Q1: Why is \( \| x \|_0 = \min \{ |x_i| \} \) not a norm? (Choose one that applies - list the 3 properties)

\( \| (3,0) \|_0 = 0 \), so \( \| x \|_0 \) is not \( \geq 0 \). Fails triangle: \( 3 = \| (3,0) + (0,3) \|_0 \leq \| (3,0) \|_0 + \| (0,3) \|_0 = 0 + 0 = 0 \).

These are important, but more important is matrix norms.

Matrix norms are like vector norms, but you can multiply matrices. For a matrix norm, you want a norm that "plays nice" with matrix multiplication.

Def: A matrix norm is a norm (follows the 3 rules) and is "submultiplicative."

\[ \| AB \| \leq \| A \| \| B \| \] (like triangle inequality for multiplication.)

ex: Frobenius norm

\[ \sum_{i,j} | a_{ij} |^2 \]

Satisfies 3 properties, since same as vector 2-norm.

Submultiplicativity uses Cauchy-Schwarz, but we'll skip.

Why skip? Though (vector) 2-norm is important, the Frobenius norm turns out to not be as important in this course.

ex: Induced norm or operator norm.

Can think of \( A \) as being a transformation of vectors, you feed it a vector \( x \), get out the vector \( b = Ax \). How can we measure what \( A \) does over all? Find the vector that \( A \) makes biggest relative to its original size.

\[ \| A \|_M = \max_{x \neq 0} \frac{\| Ax \|_1}{\| x \|_1} \]

\( \| \cdot \|_M \) is vector norm

\( \| \cdot \|_M \) is (induced) matrix norm.

\( \frac{\text{relative size}}{\text{original size or magnitude}} \)

Usually, just write same subscript for both: \( \| A \|_2 = \max_{x \neq 0} \frac{\| Ax \|_2}{\| x \|_2} \), and you know difference based on whether vector/matrix is inside.

\( O(q^2) \) for \( \| A \|_2 \).

Very useful, very important, but unfortunately, often hard to calculate exactly.

Fortuitously, easy to estimate.

Thm: For an induced norm, \( \| Ax \| \leq \| A \| \| x \| \), for any vector \( x \). (And there is an \( x \) that makes this an equality.)

Q2: Why is \( \| x \|_2 \leq \| A \| \leq \| x \| \) for any vector \( x \)?

\[ \| A \|_2 = \max_{x \neq 0} \frac{\| Ax \|_2}{\| x \|_2} \]

so we can always estimate by just plugging in a random vector.
Thm: induced norm is a matrix norm. norm properties are easy using def, and the same properties for vector norms.

\[ \|xA\| \text{ if } \|x\| = \max \|Ax\| \]

\[ \|A\| \text{ if } \|A\| = \max \|Ax\| \]

Def: The matrix \( p \)-norm is the induced \( p \)-norm:

\[ \|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \]

The matrix 2-norm is also known as the spectral norm (since it has to do with eigenvalues.)

Q3: \( A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \), What is \( \|A\|_F \)?

\[ \|A\|_F = \sqrt{1^2 + 1^2} = \sqrt{2} \]

Q4: \( A = \begin{bmatrix} 1 & 0 \\ 0 & 7 \end{bmatrix} \), What is \( \|A\|_2 \)?

\[ \|A\|_2 \neq \sqrt{2} \]

Want to maximize magnitude. But \( A \) leaves things the same, so \( \max \text{ mag} = 1 \).

So \( \|A\|_2 = 1 \).

In fact, same logic shows \( \|I\|_1 = 1 \) for ANY induced norm!

