MA 580; Numerical Analysis I

C. T. Kelley
NC State University
tim_kelley@ncsu.edu
Version of September 5, 2016

NCSU, Fall 2016
Part II: Notation, Background, Errors
Contents

1 Vectors and Matrices
2 Norms
3 Eigenvalues and Eigenvectors
4 Special Matrices
5 Errors and Floating Point Arithmetic
6 Stability and Conditioning
Notation: Vectors

- \( \mathbb{R}^N \) is the space of \( N \) dimensional vectors.
- \( \mathbb{R}^{M \times N} \) is the space of \( M \times N \) matrices.
- Vectors \( \mathbf{x} \) are column vectors

\[
\mathbf{x} = \begin{pmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_N
\end{pmatrix} = (x_1, \ldots, x_N)^T.
\]
Matrices $A$ are $M \times N$. The $ij$th component is $a_{ij}$ or, if $A$ is a complex expression $A_{ij}$.

If $a_j \in \mathbb{R}^M$ is the $j$th column of $A$, we can write $A$ as

$$A = (a_1, \ldots, a_N).$$
Matrix Products

If $A$ is $M \times L$ and $B$ is $L \times N$, the matrix product $C = AB$ is $M \times N$ and

$$c_{ij} = \sum_{l=1}^{L} a_{il} b_{lj}$$

Example: scalar product

$$v^T u = \sum_{i=1}^{N} v_i u_i$$
More Matrix Notation

- The transpose of $A \in \mathbb{R}^{M \times N}$ is $A^T \in \mathbb{R}^{N \times M}$ and $A^T_{ij} = A_{ji}$. So, the transpose of a column vector is a row vector.
- The identity matrix $I \in \mathbb{R}^{N \times N}$: $Ix = x$ for all $x$.

$$ (I)_{ij} = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} $$

- Define the inverse of $A \in \mathbb{R}^{N \times N}$, $A^{-1}$: $A^{-1}A = AA^{-1} = I$
Canonical vectors

We call the columns of $\mathbf{I}$

$$\mathbf{I} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = (\mathbf{e}_1 \quad \mathbf{e}_2 \quad \ldots \quad \mathbf{e}_n)$$

the canonical vectors.

Warning: sometimes $\mathbf{e}$ is also an error. You’ll figure it out from the context.
Canonical Vectors

\[ \mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \cdots, \quad \mathbf{e}_N = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}. \]
Sequences of Vectors

If \{x_k\} is a sequence of vectors, the \(i\)th component of \(x_k\) is

\[ x_{ik} \]

as it would be if you regarded the sequence as a large matrix

\[ X = (x_1, x_2, \ldots) \]

and used \(i\) as the row index and the iteration counter as the column index.
Vector Norms

Review the definition of norm if you don’t remember it. The three important norms are $\ell^1$, $\ell^2$, $\ell^\infty$.

\[
\|x\|_1 = \sum_{i=1}^{N} |x_i|, \quad \|x\|_2 = \left( \sum_{i=1}^{N} |x_i|^2 \right)^{1/2}, \text{ and } \|x\|_\infty = \max_i |x_i|.
\]

More generally, the $\ell^p$ norm:

\[
\|x\|_p = \left( \sum_{i=1}^{N} |x_i|^p \right)^{1/p}
\]
for $1 \leq p < \infty$.

Generally the specific norm will not be so important. If I don’t say what the norm is then it is any vector norm.
We will consider matrix norms which are **induced** by vector norms. This means

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

For \( A \in \mathbb{R}^{N \times N} \), induced norms satisfy

\[ \|AB\| \leq \|A\|\|B\| \]

and

\[ \|Ax\| \leq \|A\|\|x\| \]

The norm of the product is less than or equal to the product of the norms.
Computing the $\ell^\infty$ matrix norm: 1

Do the math (absolute value of sum $\leq$ sum of absolute values) to get

$$\left| (Ax)_i \right| = \left| \sum_{j=1}^{N} a_{ij} x_j \right| \leq \sum_{j=1}^{N} |a_{ij}| |x_j| \leq \|x\|_\infty \max_i \sum_{j=1}^{N} |a_{ij}|$$

for all $x$ and all $i$. So,

$$\|A\|_\infty \leq \max_i \sum_{j=1}^{N} |a_{ij}| = \text{maximum absolute row sum}$$

We’ll show that the norm is the maximum absolute row sum.
Computing the $\ell^{\infty}$ matrix norm: II

To show that the $\leq$ is really $=$. You’ll need to find a vector that attains the bound. Here it is.

