2072U Computational Science I
Winter 2022

Week	Topic
1	Introduction
1–2	Solving nonlinear equations in one variable
3–4	Solving systems of (non)linear equations
5–6	Computational complexity
6–8	Interpolation and least squares
8–10	Integration & differentiation
10–12	Additional Topics

1. The three questions...
2. Vector norms
3. Quantifying errors using norms
4. Conditioning of linear equations

Central questions:

- What is a vector norm?
- What are some examples of norms?
- How are vector norms computed in SciPy?
- How are norms useful in quantifying errors in solving linear systems?
- What are the singular values of a matrix?
- What is the condition number of a matrix? Computing it with `NumPy`.
- What does conditioning mean intuitively?

The *three questions* for approximation solutions to linear systems:

1. When does my computation work?
2. How accurate is the result?
3. How fast does my computation work?

Answer:

The *three questions* for approximation solutions to linear systems:

1. When does my computation work?
2. How accurate is the result?
3. How fast does my computation work?

Answer:

- The $A = LU$ decomposition works if and only if all leading principal submatrices of A (i.e. $A(1:k, 1:k)$ for $k \leq n$) are nonsingular. Not recommended for linear solving!

The *three questions* for approximation solutions to linear systems:

1. When does my computation work?
2. How accurate is the result?
3. How fast does my computation work?

Answer:

- The $A = LU$ decomposition works if and only if all leading principal submatrices of A (i.e. $A(1:k, 1:k)$ for $k \leq n$) are nonsingular. Not recommended for linear solving!
- The $PA = LU$ decomposition works if A is nonsingular. This is the default method for linear solving:

$$\begin{aligned}\text{step 1: solve } Ly = Pb & \text{ using forward substitution}\\ \text{step 2: solve } Ux = y & \text{ using backward substitution}\end{aligned}$$

Next, we turn to the second question...

Motivation for vector norms

If solutions to linear systems are vectors, how can you tell how big the error is?

- Real numbers are ordered:

$$\text{given } a, b \in \mathbb{R}, \text{ either } a < b, a > b, \text{ or } a = b$$

- Vectors in $\mathbb{R}^n$ are not ordered,e.g., expressions like

$$\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} > \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix} \text{ or } \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} < \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$$

  do not make sense.
- Norms provide a way to order vectors, measure distance.

Definition (Vector norm)

Given a vector space V, a norm is a function $\|\cdot\|: V \to [0, \infty)$ satisfying three postulates:

1. $\|\mathbf{v}\| > 0$ if $\mathbf{v} \neq \mathbf{0}$ for every $\mathbf{v} \in V$
2. $\|\lambda\mathbf{v}\| = |\lambda|\|\mathbf{v}\|$ for every $\lambda \in \mathbb{R}$, $\mathbf{v} \in V$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ for every $\mathbf{u}, \mathbf{v} \in V$ (triangle inequality)

Definition (Vector norm)

Given a vector space V, a norm is a function $\|\cdot\| : V \to [0,\infty)$ satisfying three postulates:

1. $\|\mathbf{v}\| > 0$ if $\mathbf{v} \neq \mathbf{0}$ for every $\mathbf{v} \in V$
2. $\|\lambda\mathbf{v}\| = |\lambda|\|\mathbf{v}\|$ for every $\lambda \in \mathbb{R}$, $\mathbf{v} \in V$
3. $\|\mathbf{u}+\mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ for every $\mathbf{u}, \mathbf{v} \in V$ (triangle inequality)

- $\|\mathbf{x}\|$ provides notion of length or size of vector $\mathbf{x}$.

Definition (Vector norm)

Given a vector space V, a norm is a function $\|\cdot\| : V \to [0, \infty)$ satisfying three postulates:

1. $\|\mathbf{v}\| > 0$ if $\mathbf{v} \neq \mathbf{0}$ for every $\mathbf{v} \in V$
2. $\|\lambda \mathbf{v}\| = |\lambda| \|\mathbf{v}\|$ for every $\lambda \in \mathbb{R}$, $\mathbf{v} \in V$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ for every $\mathbf{u}, \mathbf{v} \in V$ (triangle inequality)

- $\|\mathbf{x}\|$ provides notion of length or size of vector $\mathbf{x}$.
- $\|\mathbf{x} - \mathbf{y}\|$ provides notion of distance between vectors $\mathbf{x}$, $\mathbf{y}$.

