Week 7: aim to cover

- Vector & matrix norms; sensitivity (Lecture 13)
- `chol`, `\` (Lab 7)
- error analysis of linear systems (Lecture 14)

Vector and Matrix norms

In order to discuss sensitivity and numerical stability in solving a linear system, need to measure the 'size' of errors in the inputs and outputs.

In the problem of solving $\mathbf{Ax} = \mathbf{b}$,

- the inputs are $\mathbf{A}, \mathbf{b}$: a matrix and a vector
- the output is $\mathbf{x}$: a vector

So have to introduce a way to measure the 'size' of vectors and matrices.

Vector norms

Definition

The **norm** of a vector $\|\mathbf{x}\|$ is a function $\mathbb{R}^n \mapsto \mathbb{R}$ such that

1. $\|x\| \geq 0 \; \forall x \in V$ where $\|x\| = 0 \Leftrightarrow x = 0$ (norms are positive for nonzero vectors)

2. $\|\alpha x\| = |\alpha| \|x\|$ (scaling a vector scales norm by the same amount)
3. $\|x + y\| \leq \|x\| + \|y\| \; \forall x, y \in V$ (triangle inequality)

Example

The 3 most common vector norms are:

1. $\|x\|_1 = \sum |x_i|$ (the 1-norm)
2. $\|x\|_2 = (\sum x_i^2)^{1/2} = \sqrt{x^T x}$ (the 2-norm i.e. the usual Euclidean norm)
3. $\|x\|_\infty = \max_i |x_i|$ (the ∞-norm)

which are all special cases of the p-norm:

$$\|x\|_p = (\sum |x_i|^p)^{1/p}$$

Example

Unit vectors in 1,2 and ∞ norms

Norm equivalence

In finite-dimensional spaces, it doesn't matter much precisely which norm you use, since they can't differ by more than a factor n from each other.

$$\|x\|_2 \le \|x\|_1 \le \sqrt{n}\|x\|_2$$

$$\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\|x\|_\infty$$

$$\|x\|_\infty \le \|x\|_1 \le n\|x\|_\infty$$

So you may as well use whichever one is convenient.

Matrix norms

A matrix norm is just a vector norm on the $m \times n$ dimensional vector space of $m \times n$ matrices.

Definition

The **norm** of a matrix $\|\mathbf{A}\|$ is a function $M_{m\times n} \mapsto \mathbb{R}$ such that

1. $\|A\| \geq 0 \ \forall A \in M_{m\times n}$ where $\|A\| = 0 \Leftrightarrow A = 0$ (norms are positive for nonzero matrices)
2. $\|\alpha A\| = |\alpha| \|A\|$ (scaling a matrix scales norm by the same amount)
3. $\|A + B\| \leq \|A\| + \|B\| \ \forall A, B \in M_{m\times n}$ (triangle inequality)

Example

The following are matrix norms:

1. $\|A\|_F = (\sum A_{ij}^2)^{1/2}$ (the Frobenius norm)
2. $\|A\|_M = \max_{i,j} |A_{ij}|$ (the max-norm)

Mostly we'll use a common class of matrix norms, called **subordinate matrix norms**.

Subordinate matrix norms

Definition

The **subordinate norm** of a (square) matrix **A** is given by

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

for any vector norm. This is equivalent to:

$$\|A\| = \max_{\|x\|=1} \|Ax\|$$

In words, the (subordinate) norm of a matrix is the norm of the largest image under the map **A** of a unit vector.

Example

The **subordinate p-norms** correspond to the vector norms listed above:

1. $\|A\|_1 = \max_j \sum_i |A_{ij}|$ (the maximum column sum)
2. $\|A\|_2$ (the 2-norm)
3. $\|A\|_\infty = \max_i \sum_j |A_{ij}|$ (the maximum row sum)

These are the only subordinate p-norms that are easy to compute.
The subordinate norms have some useful **submultiplicative** properties:

$$\|Ax\|_p \le \|A\|_p \|x\|_p$$

by definition; and

$$\|AB\|_p \le \|A\|_p \|B\|_p$$

Any norm with the latter property is called **consistent**.
Again, all matrix norms are within a factor of n of each other, so you may as well use whichever one is convenient.

Error analysis in numerical linear algebra

In analyzing the errors made in Gauss elimination, Wilkinson (1960) realized there were different sources of error in a computation:

- the sensitivity of the problem (whatever algorithm you use)
- the quality of your algorithm to solve that problem

These ideas are now used in many other areas of numerical analysis.

Forward vs. backward error

Given a computation $Y = f(X)$, in general we will produce an approximation $\hat{Y}$. The **forward error** is $\Delta Y = \hat{Y} - Y$. We need a small forward error for an accurate answer BUT it can be hard to estimate forward error.

Instead ask a different question:
Is the computed answer the exact answer to a nearby problem?
Is $\hat{Y} = f(X + \Delta X)$ for some small **backward error** ΔX?
Did we solve a problem close to the one we wanted to solve?
It turns out to be often easier to bound the backward error.