Let $i^*$ be the row of $A$ with max norm

$$
\sum_{j=1}^{N} |a_{i^*j}| = \max_{i} \sum_{j=1}^{N} |a_{ij}|
$$

and let $x_{j} = \text{sign}(a_{i^*j})$. Then $\|x\| = 1$ and

$$
(Ax)_{i^*} = \sum_{j=1}^{N} |a_{i^*j}|
$$

SHAZAM!
\( \ell^1 \) matrix norm?

\[
\|A\|_1 = \max_j \sum_{i=1}^N |a_{ij}|
\]

the max absolute column sum.
You should be able to prove this.
We’ll do \( \ell^2 \) later.
\( \lambda \) is an eigenvalue of \( A \) with corresponding eigenvector \( x \) if

\[
Ax = \lambda x
\]

Any eigenvalue is a root of the characteristic polynomial of \( A \)

\[
p_C(z) = \det(zI - A)
\]

\( p_C \) has \( N \) roots by the fundamental theorem of algebra. So there are \( N \) eigenvalues, counted with multiplicity.
The **algebraic multiplicity** of an eigenvalue $\lambda$ is the number of times it appears in the factorization of $p_C$ into linear terms

$$p_C(z) = \prod_{i=1}^{N} (z - \lambda_i)$$

The **geometric multiplicity** of $\lambda$ is the dimension of the null space of $\lambda I - A$.

The two notions of multiplicity are not the same. Take

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

please.
Diagonal Matrices

A matrix $D$ is diagonal if $d_{ij} = 0$ unless $i = j$.

$$
D = \begin{pmatrix}
  d_{11} & & \\
  & \ddots & \\
  & & d_{nn}
\end{pmatrix}.
$$

In this case we sometimes abbreviate $d_{ii}$ as $d_i$ and write

$$
D = \text{diag}(d_1, \ldots, d_N)
$$

where $d_i$ is the $i$th entry in the diagonal.

Example: $I = \text{diag}(1, 1, \ldots, 1)$
A matrix is diagonalizable if it has a basis of eigenvectors $\{v_i\}_{i=1}^N$. In this case the matrix $V = (v_1, \ldots, v_N)$ is nonsingular and

$$A = V \Lambda V^{-1}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N)$. 
A ∈ $R^{N×N}$ is **upper triangular** if $a_{ij} = 0$ for $i > j$. That is,

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ & \ddots & \ddots \\ & & a_{nn} \end{pmatrix}.$$  

A is lower triangular if $A^T$ is upper triangular.
An $N \times N$ matrix $U$ is orthogonal if

$$U^{-1} = U^T$$

This means that the columns of $U$ form an orthonormal basis. Orthogonal matrices are a very big deal in this course.
Symmetric Matrices

\( A \) is symmetric is \( A = A^T \).

A few facts:

- The eigenvalues \( \{ \lambda_i \}_{i=1}^N \) of \( A \) are real.
- There’s an orthonormal basis \( \{ u_i \}_{i=1}^N \) of eigenvectors of \( A \).
- \( U = (u_1, \ldots, u_N) \) is an orthogonal matrix.
- \( A = U \Lambda U^T \) (spectral theorem for symmetric matrices)

Prove some of this stuff.
A symmetric matrix $A$ is symmetric positive definite (spd) if
\[ x^T A x > 0 \text{ for all } x \neq 0 \]

Alternatively, $A$ is spd if all its eigenvalues are positive.

Positive semidefinite: all eigenvalues nonnegative or
\[ x^T A x \geq 0 \text{ for all } x \]
Rank-one or Outer Product Matrices

Recall the **inner product** of \( \mathbf{v} \) and \( \mathbf{u} \)

\[
\mathbf{u}^T \mathbf{v} = (1 \times N)(N \times 1) = (1 \times 1) = \sum_{i=1}^{N} u_i v_i.
\]

The **outer product** is

\[
\mathbf{uv}^T = (N \times 1)(1 \times N) = N \times N,
\]

and \((\mathbf{uv}^T)_{ij} = u_i v_j\)
Rank One Matrices

- A is rank-one matrix if the range $R(A)$ of $A$ has dimension one.
- Fact: $A$ is rank-one if and only if $A = uv^T$ for vectors $u$ and $v$. You get to prove this in the homework.
Eigen-decomposition of symmetric matrices

If $A = A^T$ then

$$A = U \Lambda U^T = \sum_{i=1}^{N} \lambda_i u_i u_i^T.$$  

From this you can prove, for symmetric $A$,

$$\|A\|_2 = \max_i |\lambda_i|$$

You’ll see things like this on exams.
Special Matrices

Eigenvalues/vectors of Rank-One matrices

\[ A = xy^T \]

Any vector in \( E = \{z \mid y^T z = 0\} \) is a null-vector of \( A \) \((Ax = 0)\).

