Lecture Notes:
Condition Numbers and Geometry

Paul Breiding and Elima Shehu

May 4, 2021

This document contains lecture notes for the course *Geometry and Condition* held during the summer 2021 at the Max-Planck-Institute for Mathematics in the Sciences Leipzig. Each chapter contains material for a 90 minutes lecture.

The lecture is in parts based on the book *Condition: The Geometry of Numerical Algorithms* by Bürgisser and Cucker [BC13].

Both authors have been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer 445466444.

Author's addresses:
Paul Breiding, MPI MiS, paul.breiding@mis.mpg.de.
Elima Shehu, MPI MiS, elima.shehu@mis.mpg.de.

Contents

Lecture 1

1.1 Motivation and definition of condition numbers

We start with a quote by [Dem96]

> "The correct answers produced by numerical algorithms are seldom exactly correct. There are two sources of error. First, there may be errors in the input data to the algorithm, caused by prior calculations or perhaps measurements errors. Second, there are errors caused by the algorithm itself, due to approximations made within the algorithm. In order to estimate the errors in the computed answers from both these sources, we need to understand how much the solution of a problem is changed, if the input data is slightly perturbed."

The first source of error that Demmel describes is a property of data. The second source is a property of algorithms.
Any algorithm has to cope with the first source of errors!

Example 1.1 (Exact algorithm)**.** Consider the following computational problem: on input $(A, b) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^2$ with $\det(A) \neq 0$ find $x \in \mathbb{R}^2$, such that $Ax = b$.
We consider two different inputs:
Input 1:

$$A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \quad \text{and} \quad b = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

An error in the data could have given us the input

$$\tilde{A} = \begin{pmatrix} 1 & 1 \\ -1+\epsilon & 1 \end{pmatrix}, \quad \text{and} \quad \tilde{b} = \begin{pmatrix} 2 \\ 0 \end{pmatrix},$$

where $\epsilon > 0$ is small. The *exact* solutions for the equations $Ax = b$ and $\tilde{A}\tilde{x} = \tilde{b}$ are

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \tilde{x} = \tilde{A}^{-1}\tilde{b} = \begin{pmatrix} 1 + \frac{\epsilon}{2-\epsilon} \\ 1 - \frac{\epsilon}{2-\epsilon} \end{pmatrix}.$$

Comparing the errors we find

$$\frac{\|x - \tilde{x}\|}{\left\|A - \tilde{A}\right\|} = \frac{1}{\epsilon} \cdot \frac{\sqrt{2}\epsilon}{2-\epsilon} = \frac{\sqrt{2}}{2-\epsilon} \approx \frac{1}{\sqrt{2}} \tag{1.1}$$

This shows that the error in the input $\left\|A - \tilde{A}\right\|$ is amplified in the out $\|x - \tilde{x}\|$ by a factor of $\frac{1}{\sqrt{2}}$. Next, we consider a second input.

Input 2:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1+10^{-8} \end{pmatrix}, \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 10^{-8} \end{pmatrix}.$$

Consider the following peturbation for a small $\epsilon > 0$.

$$\tilde{A} = \begin{pmatrix} 1 & 1 \\ 1+\epsilon & 1+10^{-8} \end{pmatrix}, \quad \text{and} \quad \tilde{b} = \begin{pmatrix} 0 \\ 10^{-8} \end{pmatrix}.$$

The exact solutions for the equations $Ax = b$ and $\tilde{A}\tilde{x} = \tilde{b}$ are

$$x = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \text{and} \quad \tilde{x} = \begin{pmatrix} -1 - \frac{\epsilon}{10^{-8}-\epsilon} \\ 1 + \frac{\epsilon}{10^{-8}-\epsilon} \end{pmatrix}.$$

This implies

$$\frac{\|x - \tilde{x}\|}{\left\|A - \tilde{A}\right\|} = \frac{1}{\epsilon} \frac{|\,\epsilon\,|\sqrt{2}}{|\,10^{-8} - \epsilon\,|} = \frac{\sqrt{2}}{|\,10^{-8} - \epsilon\,|}. \tag{1.2}$$

