Why matrix norm is important?

You know that a computer cannot represent real numbers exactly unless they are dyadic numbers: $d_0, d_1, \ldots, d_t \times 2^e$

So, suppose you want to solve

$$A \mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{m \times m}$$

But in reality, you have to encode $A, \mathbf{x}, \mathbf{b}$ on the computer.

Let $A' = \text{fl}(A)$, $\mathbf{x}' = \text{fl}(\mathbf{x})$, $\mathbf{b}' = \text{fl}(\mathbf{b})$.

i.e., you end up solving

$$A' \mathbf{x}' = \mathbf{b}'$$

Suppose for the moment, let's assume $\mathbf{b}' = \mathbf{b}$ for simplicity.

Now, you want to know the relative error of the solution:

$$\frac{\| \mathbf{x}' - \mathbf{x} \|}{\| \mathbf{x}' \|} = \frac{\| \mathbf{x}' - A^{-1} \mathbf{b} \|}{\| \mathbf{x}' \|}$$

$$\text{rel. error in solution} = \frac{\| \mathbf{x}' - A^{-1} A' \mathbf{x}' \|}{\| \mathbf{x}' \|}$$

$$= \frac{\| A^{-1} (A - A') \mathbf{x}' \|}{\| \mathbf{x}' \|}$$

$$\leq \frac{\| A^{-1} \| \cdot \| A - A' \| \cdot \| \mathbf{x}' \|}{\| \mathbf{x}' \|}$$
\[ = \frac{\| A \| \| A^{-1} \|}{\| A-A' \|} \]

**amplification factor**  
**relative error in matrix**

Now define the condition number of \( A \) by
\[ \kappa(A) = \text{cond}(A) := \| A \| \cdot \| A^{-1} \| \]

If \( \kappa(A) \) is large, then \( A \) is **pretty bad**, i.e., \( \exists \) large error in solution \( x = A^{-1}b \).

- Roughly speaking, to compute \( A^{-1} \) or the solution of \( Ax = b \), we lose \( \approx \log_{10} \kappa(A) \) digits.

- In particular, if \( A \) is singular, \( \kappa(A) = +\infty \).
A Brief Intro to

Least Squares Problem

Since the error analysis of Gaussian elimination & LU decomposition are subtle and difficult, we'll first talk about the least squares problem, then talk about the projections, QR decomposition, etc.

The Least Squares Problem was conceived by Gauss and Legendre around 1800 in the fields of astronomy & geodesy, in particular, model fitting to measured data.

Want to solve

\( \mathbf{Ax} = \mathbf{b} \)

where \( \mathbf{A} \in \mathbb{R}^{m \times n}, m > n \)

\( \mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m \)

In general, \( \mathbf{x} \) has no exact solution unless \( \mathbf{b} \in \text{range}(\mathbf{A}) \).

This usually does not happen!

\( \Rightarrow \) Check the size of the residual

\( \mathbf{r} := \mathbf{b} - \mathbf{Ax} \in \mathbb{R}^m \)

and want \( \mathbf{r} \) as small as possible.

Given \( \mathbf{A} \in \mathbb{R}^{m \times n}, m \geq n, \mathbf{b} \in \mathbb{R}^m \)

Find \( \mathbf{x} \in \mathbb{R}^n \) s.t. \( \| \mathbf{b} - \mathbf{Ax} \|_2 \rightarrow \min \)
This is called a general linear least squares problem.

Why 2-norm is used? ⇒ its geometric interpretation!

\[ \begin{align*}
\text{Thm.} \quad & \text{Let } A \in \mathbb{R}^{m \times n}, \ m \geq n, \ \mathbf{b} \in \mathbb{R}^m. \\
& \text{Then, } \mathbf{x} \in \mathbb{R}^n \text{ is the minimizer of} \\
& \| \mathbf{r} \|_2 = \| \mathbf{b} - A\mathbf{x} \|_2
\end{align*} \]

\[\iff \mathbf{r} \perp \text{range } (A) \]
\[\iff A^T \mathbf{r} = 0 \]
\[\iff A^T A \mathbf{x} = A^T \mathbf{b} \quad \text{(the normal equation)}\]
\[\iff A (A^T A)^{-1} A^T \mathbf{b} = A \mathbf{x} \]

Furthermore,
\[A^T A \text{ is nonsingular } \iff A \text{ is full rank} \]
Consequently,
\[\text{rank}(A) = n \quad \text{if } m \geq n \]
The solution \( \mathbf{x} \) is unique \( \iff A \text{ : full rank} \)
(Proof) These statements are essentially obvious from the figure.