If \( y^T x \neq 0 \) \((x \not\in E)\) then

\[ Ax = (y^T x)x \]

so \( \lambda = y^T x \) is an eigenvalue with eigenvector \( x \).

Algebraic multiplicity = geometric multiplicity!

What if \( y^T x = 0? \)
Special Matrices

$\ell^2$ norm of symmetric matrix $A = U^T U$

Since $U$ is orthogonal then every $x \in \mathbb{R}^N$ has the expansion

$$x = \sum_i (u_i^T x) u_i,$$

which implies that

$$Ax = \sum_i \lambda_i (u_i^T x) u_i.$$

So,

$$\|Ax\|_2^2 = \sum_i \lambda_i^2 (u_i^T x)^2 \leq \lambda_N^2 \sum_i (u_i^T x)^2 = \lambda_N^2 \|x\|^2,$$

with equality if $x = u_N$. That’s it.
\[ \|Ax\|_2^2 = (Ax)^T(Ax) = x^T(A^TA)x. \]

\( A^TA \) is symmetric positive semidefinite, so

\[ A^TA = U\Lambda U^T \text{ where } 0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N. \]

This means (you should fill in the details) that

\[ \|Ax\|_2^2 \leq \lambda_N \|x\|^2 \text{ with equality if } x = u_N. \]

So \( \|A\|_2 = \sqrt{\lambda_N} \) where \( \lambda_N \) is the largest eigenvalue of \( A^TA \).
The Sherman-Morrison Formula: Rank-one changes

Assume \( A \in \mathbb{R}^{N \times N}; \mathbf{v}, \mathbf{u} \in \mathbb{R}^N \) and

- \( A \) is nonsingular,
- \( 1 + \mathbf{v}^T A^{-1} \mathbf{u} \neq 0. \)

Then \( A + \mathbf{u} \mathbf{v}^T \) is nonsingular and

\[
(A + \mathbf{u} \mathbf{v}^T)^{-1} = \left( I + \frac{(A^{-1} \mathbf{u}) \mathbf{v}^T}{1 + \mathbf{v}^T A^{-1} \mathbf{u}} \right) A^{-1}.
\]

How do you prove this? It’s your problem in the homework.
Things to think about

- In what sense is the spectral decomposition for a symmetric matrix $A = U \Lambda U^T$ unique? What about the $1 \times 1$ case? What about the $2 \times 2$ identity?
- What are the eigenvalues of an orthogonal matrix?
- Is an orthogonal matrix diagonalizable?
- What are the orthogonal spd matrices?
Relative and Absolute Errors

Let \( \mathbf{x} \) be a vector, matrix, scalar, \ldots Suppose you approximate \( \mathbf{x} \) by \( \tilde{\mathbf{x}} \). The **absolute** error is

\[
\| \mathbf{x} - \tilde{\mathbf{x}} \| \text{ sensitive to units}
\]

The **relative** error (for \( \mathbf{x} \neq 0 \)) is

\[
\frac{\| \mathbf{x} - \tilde{\mathbf{x}} \|}{\| \mathbf{x} \|} \text{ not sensitive to units}
\]

It’s ok to think of a relative error of \( 10^{-k} \) as meaning that \( \mathbf{x} \) and \( \tilde{\mathbf{x}} \) agree to \( k \) decimal digits.
A real number is an abstraction. It does not physically exist.

- There are uncountably many real numbers.
- There are uncountably many real numbers between any two real numbers.
- You cannot store all the real numbers in a physical device.

On the other hand . . .
A floating point number is like a dog.

It is not an abstraction. It is realer than a real number.
- It consumes energy.
- It radiates heat.
- It occupies space.
- It makes noise.
Floating point numbers are expressed in three fields in \textit{radix} (or base) \( P \) arithmetic.

