

15. Problem condition and numerical stability

- condition of a mathematical problem
- condition of a set of linear equations
- condition number of a matrix
- cancellation
- numerical stability

Sources of error in numerical computation

example: evaluate a function $f : \mathbf{R} \rightarrow \mathbf{R}$ at a given x

sources of error in the result:

- x is not exactly known
 - measurement errors
 - errors in previous computations
 - how sensitive is $f(x)$ to errors in x ?
- the algorithm for computing $f(x)$ is not exact
 - discretization (*e.g.*, the algorithm uses a table to look up $f(x)$)
 - truncation (*e.g.*, f is computed by truncating a Taylor series)
 - rounding error during the computation
 - how large is the error introduced by the algorithm?

The condition of a problem

sensitivity of the solution with respect to errors in the data

- a problem is *well-conditioned* if small errors in the data produce small errors in the solution
- a problem is *ill-conditioned* if small errors in the data may produce large errors in the solution

rigorous definition depends on what 'large error' means (absolute or relative error, which norm is used, . . .)

example: function evaluation

$$y = f(x), \quad y + \Delta y = f(x + \Delta x)$$

- absolute error

$$|\Delta y| \approx |f'(x)| |\Delta x|$$

ill-conditioned with respect to absolute error if $|f'(x)|$ is very large

- relative error

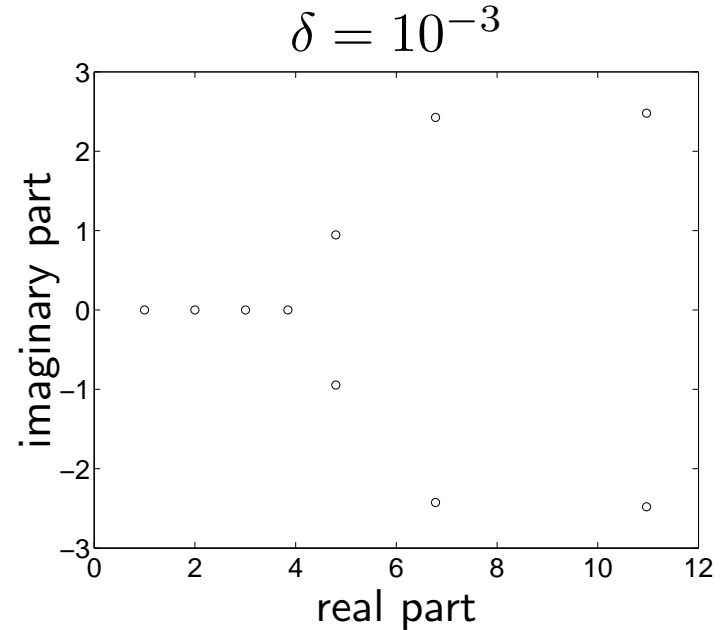
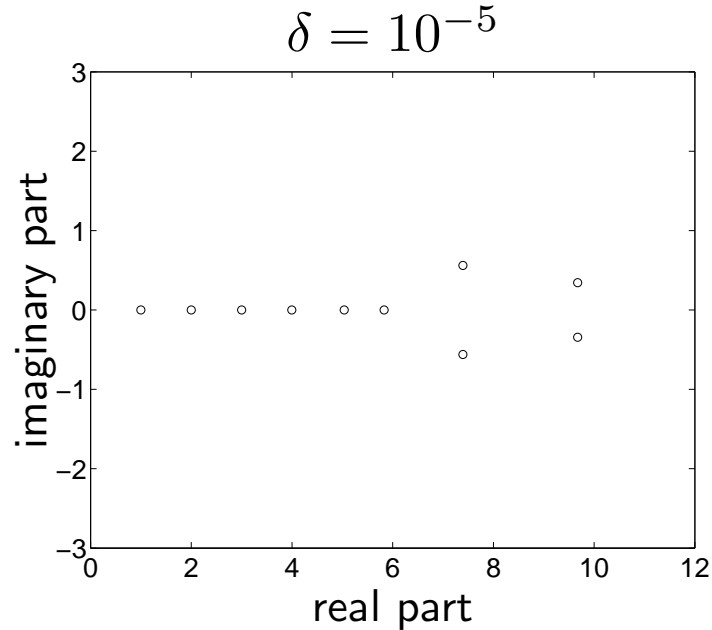
$$\frac{|\Delta y|}{|y|} \approx \frac{|f'(x)| |x| |\Delta x|}{|f(x)| |x|}$$

ill-conditioned w.r.t relative error if $|f'(x)| |x| / |f(x)|$ is very large

Roots of a polynomial

$$p(x) = (x - 1)(x - 2) \cdots (x - 10) + \delta \cdot x^{10}$$

roots of p computed by MATLAB for two values of δ



roots are very sensitive to errors in the coefficients

Condition of a set of linear equations

assume A is nonsingular and $Ax = b$

if we change b to $b + \Delta b$, the new solution is $x + \Delta x$ with

$$A(x + \Delta x) = b + \Delta b$$

the change in x is

$$\Delta x = A^{-1} \Delta b$$

'condition' of the solution

- the equations are *well-conditioned* if small Δb results in small Δx
- the equations are *ill-conditioned* if small Δb can result in large Δx