This shows that, if $\epsilon \leq 10^{-8}$, then we have $\frac{\|x-\tilde{x}\|}{\left\|A-\tilde{A}\right\|} > 10^{8}$.
Even though we applied an exact algorithm to the problem we got different quantities in the output:

Output 1: close to the exact solution

Output 2: is far from the exact solution

The theory of condition numbers explains these different behaviours of data with respect to perturbations. A general theory for the condition numbers was given by [Ric66]. But, What is a condition number? What a condition number measure? How is it defined? A condition number of a problem measures the sensitivity of the solution to small perturbations in the input data. The condition number depends on the problem and the input data, on the norm used to measure size, and on whether perturbations are measured in an absolute or a relative sense.

Definition 1.2. A computational problem is a function $f : I \longrightarrow O$ from a space of inputs I to a space of outputs O.

For the example from the above: the input space is $I = \{(A, b) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid \det(A) \neq 0\}$, the output space is $O = \mathbb{R}^2$, and $f(A, b) = A^{-1}b$.

Definition 1.3 (Classic definition of condition number; see, e.g., [TB97])**.** Let I and O be finite dimensional normed vector spaces. The (absolute) condition number of f at $x \in I$ is

$$\kappa[f](x) := \lim_{\epsilon \longrightarrow 0} \sup_{y \in I,\, \|x-y\| \leqslant \epsilon} \frac{\|f(x) - f(y)\|}{\|x - y\|} \tag{1.3}$$

We have the following properties of $\kappa[f]$.

Lemma 1.4. *Suppose that $I = \mathbb{R}^n$ and $O = \mathbb{R}^m$.*

1. *If f is differentiable, we have $\kappa[f](x) = \|J(x)\|$ where $J(x) = \left(\frac{\partial f_i}{\partial x_i}\right)$ is the Jacobian of the partial derivatives of $f = (f_1, f_1, \ldots, f_m)$ and $\|A\| := \max_{\|x\|=1} \|Ax\|$ is the operator norm.*

2. *In the limit* $\|x-y\| \to 0$ *we have* $\|f(x)-f(y)\| \leqslant \kappa[f]\|x-y\|$.

3. $\kappa[f \circ g](x) \leqslant \kappa[f](g(x)) \cdot \kappa[g](x)$ *for* $\kappa[g](x) \leqslant \infty$.

What does "small" means? If $\|x\| = 10^{-4}$, is $\|x-y\| = 10^2$ small or large?
The ambiguity motivates the following definition

Definition 1.5. The error between x and y relative to x is

$$\operatorname{RelError}(x,y) = \frac{\|x-y\|}{\|x\|}$$

Definition 1.6. The relative condition number is defined

$$\kappa_{\mathrm{REL}}[f](x) := \lim_{\epsilon \to 0} \sup_{\operatorname{RelError}(x,y) \leqslant \epsilon} \frac{\operatorname{RelError}(f(x), f(y))}{\operatorname{RelError}(x,y)} \tag{1.4}$$

Lemma 1.7. *If* f *is differentiable and* $x \neq 0$*, then*

$$\kappa_{\mathrm{REL}}[f](x) = \|J(x)\| \frac{\|(x)\|}{\|f(x)\|} \tag{1.5}$$

The condition numbers κ and κ_{REL} are called normwise condition numbers.

Definition 1.8. The componentwise relative condition number is defined as

$$\mathrm{CW}[f](x) := \max_j \lim_{\epsilon \to 0} \sup_{\max_i \operatorname{RelError}(x_i,y_i) \leq \epsilon} \frac{\operatorname{RelError}(f_j(x), f_j(y))}{\max_i \operatorname{RelError}(x_i,y_i)}$$

where $x = (x_i)$ and $f = (f_j)$.