The ℓ_2-norm

$$\|\mathbf{x}\|_2 := \left[\sum_{k=1}^{n} |x_k|^2\right]^{\frac{1}{2}} \quad \forall \mathbf{x} \in \mathbb{R}^n$$

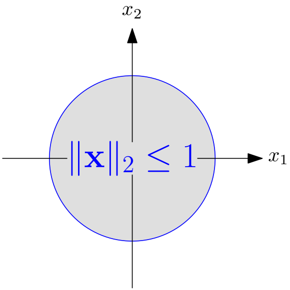
- Also called Euclidean norm or 2-norm.
- Compute by `scipy.linalg.norm(x,2)`.

$$\left\| \left[3, -4, 0, \frac{3}{2}\right]^T \right\|_2 = \sqrt{(3)^2 + (-4)^2 + (0)^2 + \left(\frac{3}{2}\right)^2} = \boxed{\frac{1}{2}\sqrt{109}}$$

$$\left\| [2, 1, -3, 4]^T \right\|_2 = \sqrt{(2)^2 + (1)^2 + (-3)^2 + (4)^2} = \boxed{\sqrt{30}}$$

ℓ_2-norm in $\mathbb{R}^2$:

$$\|\mathbf{x}\|_2 = \left(\sum_{k=1}^{n} |x_k|^2\right)^{\frac{1}{2}}$$



$$\begin{aligned}
\text{Unit ball in } \ell_2\text{-norm} &= \text{set of all vectors } \mathbf{x} \in \mathbb{R}^2 \text{ with } \|\mathbf{x}\|_2 \leq 1 \\
&= \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\|_2 \leq 1\} \\
&= \{(x_1, x_2) \in \mathbb{R}^2 \mid \sqrt{|x_1|^2 + |x_2|^2} \leq 1\} \\
&= \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1\} \\
&= \text{circle of radius 1 centred at origin}
\end{aligned}$$

The ℓ_1-norm

$$\|\mathbf{x}\|_1 := \sum_{k=1}^{n} |x_k| \quad \forall \mathbf{x} \in \mathbb{R}^n$$

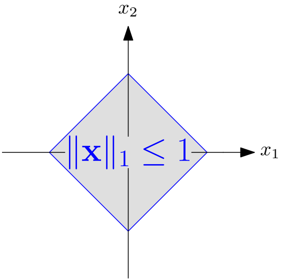
- Also called Manhattan norm or 1-norm.
- Compute by `scipy.linalg.norm(x,1)`.

$$\left\| \left[3, -4, 0, \frac{3}{2}\right]^T \right\|_1 = |3| + |-4| + |0| + \left|\frac{3}{2}\right| = \boxed{\frac{17}{2}}$$

$$\left\| [2, 1, -3, 4]^T \right\|_1 = |2| + |1| + |-3| + |4| = \boxed{10}$$

ℓ_1-norm in $\mathbb{R}^2$:

$$\|\mathbf{x}\|_1 = \sum_{k=1}^{n} |x_k|$$



$$\begin{aligned}
\text{Unit ball in } \ell_1\text{-norm} &= \text{set of all vectors } \mathbf{x} \in \mathbb{R}^2 \text{ with } \|\mathbf{x}\|_1 \leq 1 \\
&= \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\|_1 \leq 1\} \\
&= \{(x_1, x_2) \in \mathbb{R}^2 \mid |x_1| + |x_2| \leq 1\} \\
&= \{(x_1, x_2) \in \mathbb{R}^2 \mid (\pm x_1) + (\pm x_2) \leq 1\} \\
&= \text{square with vertices } (\pm 1, 0), (0, \pm 1)
\end{aligned}$$

The ℓ_∞-norm

$$\|\mathbf{x}\|_\infty := \max\left(|x_1|, |x_2|, \ldots, |x_n|\right) \quad \forall \mathbf{x} \in \mathbb{R}^n$$