Diagram

Example

Backward error

If the backward error is not too big (compared to the relevant data error) we say the algorithm is **backward stable** or just **stable**.

Example

For solving linear systems, the input errors are roundoff errors so we want the **relative backward error** not too big compared to unit roundoff u.

Once we have introduced the concept of backward error, the error made by the algorithm can be treated as if it was error in the data. Then the forward error only depends on the sensitivity of the problem f and the size of the input error i.e. the backward error.

Sensitivity = conditioning

A problem which is sensitive to small errors is called **ill-conditioned** or **ill-posed** $\implies$ no numerical method will get a very accurate answer.
Typically sensitivity is determined using perturbation analysis , assuming small errors.

We quantify sensitivity by defining a **condition number** to measure how sensitive the answer is to errors in the input.
Because roundoff errors are relative errors, we use the **relative condition number** $\sim$ relative error of output/relative error of input i.e. the magnification factor of relative error

Rule of thumb

$\boxed{\textbf{Forward error} \lesssim \textbf{condition number} \times \textbf{backward error}}$

- a stable method on a well-conditioned problem $\longrightarrow$ accurate answer
  this is what we want!
- a stable method on an ill-conditioned problem $\longrightarrow$ inaccurate answer
  re-examine the formulation of your problem!
- an unstable method on a well-conditioned problem $\longrightarrow$ inaccurate answer
  This is what we must avoid!

Trefethen's Maxims

If the answer is highly sensitive to perturbations, you have probably asked the wrong question.

All that it is reasonable to ask for in a scientific calculation is stability, not accuracy.

Sensitivity of a linear system

Suppose $\mathbf{A}, \mathbf{b}$ are perturbed by $\mathbf{\Delta A}, \mathbf{\Delta b}$ — how much is the solution $\mathbf{x}$ changed?

To measure the input and output errors, we use vector and matrix norms.

Theorem

$$\frac{\|\mathbf{\Delta x}\|}{\|\mathbf{x}\|} \le \frac{\kappa(A)}{1-\kappa(A)\frac{\|\mathbf{\Delta A}\|}{\|\mathbf{A}\|}}\left(\frac{\|\mathbf{\Delta A}\|}{\|\mathbf{A}\|}+\frac{\|\mathbf{\Delta b}\|}{\|\mathbf{b}\|}\right)$$

provided $\kappa(A)\frac{\|\mathbf{\Delta A}\|}{\|\mathbf{A}\|} < 1$, *a condition which ensures that the matrix* $\mathbf{A} + \mathbf{\Delta A}$ *is nonsingular.*

$\kappa(A)$ is the (normwise relative) condition number for solving a linear system

The condition number

Definition

The (normwise) **condition number** of a square nonsingular matrix is

$$\kappa(A) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

For small enough $\|\mathbf{\Delta A}\|$, we will have $\kappa(A)\frac{\|\mathbf{\Delta A}\|}{\|\mathbf{A}\|} \ll 1$ so that

$$\boxed{\frac{\|\mathbf{\Delta x}\|}{\|\mathbf{x}\|} \lesssim \kappa(A)\left(\frac{\|\mathbf{\Delta A}\|}{\|\mathbf{A}\|} + \frac{\|\mathbf{\Delta b}\|}{\|\mathbf{b}\|}\right)}$$

relative forward error $\lesssim$ (condition number) (relative error in $\mathbf{A}, \mathbf{b}$)

Note: it's not cheap to compute $\kappa(A)$ since it takes $\approx n^3$ operations to find $\mathbf{A}^{-1}$. Most codes instead try to **estimate** $\kappa(A)$.

Some properties of the condition number

The precise value depends on what norm you're using but they will all be quite similar in size (to within factors of n)

1. $\kappa(\mathrm{I}) = \| \mathrm{I} \| \| \mathrm{I}^{-1} \| = \| \mathrm{I} \|^2 = 1$ in any subordinate norm
2. $\kappa(\mathbf{A}) = \| \mathbf{A} \| \| \mathbf{A}^{-1} \| \geq \| \mathbf{A}\mathbf{A}^{-1} \| = \| \mathrm{I} \| = 1$ so $\kappa(\mathbf{A}) \geq 1$

Matrices with $\kappa(A) \gg 1$ are **ill-conditioned** $\rightarrow$ the solution is very sensitive errors in $\mathbf{A}$ or $\mathbf{b}$ (in the worst case)

Heuristic

If $\kappa(A) \sim 10^k$ then in solving $\mathbf{Ax} = \mathbf{b}$ in t-digit arithmetic, you will lose k decimal digits of precision $\implies$ $\mathbf{x}$ has $t - k$ digits of precision
$\implies$ if $\kappa(A) > 10^t$ it's not worth solving the system since $\mathbf{x}$ may have no significant figures!

Example

The Hilbert matrix