Trefethen and Bau [TB97, p. 91]:

> "Both absolute and relative condition numbers have their uses, but the latter are more important in numerical analysis. This is ultimately because the floating point arithmetic used by computers introduces relative errors rather than absolute ones."

1.2 Floating point arithmetic

On a computer we can represent numbers using only a finite amount of information so we must work with approximations of real numbers.
The most commonly used number system on a computer are floating point numbers.

Definition 1.9. 1. A floating point number system $F \subseteq \mathbb{R}$ is a subset of the reals of the form: $F = \left\{ \pm \beta^e \sum_{i=1}^{t} \frac{d_i}{\beta_i} \mid 0 \leq d_i \leq \beta - 1, e_{\min} \leq e \leq e_{\max} \right\}$, where $\beta, t, e_{\min}, e_{\max}$ are integers.

- β is called the base
- t is called precision
- $[e_{\min}, e_{\max}]$ is called exponential range.

2. For $G := \left\{ \pm \beta^e \sum_{i=1}^{t} \frac{d_i}{\beta_i} \mid 0 \leq d_i \leq \beta - 1 \right\}$ we put

$$\mathrm{fl} : \mathbb{R} \to G, \quad x \mapsto \arg\min_{y \in G} |x - y|.$$

This is called the *rounding map.*

3. The range of F is $\mathrm{range}(F) := \{x \in \mathbb{R} \mid \beta^{e_{\min}-1} \leq |x| \leq \beta^{e_{\max}} (1 - \beta^{-1})\}$. All numbers in $\mathrm{range}(F)$ are approximated by relative precision $u := \frac{1}{2}\beta^{1-t}$.

Theorem 1.10. *For all* $x \in \mathrm{range}(F)$, *then* $\mathrm{fl}(x) = x(1+\delta) \in F, \quad |\delta| \leqslant u$

This theorem shows that every real number lying in $\mathrm{range}(F)$ can be approximated by an element of F with the relative error no larger than $u = \frac{1}{2}\beta^{1-t}$. So, we have:

$$\mathrm{RelError}(x, \mathrm{fl}(x)) = \frac{\|x - x(1+\delta)\|}{\|x\|} = \|\delta\| \leqslant u,$$

where u is called machine precision and $\epsilon_{\mathrm{MACH}} := \beta^{1-t}$ is called machine epsilon. For more details see [Ste97],[Knu98], and [Hig96].
The bound of the relative error can be refined, so that $\mathrm{RelError}(x, \mathrm{fl}(x)) \leqslant \frac{u}{1+u}$

Remark 1.11. In [JR18] optimal bounds on relative errors are established for each five basic operations. Each of these bounds is attained for some explicit input values in F and rounding functions under some mild conditions on β and t .

The IEEE 754 standart defines a floating point arithmetic system with $\mathrm{fl}\,(x \circ y) = (x \circ y)\,(1+\delta)$, $|\,\delta\,| \leqslant u$ where $\circ \in \{+,\,-,\,\times, /,\,\sqrt{\cdot}\}$ and formats.

	β	t	e_{min}	e_{max}	u
half (16 bit)	2	11	-14	$16 = 2^4$	$\approx 5 \cdot 10^{-4}$
single (32 bit)	2	24	-125	$128 = 2^7$	$\approx 6 \cdot 10^{-8}$
double (64 bit)	2	53	-1021	$1024 = 2^{10}$	$\approx 10^{-16}$

For instance, 64-bit floating point number system can approximate any real number within its range with a relative error of at most $u \approx 10^{-16}$.

Lecture 2

2.1 The loss of precison

Computing with finite-precision arithmetic such as floating-point arithmetic means we will face the negative effect of loss of precision. Recall that u is the machine precision, which the smallest relative difference between two numbers that the computer recognizes. For instance, 64-bit floating point arithmetic is based on $u \approx 10^{-16}$.