Modern computers use radix 2 (binary) arithmetic. The standard is IEEE-754 (1985).

We will use radix 10 (decimal) for simple examples in this part of the course.

Historical systems have used 16 (hexadecimal) and 8 (octal).
Geography of Floating Point Numbers: The Fields

<table>
<thead>
<tr>
<th>±</th>
<th>Exponent</th>
<th>Mantissa</th>
</tr>
</thead>
</table>

- Sign bit.
- Exponent field
- Mantissa, significand, or fraction.
- Conventions for uniqueness: normalized numbers
  - The point is before the leading digit.
  - The leading digit of a non-zero number is non-zero.

You can think of this as scientific notation with limited range.
How big is an IEEE float?

<table>
<thead>
<tr>
<th></th>
<th>Width (bits)</th>
<th>Sign</th>
<th>Exponent</th>
<th>Manitssa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double</td>
<td>64</td>
<td>1</td>
<td>11</td>
<td>52</td>
</tr>
<tr>
<td>Single</td>
<td>32</td>
<td>1</td>
<td>8</td>
<td>23</td>
</tr>
</tbody>
</table>

- Matlab default is 64 bit IEEE double precision.
- Special objects:
  - INF 1/0, $10^{1000}, \ldots$
  - NaN 0/0, INF * 0, missing data in statistics

If you get these, you have a bug!
I have a base 10 brain. What does this mean to me?

In double precision:

- **Exponent Range:**
  - Largest > 0 floating point number: \( \approx 1.8 \times 10^{308} \)
  - Smallest normalized > 0 floating point number:
    \( \approx 2.22 \times 10^{-308} = 2.22e-302 \)
  - Denormalized: \( 4.9407e-324, \text{eps}(0) \)

- **Unit roundoff:** \( \epsilon_u \approx 1.11 \times 10^{-16} \)
  - Relative error bound for conversion from real to float and elementary operations.

- **Difference between 1 and floating point number > 1 nearest to 1:** \( 2.2204e-16 \)
  - \( \text{eps or eps}(1) \)
Why would anyone use single precision?

Ideas?
What you need to remember

- There are finitely many floating point numbers.
- Floating point numbers are not equally spaced.
- There's a role (but not in MA580) for single (and lower) precision.
- If computation arrives at
  - something larger than the largest float: **overflow**, get INF
  - something $0 < x < \text{eps}(0)$: **underflow**
    - gradual or zero
- Overflow is a problem. Underflow is usually not.
Rounding and Floating Point Operations

- Real number: $x$, Floating point representation
  \[ fl(x) = x(1 + \epsilon), \ |\epsilon| \leq \epsilon_u \]
- IEEE 754 default: round to nearest.
- Operations: Given $x, y \in R$ and elementary operation $\circ$
  \[ fl(x \circ y) = fl(fl(x) \circ fl(y)) = (x \circ y)(1 + \epsilon_u) \]

Given something to evaluate $f(x)$ you hope that

\[ \|fl(f(x)) - f(x)\| = \epsilon_f \|f(x)\|, \text{ where } \epsilon_f \approx \text{sizeof(computation)} \epsilon_u \]
A Toy Floating Point Number System

Here’s a radix 10 system.

- Two digit mantissa
- Exponent range: -2:2
- Largest float = \(0.99 \times 10^2 = 99\)
- \(\varepsilon(0) = 0.10 \times 10^{-2}\)
- \(\varepsilon(1) = 1.1 - 1 = 0.1\)
- Round to nearest.
Given \( \mathbf{x} = (10, 10)^T \) compute \( \|\mathbf{x}\|_2 \).

Answer = \( \sqrt{100 + 100} = \sqrt{200} \) which rounds to \( .14 \times 10^2 \), a legal floating point number.

Wrong way: Add \( x_1y_1 \) (Overflow) to \( x_2y_2 \) (Overflow) and take the square root.

Right way:

- Normalize \( \mathbf{x} \) to get \( \mathbf{w} = \mathbf{x} / \|\mathbf{x}\|_\infty = (1, 1)^T \)
- Compute \( \|\mathbf{w}\|_2 = \sqrt{2} \) rounds to \( .14 \times 10^1 \).
- Multiply \( \|\mathbf{w}\| \) by \( \|\mathbf{x}\|_\infty = 10 \) to get \( \|\mathbf{x}\|_2 = .14 \times 10^2 \)
Order of summation

Let's compute

\[ \sum_{i=1}^{101} a_i, \text{ where } a_1 = 90, a_2 = a_3 = \ldots a_{101} = .01 \]

If do we this in order

\[ fl(a_1 + a_2) = fl(90.01) = 90, \ldots \]

and the summation stagnates with a result of 90 (wrong).