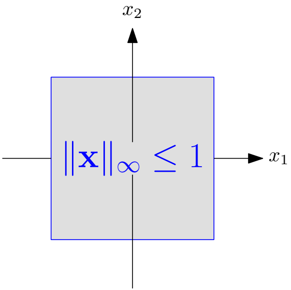
- Also called max/infinity/Chebyschev norm.
- Compute by `scipy.linalg.norm(x,scipy.inf)`.

$$\left\|\left[3, -4, 0, \frac{3}{2}\right]^T\right\|_\infty = \max\left(|3|, |-4|, |0|, \left|\frac{3}{2}\right|\right) = \boxed{4}$$

$$\left\|[2, 1, -3, 4]^T\right\|_\infty = \max\left(|2|, |1|, |-3|, |4|\right) = \boxed{4}$$

ℓ_∞-norm in $\mathbb{R}^2$:

$$\|\mathbf{x}\|_\infty = \max_{1\le k\le n} |x_k|$$



$$\begin{aligned}
\text{Unit ball in } \ell_\infty\text{-norm} &= \text{set of all vectors } \mathbf{x} \in \mathbb{R}^2 \text{ with } \|\mathbf{x}\|_\infty \le 1 \\
&= \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\|_\infty \le 1\} \\
&= \{(x_1, x_2) \in \mathbb{R}^2 \mid \max\{|x_1|, |x_2|\} \le 1\} \\
&= \{(x_1, x_2) \in \mathbb{R}^2 \mid |x_1| \le 1 \text{ and } |x_2| \le 1\} \\
&= \text{square with vertices } (\pm 1, \pm 1)
\end{aligned}$$

The ℓ_p-norm ($p \geq 1$)

$$\|\mathbf{x}\|_p := \left[\sum_{k=1}^{n} |x_k|^p\right]^{\frac{1}{p}} \quad \forall \mathbf{x} \in \mathbb{R}^n$$

- Generalises norms observed so far.
- Compute by `scipy.linalg.norm(x,p)`.

$$\left\| \left[3, -4, 0, \frac{3}{2}\right]^T \right\|_4 = \sqrt[4]{|3|^4 + |-4|^4 + |0|^4 + \left|\frac{3}{2}\right|^4} = \boxed{\frac{1}{2}\sqrt[4]{5473}}$$

$$\left\| [2, 1, -3, 4]^T \right\|_3 = \sqrt[3]{|2|^3 + |1|^3 + |-3|^3 + |4|^2} = \boxed{\sqrt[3]{100}}$$

Norms and relative errors

- Help quantify the second of the three question:
  1. When does my computation work?
  2. How accurate is the result?
  3. How fast does my computation work?
- $\|\mathbf{x} - \mathbf{x}_*\|$ small means $\mathbf{x}_* \in \mathbb{R}^n$ approximates $\mathbf{x} \in \mathbb{R}^n$ well.

Norms and relative errors

- Help quantify the second of the three question:
  1. When does my computation work?
  2. How accurate is the result?
  3. How fast does my computation work?
- $\|\mathbf{x} - \mathbf{x}_*\|$ small means $\mathbf{x}_* \in \mathbb{R}^n$ approximates $\mathbf{x} \in \mathbb{R}^n$ well.
- Define relative error of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$:

$$\begin{array}{c}\text{Relative} \\ \text{error of } \mathbf{x}_* \\ \text{in norm } \|\cdot\|\end{array} := \frac{\|\mathbf{x} - \mathbf{x}_*\|}{\|\mathbf{x}\|} \qquad (\text{assuming } \mathbf{x} \neq \mathbf{0})$$

Norms and relative errors

- Help quantify the second of the three question:
  1. When does my computation work?
  2. How accurate is the result?
  3. How fast does my computation work?
- $\|\mathbf{x} - \mathbf{x}_*\|$ small means $\mathbf{x}_* \in \mathbb{R}^n$ approximates $\mathbf{x} \in \mathbb{R}^n$ well.
- Define relative error of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$:

$$\begin{array}{c}\text{Relative}\\ \text{error of } \mathbf{x}_*\\ \text{in norm } \|\cdot\|\end{array} := \frac{\|\mathbf{x} - \mathbf{x}_*\|}{\|\mathbf{x}\|} \qquad (\text{assuming } \mathbf{x} \neq \mathbf{0})$$

- Computing (relative) error requires choosing a norm.