Definition 2.1 (Loss of precision)**.** Let β be the base for a floating point number system. The loss of precision in the computation of f is

$$\mathrm{LOP}\left(f, x\right) := \log_\beta \frac{\mathrm{RelError}\left(f\left(x\right), f\left(y\right)\right)}{u},$$

where $y = \mathrm{fl}\left(x\right)$.

The loss of precision tells us how many digits are lost in the computation of f in floating point arithmetic with base β. For instance, of $\mathrm{LOP}\left(f, x\right) = k$ it means that the the first k terms in β-adic expansions of $f(\mathrm{fl}(y))$ and $f(x)$ do not necessarily coincide.
Rule of thumb: $\mathrm{LOP}\left(f, x\right) \approx \log_\beta \kappa_{\mathrm{REL}}[f]\left(x\right)$.
Note that the loss of precision is a property of the problem f and the data x, but not of an algorithm computing $f(x)$. Any algorithm has to cope with loss of precision. An algorithm that for an input x computes accurately $f\left(y\right)$, where $\mathrm{RelError}(x, y)$ is small, is called backward stable.

2.2 Matrix-Vector-Multiplication

The material in this section is based on [BC13, Section O.4] and [Hig96, Section 3.5].
Let us illustrate condition numbers and floating point arithmetic with an example. The problem of matrix-vector-multiplication is modelled as on the following three maps:

1. $f : \mathbb{R}^{n\times n} \times \mathbb{R}^n \to \mathbb{R}^n$, $(A, x) \mapsto Ax$ (both A and x are variables);
2. $g : \mathbb{R}^{n\times n} \to \mathbb{R}^n$, $A \mapsto Ax$ (A is a variable, x is fixed);
3. $h : \mathbb{R}^n \to \mathbb{R}^n$, $x \mapsto Ax$ (A is fixed, x is a variable).

Depending on where we allow perturbations, we have to choose one of these maps for modelling matrix-vector-multiplication. If we allow perturbations in both A and x, we choose f. If we only allow perturbations in A, we choose g. If we only allow perturbations in x, we choose h. It seems reasonable to choose f for modelling the problem. However, the next theorem shows that we should actually choose g!
Regardless on whether or not we allow perturbations on A or x, let us assume that (A, x) is given exactly up to machine precision. Thus, we have the following theorem:

Theorem 2.2. *There is a finite precision, which on input $(A, x) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$ computes $\tilde{b}$. If $(\lceil \log_2 n \rceil + 2)^2 u < 1$, then $\tilde{b} = \tilde{A}x$ with*

$$\frac{|\, \tilde{a}_{ij} - a_{ij} \,|}{|\, a_{ij} \,|} \leq (\lceil \log_2 n \rceil + 2)\, u.$$

Before we prove this theorem, let us first discuss its implications: there exists a backward stable algorithm for computing matrix-vector-multiplication, and that this algorithm only takes into account errors in A. Even if the input data has errors in both A and x, the algorithm will produce the correct output for $(\tilde{A}, x)$ where $\mathrm{RelError}(A, \tilde{A})$ is bounded by $(\lceil \log_2 n \rceil + 2)\, u$ provided that $(\lceil \log_2 n \rceil + 2)^2 u < 1$. For 64-bit floating point arithmetic we have $u \approx 10^{-16}$ and so the range for n in this case is roughly $n < 2^{(10^8)}$. For instance, if $n = 10^4$, then the bound implied by the theorem is $\approx 2 \cdot 10^{-15}$.
The theorem justifies studying the condition of matrix-vector-multiplication through g.

We need an auxiliary lemma in the proof of the theorem.

