\det(B^T) = (b_{11}b_{22}\dots b_{nn})^2$$

Example

We have

$$A = \begin{pmatrix} 4 & -2 & 0 \\ -2 & 2 & 3 \\ 0 & 3 & 10 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

$$x^T Ax = (2x_1 - x_2)^2 + (x_2 + 3x_3)^2 + x_3^2 > 0$$

Some exercises (working over $\mathbb{R}$)

1. Let V be an orthogonal matrix. Show that $\mathrm{cond}_2(V) = 1$.
2. Let $\|\cdot\|$ be a subordinate norm. Show that
$$\|Ax\| \leq \|A\|\|x\| \qquad \text{and} \qquad \|AB\| \leq \|A\|\|B\|$$
3. Let $\alpha \neq 0$ and U, V be orthogonal. Prove
$$(a)\quad \mathrm{cond}(A) \geq 1 \qquad (b)\quad \mathrm{cond}(A) = \mathrm{cond}(A^{-1})$$
$$(c)\quad \mathrm{cond}(\alpha A) = \mathrm{cond}(A) \qquad (d)\quad \mathrm{cond}_2(A) = \mathrm{cond}_2(UAV)$$
4. Let $A = \begin{pmatrix} 1 & \epsilon \\ \epsilon & 1 \end{pmatrix}$ for $0 < \epsilon < 1$ and let $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.
   (a) Using Gauss' method solve $Ax = b$.
   (b) Using Gauss-Jacobi method compute A^{-1}.
   (c) Compute $\mathrm{cond}_1(A)$ and $\mathrm{cond}_\infty(A)$. When is A well-conditioned? What happens at $\epsilon = 1$?
   (d) For what values of ϵ is A positive definite?
   (e) Find the LU decomposition of A.
   (f) Compute the Cholesky decomposition of A.
5. Suppose $Ax = b$ and $A'x' = b$. Show that
$$\frac{\|x' - x\|}{\|x'\|} \leq \mathrm{cond}(A)\frac{\|A' - A\|}{\|A\|}$$