Norms and relative errors

- Help quantify the second of the three question:
  1. When does my computation work?
  2. How accurate is the result?
  3. How fast does my computation work?
- $\|\mathbf{x}-\mathbf{x}_*\|$ small means $\mathbf{x}_* \in \mathbb{R}^n$ approximates $\mathbf{x} \in \mathbb{R}^n$ well.
- Define relative error of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$:

$$\begin{array}{c}\text{Relative}\\ \text{error of } \mathbf{x}_*\\ \text{in norm } \|\cdot\|\end{array} := \frac{\|\mathbf{x}-\mathbf{x}_*\|}{\|\mathbf{x}\|} \qquad (\text{assuming } \mathbf{x} \neq \mathbf{0})$$

- Computing (relative) error requires choosing a norm.
- Norm-wise errors can hide component-wise errors in vectors.

Example

Compute relative errors in the ∞–norm, 1–norm, and 2–norm norms of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$ if

$$\mathbf{x} = \begin{pmatrix} 1.0000 \\ 0.0100 \\ 0.0001 \end{pmatrix} \text{ and } \mathbf{x}_* = \begin{pmatrix} 1.0002 \\ 0.0103 \\ 0.0002 \end{pmatrix}$$

Example

Compute relative errors in the ∞–norm, 1–norm, and 2–norm norms of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$ if

$$\mathbf{x} = \begin{pmatrix} 1.0000 \\ 0.0100 \\ 0.0001 \end{pmatrix} \text{ and } \mathbf{x}_* = \begin{pmatrix} 1.0002 \\ 0.0103 \\ 0.0002 \end{pmatrix}$$

$$\Rightarrow \mathbf{e} := \mathbf{x} - \mathbf{x}_* = \begin{pmatrix} -0.0002 \\ -0.0003 \\ -0.0001 \end{pmatrix} = -10^{-4} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$

Example

Compute relative errors in the ∞–norm, 1–norm, and 2–norm norms of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$ if

$$\mathbf{x} = \begin{pmatrix} 1.0000 \\ 0.0100 \\ 0.0001 \end{pmatrix} \text{ and } \mathbf{x}_* = \begin{pmatrix} 1.0002 \\ 0.0103 \\ 0.0002 \end{pmatrix}$$

$$\Rightarrow \mathbf{e} := \mathbf{x} - \mathbf{x}_* = \begin{pmatrix} -0.0002 \\ -0.0003 \\ -0.0001 \end{pmatrix} = -10^{-4} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$

Absolute/relative error measured in all three norms $\simeq 10^{-4}$

Example

Compute relative errors in the ∞–norm, 1–norm, and 2–norm norms of $\mathbf{x}_*$ as an approximation of $\mathbf{x}$ if

$$\mathbf{x} = \begin{pmatrix} 1.0000 \\ 0.0100 \\ 0.0001 \end{pmatrix} \text{ and } \mathbf{x}_* = \begin{pmatrix} 1.0002 \\ 0.0103 \\ 0.0002 \end{pmatrix}$$

$$\Rightarrow \mathbf{e} := \mathbf{x} - \mathbf{x}_* = \begin{pmatrix} -0.0002 \\ -0.0003 \\ -0.0001 \end{pmatrix} = -10^{-4} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$

Absolute/relative error measured in all three norms $\simeq 10^{-4}$
However, relative error in last component is 100% !

Linear equations, errors, and residuals

- Data $A \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$ prescribed: solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$

$$\mathbf{x} = \text{true solution of } A\mathbf{x} = \mathbf{b}$$
$$\mathbf{x}_* = \text{computed solution of } A\mathbf{x} = \mathbf{b}$$

Definition: error & residual

$$\mathbf{e} := \mathbf{x} - \mathbf{x}_* = \text{error vector} \qquad \|\mathbf{e}\| = \text{error}$$
$$\mathbf{r} := \mathbf{b} - A\mathbf{x}_* = \text{residual vector} \qquad \|\mathbf{r}\| = \text{residual}$$