Lemma 2.3. *Let m be an integer with $mu < 1$. Let $\delta^{(1)}, \ldots, \delta^{(m)}$ be real numbers with $| \delta_i | \leq u$ for each i. Then, we have $\prod_{t=1}^{m} \left(1 + \delta^{(t)}\right) = (1 + \theta_m)$ with $| \theta_m | < \frac{mu}{1-mu}$.*

Proof. Let $\delta := \max |\delta_i|$. Then, by assumption $|\theta_m| \leq (1+\delta)^m - 1$, and so

$$|\theta_m| \leq \sum_{k=1}^{m} \binom{m}{k} \delta^k \leq \sum_{k=1}^{m} (mu)^k = \frac{\left(\sum_{k=1}^{m} (mu)^k\right)(1-mu)}{1-mu} = \frac{mu(1-mu^m)}{1-mu} \leq \frac{mu}{1-mu},$$

the last inequality because $mu < 1$. This finishes the proof. □

Now, we can prove Theorem 2.2.

Proof of Theorem 2.2. Let $b = Ax$, then $b_i = \sum_{j=1}^{n} a_{ij} x_j$.
By definition of the rounding map fl we have $\mathrm{fl}\,(a_{ij}x_{ij}) = (a_{ij}x_{ij})\,(1+\delta_{ij})$, where $|\delta_{ij}| < u$. Adding the first to summands in the expansion of b_i in floating point arithmetic we get

$$\begin{aligned} \mathrm{fl}\,(a_{i1}x_1 + a_{i2}x_2) &= (a_{i1}x_1\,(1+\delta_{i1}) + a_{i2}x_2\,(1+\delta_{i2}))\,(1+\delta_{i12}) \\ &= a_{i1}x_1\,(1+\delta_{i1})\,(1+\delta_{i2}) + a_{i2}x_2\,(1+\delta_{i2})\,(1+\delta_{i12})\,, \end{aligned}$$

where $|\delta_{i12}| < u$. We can group the summands in the expansion of b_i in pairs of two. There, $\lceil \log_2 n \rceil$ many such summands. Thus, if we add the pairs, then add pairs of pairs and so on, we get an algorithm based on a binary tree that computes b_i. Let us put $m := \lceil \log_2 n \rceil + 1$. This is the number of subsequent floating point operators of our algorithm. The $+1$ is due to the fact that we also have to compute each $a_{ij}x_j$ in floating point arithmetic. By construction, we get

$$\mathrm{fl}\left(\sum_{j=1}^{n} a_{ij}x_j\right) = \sum_{j=1}^{n} a_{ij}x_j \prod_{t=1}^{m} \left(1 + \delta_{ij}^{(t)}\right)$$

for some numbers $\delta_{ij}^{(t)}$ with $|\delta_{ij}^{(t)}| < u$. Here, we have freely relabelled the indices of the $\delta_{ij}^{(t)}$ in comparison to above. In the following only the number of factors will be important.
Let $\tilde{a}_{ij} := a_{ij} \prod_{t=1}^{m} (1 + \delta_{ij}^{(t)})$ be the entries of the matrix $\tilde{A} := (\tilde{a}_{ij})$. Our algorithm computes the matrix-vector product $\tilde{b} := \tilde{A}x$.

By assumption, $(m+1)^2u < 1$ and so $mu < 1$. We can apply Lemma 2.3 to get

$$\frac{\mid a_{ij} - \tilde{a}_{ij} \mid}{\mid a_{ij} \mid} = \frac{mu}{1-mu}.$$

It remains to show that $\frac{mu}{1-mu} < (m+1)u$. By assumption, we have $(m+1)^2u < 1$. This implies $0 < 1 - m(m+1)u = 1 - mu - m^2u$, which is equivalent to $m < (1+m)(1-mu)$. Multiplying both sides by $\frac{u}{1-mu}$ gives $\frac{mu}{1-mu} < (m+1)u$ as desired. This finishes the proof. □

The theorem tells us that for matrix-vector-multiplication we can focus on the case where there are only errors in the matrix A but not in x. We therefore consider the condition number of the map $g : \mathbb{R}^{n\times n} \to \mathbb{R}^n, A \mapsto Ax$ for fixed x. We study the normwise condition number and componentwise condition number of this map.