Q5: Use some vectors to estimate \( \|A\|_1 \).

\[ \|A\|_1 = \max_{\|x\|_1 = 1} \|Ax\|_1 \]

\[ \|A\|_1 = \max_{\|x\|_1 = 1} \sum |a_{ij}| \]

as we class for vectors.

\[ \|A\|_1 = \max_{\|x\|_1 = 1} \sum |a_{ij}| \]

\[ x = (1,0) \Rightarrow \|Ax\|_1 = \frac{\|A(1,0)\|_1}{\|1,0\|_1} = \frac{1}{1} = 1 \]

\[ x = (0,1) \Rightarrow \|Ax\|_1 = \frac{\|A(0,1)\|_1}{\|0,1\|_1} = \frac{6}{1} = 6 \]

In fact, that is the norm: \( \|A\|_1 = 6 \)

Thm: \( \|A\|_1 = \text{ largest } 1\text{-norm of column} = \max_j \sum_i a_{ij} \)

\( \|A\|_{\infty} = \text{ largest } 1\text{-norm of row} = \max_i \sum_j a_{ij} \)

Proof: is actually nontrivial. Basic strategy is

1. Use basic inequalities about absolute values/maximums/sums to find an upper bound on \( \|A\|_1 \leq C \)

2. Find a specific vector that gives that same estimate, so \( \|A\|_1 \geq C \)

3. Thus, since \( \|A\|_1 \geq C \) and \( \leq C \), \( \|A\|_1 = C \).

(common general strategy.)
The point in discussing norms was to be able to measure error. A simple, but important, example of this is measuring sensitivity of a system. We can solve $Ax = b$.

But often, don't know $b$ exactly, either $b/c$ of numerical rounding error, or just limitations of real life measurements.

So, really solving $A\hat{x} = b + \delta b$, where $\delta b$ (one thing, not multiplication, "delta") is a perturbation (small change) to $b$.

$\hat{x} = x + \delta x$, i.e., $x$ is perturbed as well.

We want "low sensitivity," i.e., that if $\frac{||\delta b||}{||b||}$ is small (i.e., small perturbation) then $\frac{||\delta x||}{||x||}$ is small as well. (Small perturbation is sol'n.)

If true, then we know our computed answer should be accurate (need not, actually).

How to estimate? Use induced norms:

$Ax = b$ and $A(x + \delta x) = b + \delta b$, so $A\delta x = \delta b$, or $\delta x = A^{-1}\delta b$.

$||b|| \leq ||A|| ||x||$

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

Thus: If $K(A)$ is "small" and $\frac{||\delta b||}{||b||}$ is small, then the precise sol'n $\frac{||\delta x||}{||x||}$ is small.

Q1: Calculate $K_A([1 2 3; 4 5 6])$. ($A^{-1} = \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$) (use $\infty$ norm)

$||A||_{\infty} = \text{largest 1-norm of row} = 7$

$||A^{-1}||_{\infty} = 1 + 2 = 3$. So $K_A(A) = 21$.

In fact $K_A(A) \geq 1$, for induced norms.

Q2: If $K(A)$ is large, then our $||\delta x||$ will be unacceptably high.

Not necessarily. If $\frac{||\delta b||}{||b||}$ is tiny, then we're fine.

Or, if we just get lucky. This is an upper bound on error.

But no guarantee!

Principle: If $A$ is well-conditioned ($K(A)$ small), then our error is probably small.

If $A$ is ill-conditioned ($K(A)$ large), then we do not expect our answer to be helpful.

Note: We are not yet talking about rounding errors. This estimate we proved is a bound on the error of an exact calculation, assuming an approximate $b$. Thus, we have not yet shown our computers are accurate.
Interpretation of $K(A)$.
Recall: Solving $Ax = b$ is the same as finding the intersection of many lines/planes/hyperplanes.

\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix} =
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix} \iff x + 2y = b_1 \land 3x + 4y = b_2
\]

If lines $\approx \perp$, then not large error, as here. But if $\approx \parallel$...

And actually, most ill-conditioned systems are much worse.

So: Ill-conditioned $\approx$ rows are roughly multiples of each other.