Linear equations, errors, and residuals

- Data $A \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$ prescribed: solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$

$$\mathbf{x} = \text{true solution of } A\mathbf{x} = \mathbf{b}$$
$$\mathbf{x}_* = \text{computed solution of } A\mathbf{x} = \mathbf{b}$$

Definition: error & residual

$$\mathbf{e} := \mathbf{x} - \mathbf{x}_* = \text{error vector} \qquad \|\mathbf{e}\| = \text{error}$$
$$\mathbf{r} := \mathbf{b} - A\mathbf{x}_* = \text{residual vector} \qquad \|\mathbf{r}\| = \text{residual}$$

- If $\mathbf{x}_* = \mathbf{x}$, $\|\mathbf{e}\| = \|\mathbf{r}\| = 0$.

Linear equations, errors, and residuals

- Data $A \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$ prescribed: solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$

$$\mathbf{x} = \text{true solution of } A\mathbf{x} = \mathbf{b}$$
$$\mathbf{x}_* = \text{computed solution of } A\mathbf{x} = \mathbf{b}$$

Definition: error & residual

$$\mathbf{e} := \mathbf{x} - \mathbf{x}_* = \text{error vector} \qquad \|\mathbf{e}\| = \text{error}$$
$$\mathbf{r} := \mathbf{b} - A\mathbf{x}_* = \text{residual vector} \qquad \|\mathbf{r}\| = \text{residual}$$

- If $\mathbf{x}_* = \mathbf{x}$, $\|\mathbf{e}\| = \|\mathbf{r}\| = 0$.
- Generally, $\mathbf{x}$ unknown, so $\mathbf{e}$ not computable.

Linear equations, errors, and residuals

- Data $A \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$ prescribed: solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$

$$\begin{aligned} \mathbf{x} &= \text{true solution of } A\mathbf{x} = \mathbf{b} \\ \mathbf{x}_* &= \text{computed solution of } A\mathbf{x} = \mathbf{b} \end{aligned}$$

Definition: error & residual

$$\begin{aligned} \mathbf{e} &:= \mathbf{x} - \mathbf{x}_* = \text{error vector} & \|\mathbf{e}\| &= \text{error} \\ \mathbf{r} &:= \mathbf{b} - A\mathbf{x}_* = \text{residual vector} & \|\mathbf{r}\| &= \text{residual} \end{aligned}$$

- If $\mathbf{x}_* = \mathbf{x}$, $\|\mathbf{e}\| = \|\mathbf{r}\| = 0$.
- Generally, $\mathbf{x}$ unknown, so $\mathbf{e}$ not computable.
- We know A, $\mathbf{b}$, $\mathbf{x}_*$, so $\mathbf{r}$ computable.

"Good" linear system of equations:

- Consider linear system of equations

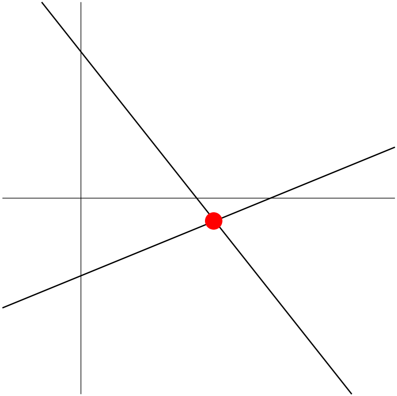
$$\begin{array}{rcrcl} x_1 & + & x_2 & = & 2 \\ x_1 & - & 3x_2 & = & 3 \end{array}$$

- In matrix form, $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -3 \end{pmatrix},$$

$$\mathbf{b} = \begin{pmatrix} 2 \\ 3 \end{pmatrix},$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



"Good" linear system of equations:

- Consider linear system of equations

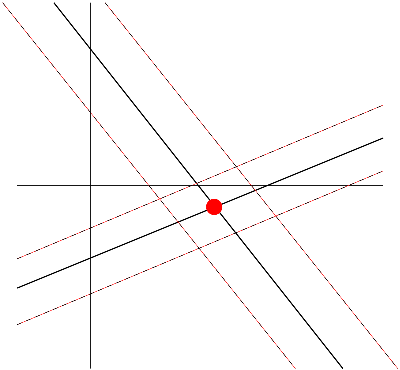
$$\begin{aligned} x_1 &+ x_2 &= 2 \\ x_1 &- 3x_2 &= 3 \end{aligned}$$