2.2.1 Normwise condition number of matrix-vector-multiplication

Proposition 2.4. *Measuring the error in the in- and output space of g with the Euclidean norm we get the normwise absolute condition number $\kappa[g](A) = \|x\|$. The normwise relative condition number is $\kappa_{\text{REL}}[g](A) = \frac{\|x\|\|A\|}{\|Ax\|}$*

Proof. We use the formula $\kappa[g](A) = \|Jg(A)\|$, where Jg is the jacobian matrix of first order partial derivatives of g. Let's look at the derivative of

$$g(A) = Ax = \begin{pmatrix} a_1^T x \\ a_2^T x \\ \vdots \\ a_n^T x \end{pmatrix}$$

where a_i are rows of A. We can also write

$$g(A) = \begin{pmatrix} x^T a_1 \\ \vdots \\ x^T a_n \end{pmatrix}$$

Then, the partial derivative of g with respect to a_{ij} is x_j- This shows the following formula for the Jacobian:

$$Jg\left(A\right) = \begin{pmatrix} x^T & \cdots & 0 \\ 0 & x^T \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x^T \end{pmatrix} \in \mathbb{R}^{n \times n^2}$$

And by definition $\|Jg\left(A\right)\| = \max_{y \in \mathbb{R}^{n^2}, \|y\|=1} \|Jg\left(A\right) y\|$. We partition the $\mathbb{R}^{n^2}$ into blocks of size n and write $y = (y_1, y_2, \cdots, y_n)$, where $y_i \in \mathbb{R}^n$. Then:

$$\|Jg\left(A\right) y\| = \left\| \begin{pmatrix} x^T y_1 \\ \vdots \\ x^T y_n \end{pmatrix} \right\| = \sqrt{\left(x^T y_1\right)^2 + \cdots + \left(x^T y_n\right)^2}$$

We know that $\mid x^T y_i \mid \leq \|x\| \, \|y\|$ by Cauchy-Schwartz and that this inequality is sharp. This fact implies:

$$\max_{y_i, \|y\|=1} \sqrt{\|x\|^2 \|y_1\|^2 + \cdots + \|x\|^2 \|y_n\|^2} = \|x\| \cdot \max_{\|y\|=1} \|y\| = \|x\|$$

This shows the claim for $\kappa[g]\left(A\right)$. For the relative condition number we use the formula $\kappa_{\text{REL}}[g]\left(A\right) = \kappa[g]\left(A\right) \cdot \frac{\|A\|}{\|Ax\|}$. This finishes the proof. □

2.2.2 Componentwise condition number of matrix-vector-multiplication

Proposition 2.5. *The relative componentwise condition number satisfies*

$$\text{CW}[g]\left(A\right) \leq \max_i \frac{1}{\mid \cos \sphericalangle \left(a_i, x\right) \mid},$$

where $a_1, \ldots, a_n$ are the rows of A and $\sphericalangle$ denotes the angle between two vectors.

Proof. Let $\tilde{A} = A + E$, where E is the error in A. Let us write $b := Ax$ and $\tilde{b} := \tilde{A}x$. Recall

from Definition 1.8 that the relative componentwise condition number of g at A is

$$\mathrm{CW}\left[g\right](A) = \max_{k} \lim_{\epsilon \to 0} \sup_{\max_{i,j} \mathrm{RelError}(a_{ij}, \tilde{a}_{ij}) \leq \epsilon} \frac{\mathrm{RelError}(b_k, \tilde{b}_k)}{\max_{i,j} \mathrm{RelError}\left(a_{ij}, \tilde{a}_{ij}\right)}, \tag{2.1}$$

where $A = (a_{ij})$, $\tilde{A} = (\tilde{a}_{ij})$ and $b = (b_k)$, $\tilde{b} = (\tilde{b}_k)$. Then, by definition,

$$\mathrm{RelError}\left(a_{ij}, \tilde{a}_{ij}\right) = \frac{\mid e_{ij} \mid}{\mid a_{ij} \mid},$$

where $E = (e_{ij})$. This implies $|e_{ij}| \leq \mathrm{RelError}\left(a_{ij}, \tilde{a}_{ij}\right) |a_{ij}|$ for all all pairs of indices (i, j).