Another way:

\[ K(A) = \frac{\max \text{mag of } A}{\min \text{mag of } A} = \frac{\max \|Ax\|}{\min \|Ax\|} \]

(not 0 cuz not singular)

\[ = \min \|b\| \frac{\|A\|}{\|b\|} = \left( \max \|A^{-1}b\| \right)^{-1} = \frac{1}{\|A^{-1}\|} \]

So $K(A) = \|A\| \|A^{-1}\| = \frac{\max \text{mag of } A}{\min \text{mag of } A}$

So $K(A)$ is large if $A$ transforms some vectors much larger than others.

One problem with condition number: Can be caused by just poor scaling of a row:

ex: 

\[
\begin{bmatrix}
1 & 0 \\
0 & \delta
\end{bmatrix}
\]

has $K_p(A) = \frac{1}{\delta}$, large if $\delta$ tiny.

But scale row by $\frac{1}{\delta}$, including any b's you might have, and it's fine.

Hard to calculate exactly. ex: Takes $2n^3$ flops to calculate $A^{-1}$. Then have to find $\|A\|$ and $\|A^{-1}\|$!

But there are shortcuts to estimate:

Then $K_p(A) \geq \frac{\|a_i\|}{\|a_j\|}$ where $a_i$ are the columns of $A$.

Or, if you have LU decomposition, pick $w$, calculate $A^{-1}w$ (by solving $Ax = w$), then $K_v(A) \leq \frac{\|A\|}{\|A^{-1}\|}$ MatLab essentially uses this.
So, far, we've only perturbed b. But often we also don't know A precisely either. How accurate is our solution if A is perturbed?

First, if solving \((A + SA) \hat{x} = b\), first we need to be careful it has a sol'n at all! \(A + SA\) may no longer be invertible!

Recall: \(K(A)\) large means lines (planes) almost parallel. If almost \(\perp\), then a small \(SA\) could make them exactly \(\perp\), and so \(A + SA\) would be singular. Precisely:

**Thm:** If \(A\) is nonsingular and if \(\frac{\|SA\|}{\|A\|} \leq \frac{1}{K(A)}\), then \(A + SA\) is nonsingular.

i.e., if \(K(A)\) is large, need \(SA\) to be tiny.

**pf:** Inequality is \(\frac{\|SA\|}{\|A\|} \leq \frac{1}{\|A^{-1}\| \cdot \|SA\|} \Rightarrow \|SA\| \cdot \|A^{-1}\| \cdot \|A\| \leq 1\).

Suppose \(A + SA\) was singular. Then there's a \(y \neq 0\) s.t. \((A + SA)y = 0\).

Rearrange as \(-y = A^{-1} (SAy)\)

\[\|y\| \leq \|A\| \cdot \|SAy\| \leq \|A^{-1}\| \cdot \|SA\| \cdot \|y\|\]

\[\leq \|A^{-1}\| \cdot \|SA\| \cdot \|A\| \cdot \|A\| \leq 1\].

But that contradicts our assumption.

So \(A + SA\) can't be singular.

Okay: Look at \((A + SA)(x + 6x) = b\). Want to estimate \(6x\).

\[\Rightarrow A\hat{x} + A\hat{SA}(x + 6x) = b\]
\[\Rightarrow 6x = A^{-1} SA (x + 6x)\] as before

\[\|6x\| \leq \|A^{-1}\| \cdot \|SA\| \cdot \|x + 6x\|\]

rearrange, \(\|6x\| \leq K(A) \cdot \|SA\| \cdot \|x + 6x\|\)

\[\|x + 6x\| \leq K(A) \cdot \|SA\| \cdot \|x\|\]

\[x = x + 6x\] usually, so okay, But would prefer just \(x\) on bottom.

**Thm:** If \(A\) is nonsingular and \((A + SA)(x + 6x) = b\), Then

\[\|6x\| \leq K(A) \cdot \|SA\| \cdot \|x + 6x\|\]

if you also know \(\frac{\|SA\|}{\|A\|} \leq \frac{1}{K(A)}\), then

\[\|6x\| \leq K(A) \cdot \|SA\| \cdot \|x\| \leq K(A) \cdot \|SA\| \cdot \|x\| \cdot \left(1 - \frac{K(A) \cdot \|SA\|}{\|A\|}\right)\]

Similar statements in book if both \(A\) and \(b\) are perturbed.