- In matrix form, $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -3 \end{pmatrix},$$

$$\mathbf{b}_* = \begin{pmatrix} 2 \pm 0.1 \\ 3 \pm 0.1 \end{pmatrix},$$

$$\mathbf{x}_* = \begin{pmatrix} x_{1*} \\ x_{2*} \end{pmatrix}$$



"Good" linear system of equations:

- Consider linear system of equations

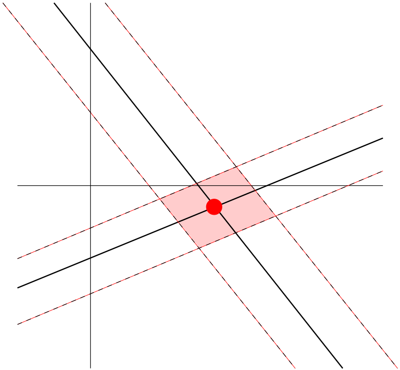
$$\begin{aligned} x_1 &+ x_2 &= 2 \\ x_1 &- 3x_2 &= 3 \end{aligned}$$

- In matrix form, $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -3 \end{pmatrix},$$

$$\mathbf{b}_* = \begin{pmatrix} 2 \pm 0.1 \\ 3 \pm 0.1 \end{pmatrix},$$

$$\mathbf{x}_* = \begin{pmatrix} x_{1*} \\ x_{2*} \end{pmatrix}$$



- Small change in b leads to small change in x_*.

"Bad" linear system of equations:

- Consider linear system of equations

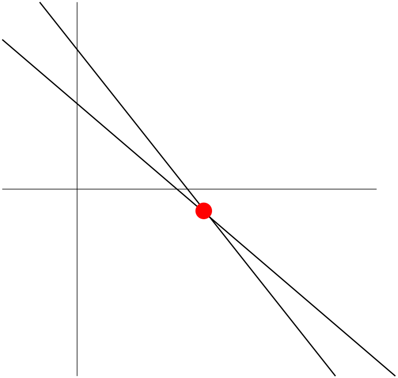
$$\begin{array}{ccccc} x_1 & + & x_2 & = & 2 \\ x_1 & + & 0.9x_2 & = & 1.9 \end{array}$$

- In matrix form, $B\mathbf{x} = \mathbf{b}$ with

$$B = \begin{pmatrix} 1 & 1 \\ 1 & 0.9 \end{pmatrix},$$

$$\mathbf{b} = \begin{pmatrix} 2 \\ 1.9 \end{pmatrix},$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



"Bad" linear system of equations:

- Consider linear system of equations

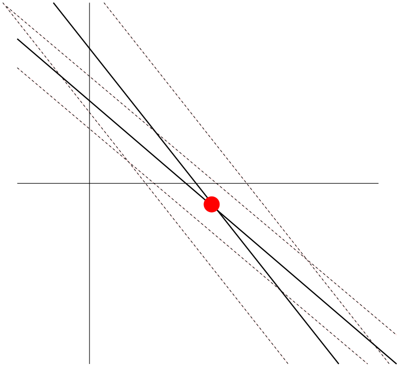
$$\begin{array}{ccccc} x_1 & + & x_2 & = & 2 \\ x_1 & + & 0.9x_2 & = & 1.9 \end{array}$$

- In matrix form, $B\mathbf{x} = \mathbf{b}$ with

$$B = \begin{pmatrix} 1 & 1 \\ 1 & 0.9 \end{pmatrix},$$

$$\mathbf{b}_* = \begin{pmatrix} 2 \pm 0.1 \\ 1.9 \pm 0.1 \end{pmatrix},$$

$$\mathbf{x}_* = \begin{pmatrix} x_{1*} \\ x_{2*} \end{pmatrix}$$



"Bad" linear system of equations:

- Consider linear system of equations

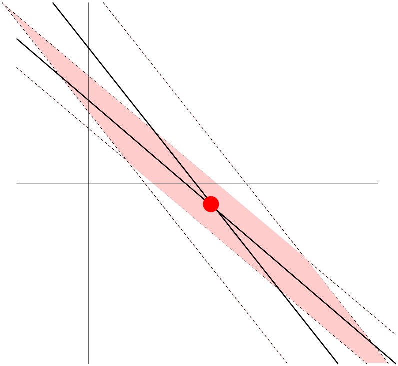
$$\begin{aligned} x_1 &+ x_2 &= 2 \\ x_1 &+ 0.9x_2 &= 1.9 \end{aligned}$$

- In matrix form, $B\mathbf{x} = \mathbf{b}$ with

$$B = \begin{pmatrix} 1 & 1 \\ 1 & 0.9 \end{pmatrix},$$

$$\mathbf{b}_* = \begin{pmatrix} 2 \pm 0.1 \\ 1.9 \pm 0.1 \end{pmatrix},$$

$$\mathbf{x}_* = \begin{pmatrix} x_{1*} \\ x_{2*} \end{pmatrix}$$



- Small change in b leads to big change in x_*.

Condition numbers

- $K(A)$: Condition number of matrix A with $1 \leq K(A) < \infty$
- Key property: $\boxed{\dfrac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq K(A)\dfrac{\|\mathbf{r}\|}{\|\mathbf{b}\|}}$, i.e.,

$$\boxed{\begin{matrix}\text{relative error}\\ \text{of } \mathbf{x}_*\end{matrix} \leq \left(\begin{matrix}\text{condition}\\ \text{number}\end{matrix}\right)\left(\begin{matrix}\text{relative residual}\\ \text{of } \mathbf{x}_*\end{matrix}\right)}$$

- Compute by `numpy.linalg.cond`

```
>>> import numpy
>>> import numpy.linalg
>>> A=numpy.matrix([[1.0,1.0],[1.0,-3.0]])
>>> numpy.linalg.cond(A,2)
2.618033988749895
>>> B=numpy.matrix([[1.0,1.0],[1.0,0.9]])
>>> numpy.linalg.cond(B,2)
38.07735174775756
```

Background:

- The condition number of a matrix is defined as the quotient of its largest to its smallest *singular values*.

$$K(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

Background:

- The condition number of a matrix is defined as the quotient of its largest to its smallest *singular values*.

$$K(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

- A singular value of A is the square root of an eigenvalue of $A^T A$, i.e.

$$A^T A\,\mathbf{w} = \sigma^2\,\mathbf{w}$$

Background:

- The condition number of a matrix is defined as the quotient of its largest to its smallest *singular values*.

$$K(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

- A singular value of A is the square root of an eigenvalue of $A^T A$, i.e.

$$A^T A\,\mathbf{w} = \sigma^2\,\mathbf{w}$$

- If A satisfies $A^T A = A\,A^T$ then its singular values equal the modulus of the eigenvalues ($\sigma = |\lambda|$).

Key property: $$\boxed{\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}}$$

- The condition number $K(A)$ is an indicator of whether a system of linear equations $A\mathbf{x} = \mathbf{b}$ is "good" or "bad"

Key property:

$$\boxed{\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}}$$

- The condition number $K(A)$ is an indicator of whether a system of linear equations $A\mathbf{x} = \mathbf{b}$ is “good” or “bad”
- If $K(A)$ is small, it’s “good”: we call it well-conditioned

Key property: $$\boxed{\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}}$$

- The condition number $K(A)$ is an indicator of whether a system of linear equations $A\mathbf{x} = \mathbf{b}$ is “good” or “bad”
- If $K(A)$ is small, it’s “good”: we call it well-conditioned
- If $K(A)$ is large, it’s “bad”: we call it ill-conditioned

Key property: $$\boxed{\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}}$$

- The condition number $K(A)$ is an indicator of whether a system of linear equations $A\mathbf{x} = \mathbf{b}$ is "good" or "bad"
- If $K(A)$ is small, it's "good": we call it well-conditioned
- If $K(A)$ is large, it's "bad": we call it ill-conditioned
- Example of ill-conditioned:

$$A = \begin{pmatrix} 1 & 100 \\ 0 & 2 \end{pmatrix} \qquad A^T A = \begin{pmatrix} 1 & 100 \\ 100 & 10004 \end{pmatrix}$$

with eigenvalues $\lambda_1 = 2$, $\lambda_2 = 1$ and singular values $\sigma_{\max} \approx 100$, $\sigma_{\min} \approx 0.02$ so $K(A) \approx 5002$.