Also \( \mathbb{R} \perp \text{range} (A) \)

\[
\iff \quad \mathbb{R} \perp A_j \quad 1 \leq j \leq n
\]

\[
\iff \quad \mathbb{R}^\top \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} = 0
\]

\[
\iff \quad \mathbb{R}^\top A = 0 \quad \iff \quad A^\top \mathbb{R} = 0
\]

Then

\[
A^\top \mathbb{R} = A^\top (b - A \mathbb{X})
\]

\[
= A^\top b - A^\top A \mathbb{X}
\]

\[
= 0
\]

\[
\iff \quad A^\top A \mathbb{X} = A^\top b
\]

Now we can also show the uniqueness of the orthogonal projection of \( b \) onto \( \text{range} (A) \) as follows:

Let \( y = A \left( A^\top A \right)^{-1} A^\top b \in \text{range} (A) \)

which obviously minimize \( \| b - y \|_2 \)

Suppose \( \exists \ Z \neq y \text{ s.t. } \| b - y \|_2 = \| b - Z \|_2 \), \( Z \in \text{range} (A) \)

Then, \( y - Z \in \text{range} (A) \).

So, \( y - Z \perp \mathbb{b} - \mathbb{y} \)

\[
\iff \quad \| b - Z \|_2^2 = \| b - y \|_2^2 + \| y - Z \|_2^2
\]

But these two are equal by the assumption.

\[
\therefore \quad \| y - Z \|_2 = 0 \iff y = Z
\]
• \((A^T A)^{-1} A^T\) is often called a pseudo-inverse of \(A\), and denoted by \(A^+\). This has infinitely many solutions in the case of \(m < n\). This case is called underdetermined. Need extra constraints to solve such LS problem, e.g., \(\min \|x\|_2\) subject to \(Ax = b\). That is, find \(x \in \mathbb{R}^n\), s.t. \(\min \|x\|_2\) subject to \(Ax = b\).

This is done by the Lagrange multipliers: Let \(J(x) := x^T x + \lambda^T (lb - Ax)\) with \(\lambda \in \mathbb{R}^m\). Then want \(\min J(x)\).

\[
\frac{\partial J}{\partial x} = 2x - A^T \lambda = 0
\]

\(\Leftrightarrow\) \(x = \frac{1}{2} A^T \lambda\), this \(x\) minimizes \(J(x)\).

Now, \(lb = Ax = \frac{1}{2} A A^T \lambda\)

\(\Leftrightarrow\) \(\lambda = 2 (A A^T)^{-1} lb\)

\(\Leftrightarrow\) \(x = \frac{1}{2} A^T \lambda = A^T (A A^T)^{-1} lb\)

Compare this with \((A^T A)^{-1} A^T lb\) in the case of \(m < n\).
Example  The LS polynomial fit

Given \( m \) distinct points \( x_1, \ldots, x_m \in \mathbb{R} \) and data \( y_1, \ldots, y_m \in \mathbb{R} \).

Want to fit a polynomial of deg. \( n-1 \)

\[
p(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}
\]

for some \( n < m \).

Such a polynomial is a LS fit to the data if it minimizes the residual

\[
(*) \quad \sum_{i=1}^{m} |p(x_i) - y_i|^2.
\]

So,

\[
\begin{bmatrix}
1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_m & x_m^2 & \cdots & x_m^{n-1}
\end{bmatrix}
\begin{bmatrix}
c_0 \\
\vdots \\
c_{n-1}
\end{bmatrix}
\approx
\begin{bmatrix}
y_1 \\
\vdots \\
y_m
\end{bmatrix}
\]

Vandermonde! \( (*) = ||Ax||_2^2 = ||b - Ax||_2^2 \)

MATLAB Demo here!