**Q2:** \(A = \begin{bmatrix} 1 & 1.01 \\ 1.01 & 1 \end{bmatrix}\), \(A' = \begin{bmatrix} 101 & -100 \\ -100 & 100 \end{bmatrix}\). If \(\|SA\| \leq 0.001\), how large could our error be from solving \((A + SA)(x + 6x) = b\)? (Use 1-norm)

\[K_1(A) = \|A\|, |A^{-1}|_1 = 2.01 \cdot 2.01 = 4.04\]

Using 1st, get \(\frac{\|6x\|}{\|x\|} \leq \frac{4.04 \cdot 0.001}{2} = 0.002\). Not great, means \(\approx 20\%\) error!

2nd, get \(\frac{\|6x\|}{\|x\|} \leq \frac{4.04 \cdot 0.001}{1 - 4.04 \cdot 0.001} = \frac{0.002}{0.997} = 0.2\). So \(\approx 25\%\) error.

Though these are worst cases. Probably much more accurate, but maybe not!
2.4: So far, we've assumed an exact solution to a problem with approximate \( A, b \). (and can still have bad solns.)

But on a computer, we have an approximate solution (due to rounding errors, etc.) to a problem with approximate (or exact, depending) \( A, b \).

How do we know our soln is accurate?? Need estimates!

A priori estimate: "a priori" means "before." This kind of estimate is made before you do the calculation. It's a worst case scenario, but an important theoretical tool, since then you know how bad it could possibly get. But hard, in general. We'll spend some time developing.

A posteriori estimate: "after." Rather than try to estimate before, this takes an already computed solution, and uses it to estimate how wrong it could be. Easy to do, important for verifying accuracy, but not a theoretical tool, since have to calculate first.

How to do: Solving \( Ax = b \), with true soln \( x \). But you end up with \( \hat{x} \) as your computed soln. It solves \( A \hat{x} = b + \delta b \) for some \( \delta b \).

If \( \hat{x} = x + \delta x \) and \( -\delta b = \hat{r} = b - A\hat{x} \) is the residual, then can use earlier estimate \( \frac{\|\delta x\|}{\|x\|} \leq K(A) \frac{\|\delta b\|}{\|b\|} \).

Thm: If \( A \) is nonsingular, and \( \hat{x} \) is any approximate soln to \( Ax = b \) (even a guess!)
then \( \frac{\|x - \hat{x}\|}{\|x\|} \leq K(A) \frac{\|\hat{r}\|}{\|b\|} \)

relative error \( \leq \) condition number \( \cdot \) relative size of residual to \( b \).

Q3: \( A = \begin{bmatrix} 1 & 1.01 \\ 1 & 1 \end{bmatrix} \), \( K(A) \approx 400 \). If I solve \( Ax = b \) for \( b = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \) and my residual is \( \hat{r} = \begin{bmatrix} 3 \cdot 10^{-5} \\ 1 \cdot 10^{-5} \end{bmatrix} \), how inaccurate could my soln be? (use l-norm)

\( \|\hat{r}\|_1 = 3 \cdot 10^{-5}, \|b\|_1 = 3, \) so \( \frac{\|\delta x\|}{\|x\|} \leq 400 \cdot 3 \cdot 10^{-5} \approx 0.004, \) so with 0.4% off true.

Usually, \( \hat{r} \) is very small. The real problem is \( K(A) \).

If \( K(A) \) is very large, nothing you do will guarantee an accurate soln. (or, at least much harder than you want...)

Calculate \( \hat{r} = b - A\hat{x} \) is \( \approx 2n^2 \) flops, so not super expensive.

But can use to check accuracy, if worried.