For the earlier examples:

- $$A = \begin{pmatrix} 1 & 1 \\ 1 & -3 \end{pmatrix} \text{ has } K(A) \approx 2.6$$

  so the relative error in $\mathbf{x}_*$ when solving $A\mathbf{x} = \mathbf{b}$ is at most 2.6 times larger than the relative residual.

- $$B = \begin{pmatrix} 1 & 1 \\ 1 & .9 \end{pmatrix} \text{ has } K(A) \approx 38$$

  so the relative error in $\mathbf{x}_*$ when solving $B\mathbf{x} = \mathbf{b}$ can be as big as 38 times the relative residual.

- As a rule of thumb, if $K(A) \approx 10^q$, you can compute q digits less for $\mathbf{x}_*$ than you know for $\mathbf{b}$.

A good example: the Vandermonde matrix (see weeks 7-8):

The *Vandermonde matrix* is defined as

$$V_{ij} = x_{i-1}^{n-j+1} \text{ for } 1 \le i \le n+1 \text{ and } 1 \le j \le n+1$$

The *Vandermonde matrix* for $n = 4$ is: V =

$$\begin{bmatrix} x_0^4 & x_0^3 & x_0^2 & x_0 & 1 \\ x_1^4 & x_1^3 & x_1^2 & x_1 & 1 \\ x_2^4 & x_2^3 & x_2^2 & x_2 & 1 \\ x_3^4 & x_3^3 & x_3^2 & x_3 & 1 \\ x_4^4 & x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix}$$

Let $x_i = -1 + i\Delta$ for $i = 0, \ldots, n$ and $\Delta = 2/n$
(gives equally spaced points between -1 and 1).

For $n = 20$, $K(V) \approx 8 \times 10^8$.
Let $b_i = x_{i-1} - x_{i-1}^2$. for $1 < i < n+1$, then

$$Vx = b$$

has the *exact* solution

$$x = \mathbf{e}_{20} - \mathbf{e}_{19}$$

Numerically solving (see accuracy.py in the code repository):

```
>>> import scipy
>>> import scipy.linalg
>>> xs=scipy.linspace(-1,1,21)
>>> V=scipy.vander(xs)
>>> def f(x):
...     return x-x*x
>>> r=f(xs)
>>> s=scipy.linalg.solve(V,r)
```

$$x_* = \begin{pmatrix} 0.000000000244668 \\ 0.00000000000004811 \\ -0.000000000929254 \\ -0.0000000000023777 \\ 0.000000001457681 \\ 0.0000000000045536 \\ -0.000000001227802 \\ -0.0000000000043483 \\ 0.000000000604410 \\ 0.0000000000021749 \\ -0.000000000177176 \\ -0.00000000000005221 \\ 0.0000000000030097 \\ 0.000000000000000312 \\ -0.00000000000002738 \\ 0.0000000000000085 \\ 0.000000000000000115 \\ -0.000000000000000012 \\ -1.000000000000000002 \\ 1.000000000000000000 \\ 0.000000000000000000 \end{pmatrix}$$

$$\frac{\|\mathbf{b} - V\mathbf{x}\|_2}{\|\mathbf{b}\|_2} \approx 10^{-15}; \quad \frac{\|\mathbf{x} - \mathbf{x}_*\|_2}{\|\mathbf{x}\|_2} \approx 10^{-9}$$

Summary

- Norms: quantify lengths of / distances between vectors.
- Definitions of ℓ_1-, ℓ_2-, and ℓ_∞-norms.
- Linear equations: errors and residuals of computed solutions.
- Condition number $K(A)$: measure of the accuracy in solving $A\mathbf{x} = \mathbf{b}$.
  - Well-conditioned $K(A) \simeq 1$; ill-conditioned $K(A) \gg 1$.
  - $K(A)$ large $\Rightarrow$ limited accuracy in solving $A\mathbf{x} = \mathbf{b}$ numerically.
  - Computation through `numpy.linalg.cond`.