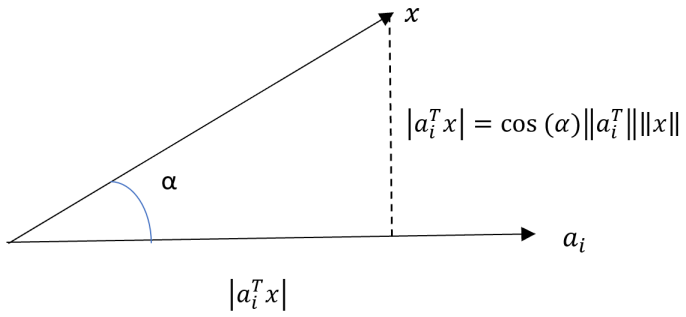


Figure 2.1: Sketch of the geometry in the proof of Proposition 2.5.

Let a_i be the rows of A and e_i are the rows of E. Taking norms of the k-th row vectors we can get the following inequality:

$$\|e_k\| \leq \|a_k\| \max_{i,j} \mathrm{RelError}\left(\tilde{a}_{ij}, \tilde{a}_{ij}\right), \tag{2.2}$$

We also have:

$$\mathrm{RelError}(b_k, \tilde{b}_k) = \frac{\mid e_k^T x \mid}{\mid a_k^T x \mid} \leq \frac{\|e_k\| \, \|x\|}{\mid a_k^T x \mid} \leq \frac{\|a_k\| \, \|x\| \max_{i,j} \mathrm{RelError}\left(a_{ij}, \tilde{a}_{ij}\right)}{\mid a_k^T x \mid};$$

the first inequality by Cauchy-Schwartz and the second by (2.2).

Combined this implies

$$\frac{\text{RelError}(b_k, \tilde{b}_k)}{\max_{i,j} \text{RelError}\left(a_{ij}, \tilde{a}_{ij}\right)} \leq \left| \frac{a_k^T x}{\|a_k\| \, \|x\|} \right|^{-1}.$$

Plugging this into (2.1) we see that

$$\text{CW}[g]\left(A\right) \leq \max_i \frac{1}{|\, \frac{a_i^T x}{\|a_i\| \|x\|} \,|} = \max_i \frac{1}{|\, \cos \sphericalangle \left(a_i, x\right) \,|}.$$

This finishes the proof. □

2.3 Ill-posedness

Let us look again at the general setting where we we have a map $f : I \to O$ between a set of input I and outputs O.

Definition 2.6. The set

$$\Sigma_\mu := \{x \in I \mid \mu[f]\left(x\right) = \infty\}$$

for either $\mu = \kappa$, $\mu = \kappa_{REL}$ or $\mu = \text{CW}$ is called the set of ill-posed inputs. If the condition number μ is clear from the context, we also omit the subscript and simply write Σ.

In the case of matrix vector multiplication the ill-posed inputs are the following sets:

$$\begin{aligned}
\Sigma_\kappa &= \emptyset; \\
\Sigma_{\kappa_{\text{REL}}} &= \{A \in \mathbb{R}^{n \times n} \mid Ax = 0\} \quad ; \\
\Sigma_{\text{CW}} &= \{A \in \mathbb{R}^{n \times n} \mid \exists i : a_i^T = 0\} \\
&= \{A \in \mathbb{R}^{n \times n} \mid \exists i : a_i \in x^\perp\};
\end{aligned}$$

the first and the second equation are by by Proposition 2.4 and the third is by Proposition 2.5. All of these are real algebraic varieties!