Do it backwards and

\[ fl\left( \sum_{i=101}^{2} .01 \right) = fl(1) = 1, \text{ and } fl(1 + 90) = 91. \]

Moral: sum the small things first.
The Quadratic Formula

From your early childhood, the solutions of

\[ ax^2 + bx + c = 0 \]

are

\[ r_\pm = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

Let’s break it.
Catastrophic Cancellation or Loss of Significance

Let's find the roots of

\[ x^2 + 4x + .1 \]

with the quadratic formula. The answer (computed in IEEE double) is

\[ r_{\pm} = \frac{-4 \pm \sqrt{16 - .4}}{2} \approx \begin{cases} -.02515 & + \\ -3.975 & - \end{cases} \]

In the toy floating point system, these round to

\[ -.25 \times 10^{-1} \text{ and } - .40 \times 10^1 \]
Apply the Formula in Toy Floats

Since

\[ fl(16 - .4) = fl(15.6) = 16 \]

we may be in trouble. When we compute \( r_- \) we get

\[ r_- = (-4 - 4)/2 = -4 \text{ right!} \]

But for \( r_+ \) we are lost

\[ r_+ = (-4 + 4)/2 = 0 \text{ wrong!} \]

The subtraction 16 \(-\).4 lost all information.
The Fix

Before doing anything else, rewrite the formula for $r_+$ to avoid the subtraction.

$$r_+ = r_+ \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} = -\frac{4ac}{2a(b + \sqrt{b^2 - 4ac})}

For our problem this is

$$\frac{-0.4}{2(8)} = -\frac{1}{4}

which rounds to $-0.25 \times 10^{-1}$. SHAZAM!
Stability

An algorithm is unstable when applied to a problem if the size of the output grows exponentially as some measure of problem complexity increases.

This is not a precise definition, but
You’ll know it when you see it.
An unstable method.

Here’s a way to compute eigenvalue-eigenvector pairs. Let $A$ be symmetric with eigenvalues
$|\lambda_1| \leq |\lambda_2| \cdots \leq |\lambda_{N-1}| < |\lambda_N|$.

- Pick a large $n$ and $x \in \mathbb{R}^N$
- Compute $w = A^n x$
- Set $u_n = w / \|w\|$; $\lambda_n = u_n^T A u_N$

**Theorem:** $u_n \to u$ and $A u = \lambda_N u$.
This algorithm is unstable even if $N = 1$! Consider $A = 3$. 
But why do you care?

Doesn’t the huge size of $\mathbf{w}$ go away when you divide by $\|\mathbf{w}\|$?

- Yes, it does in exact arithmetic,
- but not in floating point.
  You get an overflow from an intermediate step.

But you can fix this one: Pick $\mathbf{x} \in \mathbb{R}^N$

- While not happy
  - $\mathbf{w} = \mathbf{A}\mathbf{x}$
  - $\mathbf{x} = \mathbf{w}/\|\mathbf{w}\|$ 

Same results, but normalizing each time.
Fix for instability: change algorithm.
A computation is **poorly conditioned** if a small relative change in the input produces a large relative change in the output.

**Define small!!!**

This is yet another “I’ll know it when I see it.” kind of thing.

You’ll get inaccurate results for a sufficiently poorly conditioned computation no matter what you do.

**Fix for poor conditioning: reformulate.**

A problem is **well conditioned** if small relative changes in the input lead to small relative changes in the output.

There is, of course, a large and fuzzy middle ground.
Condition number

We can quantify it this time

\[ \kappa = \text{"condition number"} = \frac{\text{norm of relative change in output}}{\text{norm of relative change in output}}. \]

If \( \mathbf{x} \) is the input, \( \mathbf{S}(\mathbf{x}) \) is the output, and \( \delta \mathbf{x} \) is the change in \( \mathbf{x} \).