Example of ill-conditioned equations

$$A = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 + 10^{-10} & 1 - 10^{-10} \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} 1 - 10^{10} & 10^{10} \\ 1 + 10^{10} & -10^{10} \end{bmatrix}$$

- solution for $b = (1, 1)$ is $x = (1, 1)$
- change in x if we change b to $b + \Delta b$:

$$\Delta x = A^{-1} \Delta b = \begin{bmatrix} \Delta b_1 - 10^{10}(\Delta b_1 - \Delta b_2) \\ \Delta b_1 + 10^{10}(\Delta b_1 - \Delta b_2) \end{bmatrix}$$

small Δb can lead to extremely large Δx

Bound on absolute error

suppose A is nonsingular and $\Delta x = A^{-1}\Delta b$

upper bound on $\|\Delta x\|$

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

(follows from property 4 on page 2-22)

- small $\|A^{-1}\|$ means that $\|\Delta x\|$ is small when $\|\Delta b\|$ is small
- large $\|A^{-1}\|$ means that $\|\Delta x\|$ can be large, even when $\|\Delta b\|$ is small
- for every A , there exists Δb such that $\|\Delta x\| = \|A^{-1}\| \|\Delta b\|$ (no proof)

Bound on relative error

suppose A is nonsingular, $Ax = b$ with $b \neq 0$, and $\Delta x = A^{-1}\Delta b$

upper bound on $\|\Delta x\|/\|x\|$:

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\| \frac{\|\Delta b\|}{\|b\|} \quad (1)$$

(follows from $\|\Delta x\| \leq \|A^{-1}\|\|\Delta b\|$ and $\|b\| \leq \|A\|\|x\|$)

- $\|A\|\|A^{-1}\|$ small means $\|\Delta x\|/\|x\|$ is small when $\|\Delta b\|/\|b\|$ is small
- $\|A\|\|A^{-1}\|$ large means $\|\Delta x\|/\|x\|$ can be much larger than $\|\Delta b\|/\|b\|$
- for every A , there exist $b, \Delta b$ such that equality holds in (1) (no proof)

Condition number

definition: the condition number of a nonsingular matrix A is

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

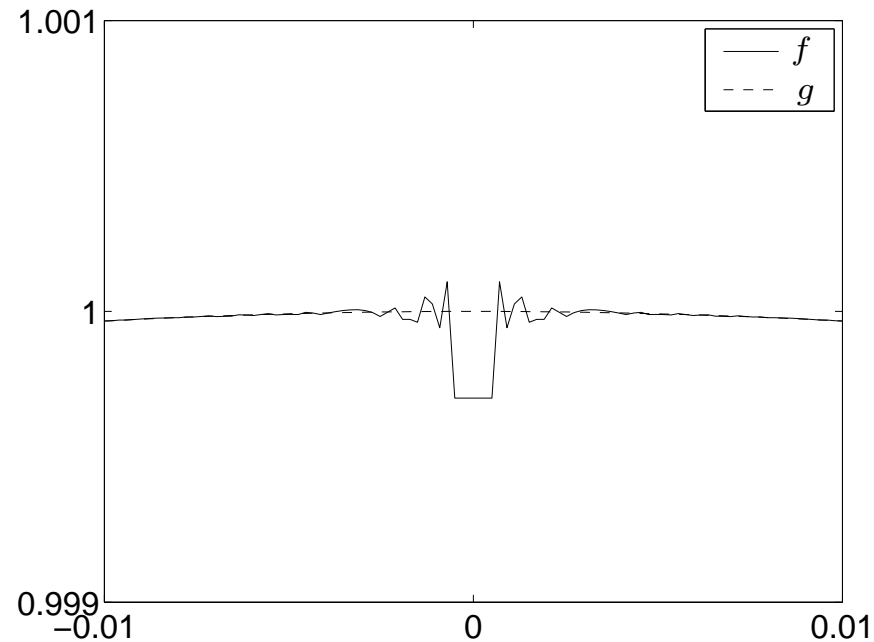
properties

- $\kappa(A) \geq 1$ for all A (last property on page page 2-22)
- A is a *well-conditioned* matrix if $\kappa(A)$ is small (close to 1):
the relative error in x is not much larger than the relative error in b
- A is *badly conditioned* or *ill-conditioned* if $\kappa(A)$ is large:
the relative error in x can be much larger than the relative error in b