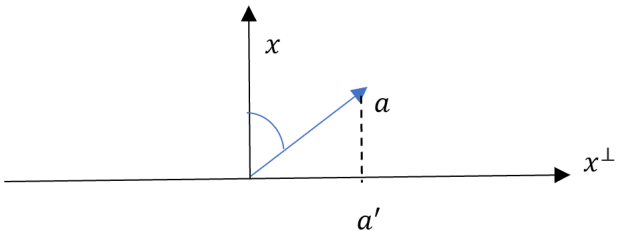
We consider in more detail the CW-condition number and Σ_{CW}. Recall:

$$\mathrm{CW}[g]\,(A) \leq \max_i \frac{1}{|\cos \sphericalangle\,(a_i, x)\,|}.$$

For fixed i, let a_i be the orthogonal projection of a_i onto $x^\perp = \{y \in \mathbb{R}^n \mid x^T y = 0\}$. Then:

$$|\cos \sphericalangle\,(a_i, x)\,| = \frac{\|a_i - a_i'\|}{\|a_i\|} = \min_{y \in x^\perp} \frac{\|a_i - y\|}{\|a_i\|}.$$

This relation is depicted in the picture just below.



This shows that

$$\mathrm{CW}[g]\,(A) \leq \max_i \frac{\|a_i\|}{\min\limits_{y \in x^\perp} \|a_i - y\|} = \max_i \frac{\|a_i\|}{\mathrm{dist}\,(a_i, x^\perp)} = \max_i \frac{\|a_i\|}{\mathrm{dist}\,(a_i, \Sigma_{\mathrm{CW}})},$$

where dist is the usual distance from a point to a set in the Euclidean metric. We have shown that the CW-condition number at A is bounded from above by the normalized distance of A to ill-posedness. This is a first instance of an equation called condition number theorem. Such theorems relate condition numbers to inverse distance to ill-posedness. We will meet other examples of this later in this lecture.

2.4 Global analysis of condition numbers

The previous discussion has shown that condition numbers can be arbitrary large. Consequently, with finite precision arithmetic we can't compute for all instances $x \in I$ their

output $f(x)$. The worst-case analysis can be

$$\sup_{x \in I} \mu(x) = \infty \quad \text{for} \quad \mu \in \{\kappa, \kappa_{\text{REL}}, CW\}.$$

However, it could be the case that $\mu(x) = \infty$ is extremely rare.

This motivates the following alternative global measures of condition:

Worst-Case Analysis:	$\sup_{x \in I} \mu[f](x)$
Average Analysis:	$\mathbb{E}_{x \sim d} \mu[f](x)$, where d is some probability distribution on the space of inputs I.
Smoothed Analysis:	$\sup_{\overline{x} \in I} \mathbb{E}_{x \sim \text{Unif}(B(\overline{x}, \sigma))} \mu[f]$, where $B(\overline{x}, \sigma)$ is the ball of radius σ centered at x and $\text{Unif}(B(\overline{x}, \sigma))$ is the uniform distribution.
Without the Black Swans:	$\mathbb{E}_{x \sim d}[\mu[f](x) \mid x \notin W]$, where d is some probability distribution on the space of inputs I and $W \subset I$ is a set of small measure.

We will discuss average and smoothed analysis in later lectures.

Bibliography

[BC13] P. Bürgisser and F. Cucker. *Condition: The Geometry of Numerical Algorithms*, volume 349 of *Grundlehren der mathematischen Wissenschaften*. Springer, Heidelberg, 2013.

[Dem96] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1996.

[Hig96] N. J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics, second edition, 1996.

[JR18] C.-P. Jeannerod and S. M. Rump. On relative errors of floating-point operations: optimal bounds and applications. *Mathematics of Computation*, 87:803–819, 2018.

[Knu98] D. E. Knuth. *The Art of Computer Programming*, volume 2. SIAM, 3 edition, 1998.

[Ric66] John R. Rice. A theory of condition. *SIAM J. Numer. Anal*, 3:287–310, 1966.

[Ste97] P. H. Sterbenz. *Floating-Point Computation*. SIAM, 1997.

[TB97] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.