Then

\[ \kappa = \frac{\| \mathbf{S}(\mathbf{x} + \delta \mathbf{x}) - \mathbf{S}(\mathbf{x}) \| / \| \mathbf{S}(\mathbf{x}) \|}{\| \delta \mathbf{x} \| / \| \mathbf{x} \|}. \]

Note: \( \kappa \) depends on what you’re doing \( \mathbf{S} \) and where you’re doing it \( \mathbf{x} \).
Example: subtraction

Here

\[ \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{\delta x} = \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}, \text{ and } \mathbf{S(x)} = x - y \]

So

\[ |\mathbf{S(x + \delta x)} - \mathbf{S(x)}| = |\delta x - \delta y|, \]

Use the \( \ell^1 \) norm and

\[ ||\mathbf{\delta x}|| = |\delta x| + |\delta y|, \text{ and } ||\mathbf{x}|| = |x| + |y|. \]
Conditioning of Subtraction

Plug it all into the formula and . . .

\[ \kappa = \frac{|\delta_x - \delta_y|/|x - y|}{|\delta_x| + |\delta_y|/(|x| + |y|)}. \]

There is no reason to expect \( \delta_x \) and \( \delta_y \) to cancel, so the most sensible estimate is

\[ \kappa = \frac{|x| + |y|}{|x - y|}. \]

Wow! Subtraction of nearly equation numbers is a bad idea again!!!
Conditioning of Matrix-Vector Product

Data: $A$ fixed, $x$, $\delta x$.

$$\kappa = \frac{\|A(x + \delta x) - Ax\|/\|Ax\|}{\|\delta x\|/\|x\|}.$$ 

We’ll use the estimate

$$\|A^{-1}\|^{-1}\|x\| \leq \|Ax\| \leq \|A\|\|x\|.$$ 

We know the one on the right because norm of product $\leq$ product of norms. The one on the left follows from

$$\|x\| = \|A^{-1}Ax\| \leq \|A^{-1}\|\|Ax\||.$$ 

Now to estimate this mess ...
Start with this

\[ \kappa = \frac{\| A\delta x \|}{\| Ax \|} \]

use

\[ \frac{1}{\| Ax \|} \geq \frac{1}{\left( \| A^{-1} \| \| x \| \right)} = \| A^{-1} \| \| x \| \]

to get

\[ \kappa \leq \frac{\| A \| \| A^{-1} \| \| \delta x \|}{\| \delta x \| \| x \|} = \| A \| \| A^{-1} \|. \]

This leads to the most profound definition in the course . . .
The condition number of an \( N \times N \) nonsingular matrix \( \mathbf{A} \) relative to the vector norm \( \| \cdot \| \) is

\[
\kappa(\mathbf{A}) = \| \mathbf{A} \| \| \mathbf{A}^{-1} \| \geq 1.
\]

Can you see why \( \kappa(\mathbf{A}) \geq 1 \)?

When the norm matters, we indicate that with \( \kappa \). For example

\[
\kappa_2(\mathbf{A}) = \| \mathbf{A} \|_2 \| \mathbf{A}^{-1} \|_2.
\]

A matrix is well-conditioned if \( \kappa(A) \) is small, ill-conditioned if \( \kappa(A) \) is HUGE.

Once again, ill-defined with lots of middle ground.
(over) Generalized view of $\kappa$

Return to

$$\kappa = \frac{\|S(x + \delta x) - S(x)\|}{\|\delta x\|/\|x\|}.$$ 

and think of

$$\frac{\|S(x + \delta x) - S(x)\|}{\|\delta x\|} \approx \|S_x(x)u\|.$$ 

where $u = \delta x/\|\delta x\|$ is a unit vector in the direction $\delta x$

Then, since $\|u\| = 1$,

$$\kappa \approx \frac{\|S_x(x)\|\|x\|}{\|S(x)\|}.$$ 

You’ll hear more about this when we get to nonlinear equations.
Bottomline on stability and conditioning

- Stability is a property of the algorithm.
  Fix instability with a change in the algorithm.
- Conditioning is a property of the problem.
  Fix ill conditioning with a reformulation of the problem.
O and o Notation

- $a_n = O(b_n)$ as $n \to \infty$ means that there is $K$
  \[ \|a_n\| \leq K\|b_n\| \text{ for all sufficiently large } n. \]

- $a_n = o(b_n)$ as $n \to \infty$ means that
  \[ \lim_{n \to \infty} \frac{\|a_n\|}{\|b_n\|} = 0. \]

- Similarly $f(h) = O(h^p)$ and $f(h) = o(h^p)$ as $h \to 0$. 