Effect of rounding error on algorithms

example: two expressions for the same function:

$$f(x) = \frac{1 - \cos^2 x}{x^2}, \quad g(x) = \frac{\sin^2 x}{x^2}$$



results of $\cos x$ and $\sin x$ were rounded to 10 decimal digits; other calculations are exact

for example, $x = 5 \cdot 10^{-5}$

$$\text{fl}(\cos x) = 0.9999999988 \quad \text{fl}(\sin x) = 0.4999999998 \cdot 10^{-5}$$

- evaluate $f(x)$:

$$\frac{1 - (\text{fl}(\cos x))^2}{x^2} = 0.9599 \dots$$

has only one correct significant digit

- evaluate $g(x)$:

$$\frac{(\text{fl}(\sin x))^2}{x^2} = 0.9999 \dots$$

has about ten correct significant digits

conclusion: f and g are equivalent mathematically, but not numerically

Cancellation

$$\hat{a} = a(1 + \Delta a), \quad \hat{b} = b(1 + \Delta b)$$

- a, b : exact data; \hat{a}, \hat{b} : approximations; $\Delta a, \Delta b$: unknown relative errors
- relative error in $\hat{x} = \hat{a} - \hat{b} = (a - b) + (a\Delta a - b\Delta b)$ is

$$\frac{|\hat{x} - x|}{|x|} = \frac{|a\Delta a - b\Delta b|}{|a - b|}$$

if $a \simeq b$, small Δa and Δb can lead to very large relative errors in x

this is called **cancellation**; cancellation occurs when:

- we subtract two numbers that are almost equal
- one or both are subject to error

example of page 15-11

cancellation occurs when we evaluate the numerator of

$$f(x) = \frac{1 - \cos^2 x}{x^2}$$

- for small x , $1 \simeq \cos^2 x$
- there is a rounding error in $\cos^2 x$

Numerical stability

refers to the accuracy of an algorithm in the presence of rounding errors

- an algorithm is *unstable* if rounding errors cause large errors in the result
- rigorous definition depends on what 'accurate' and 'large error' mean
- instability is often (but not always) caused by cancellation

examples from earlier lectures

- solving linear equations by LU factorization without pivoting can be unstable (page 7-17)
- Cholesky factorization method for least-squares is less stable than QR factorization method (page 9-17)

Roots of a quadratic equation

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

algorithm 1: use the formulas

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

unstable if $b^2 \gg |4ac|$

- if $b^2 \gg |4ac|$ and $b \leq 0$, cancellation occurs in x_2 ($-b \simeq \sqrt{b^2 - 4ac}$)
- if $b^2 \gg |4ac|$ and $b \geq 0$, cancellation occurs in x_1 ($b \simeq \sqrt{b^2 - 4ac}$)
- in both cases b may be exact, but the squareroot introduces small errors

algorithm 2

- if $b \leq 0$, calculate

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{c}{ax_1}$$

- if $b > 0$, calculate

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_1 = \frac{c}{ax_2}$$

no cancellation

Exercises

suppose `chop(x,n)` rounds x to n decimal digits
(for example `chop(pi,4)` returns 3.142000000000000)

1. cancellation occurs in $(1 - \cos x) / \sin x$ for $x \approx 0$

```
>> x = 1e-2;  
>> (1-chop(cos(x,4)))/chop(sin(x,4))
```

```
ans =
```

```
0
```

(exact value is about 0.005)

give a stable alternative method

2. evaluate

$$\sum_{k=1}^{3000} k^{-2} = 1.6446$$

rounding all intermediate results to 4 digits

```
>> sum = 0;
>> for k=1:3000
    sum = chop(sum+1/k^2, 4);
end
>> sum
```

```
sum =
```

```
1.6240
```

- result has only two correct digits
- not caused by cancellation (there are no subtractions)

explain and propose a better method

3. the number $e = 2.7182818\dots$ can be defined as

$$e = \lim_{n \rightarrow \infty} (1 + 1/n)^n$$

this suggests an algorithm for calculating e : choose n large and evaluate

$$\hat{e} = (1 + 1/n)^n$$

results:

| n | \hat{e} | # correct digits |
|-----------|-------------|------------------|
| 10^4 | 2.718145926 | 4 |
| 10^8 | 2.718281798 | 7 |
| 10^{12} | 2.718523496 | 4 |
| 10^{16} | 1.000000000 | 0 |

explain