NUMERICAL PROGRAMMING 1 CSE

Prof. Oliver Junge
Dr. Daniel Karrasch
Center for Mathematics – M3
TUM

„numerical programming":

real world problem — weather prediction, crash test, ...

$\downarrow$

mathematical problem — equations, e.g. $f(x) = 0$

$\downarrow$

numerical algorithm — Newton's method

$\downarrow$

computer program — newton.m

1. Basics

1.1 Floating point arithmetic

$\pi = 3.141592\ldots$ (base 10)

For each real number, we are only storing finitely many digits:

$$\frac{1}{10} = 0.1_{[10]} = 0.000110011 00\ldots_{[2]}$$

$1 \quad \frac{1}{10} \qquad 1 \quad \frac{1}{2}\ \frac{1}{4}\ \frac{1}{8} \cdots \qquad 2^{-n}$

$$= 1.0 \cdot 10^{-1} = 1.1001100\ldots \cdot 2^{-4}$$

floating point numbers

$$x = \underbrace{\pm}_{\text{sign}} \underbrace{d_1 . d_2 d_3 \cdots d_p}_{\text{mantissa } f} \cdot \underbrace{b}_{\text{base}}^{e \text{ exponent}}$$

with digits $d_k \in \{0, 1, \ldots, b-1\}$

normalization: „float" the decimal point such that $d_1 \neq 0$ (and adjust the exponent appropriately)

Since we can only store a bounded range of integers, we need

$$e_{min} \leq e \leq e_{max}$$

For numbers with $e > e_{max}$ we get an overflow error / Inf

$e < e_{min}$ we get an underflow error / 0

IEEE 754 floating point numbers

double (precision): 64 bits = 8 bytes

s: 1	e: 11	f: 52

$$x = \begin{cases} (-1)^s\, 1.d_2\cdots d_p \cdot e^{e-1023} & \text{if } e \in 1:2046 \\ (-1)^s\, 0.d_2\cdots d_p \cdot e^{-1023} & \text{if } e = 0 \\ \text{Inf} & \text{if } e = 2047 \text{ and } f = 0 \\ \text{NaN} \quad \text{„not a number"} & \text{if } e = 2047 \text{ and } f \neq 0 \end{cases}$$

Machine epsilon

is the smallest number ε such that (in floating point arithmetic)

$$1 + \varepsilon \neq 1$$

$$\begin{array}{rlcl} & 1.00 \cdots 0 & \cdot\, 2^0 & = 1 \\ + & 0.00 \cdots 1 & \cdot\, 2^0 & = \varepsilon \\ \hline = & 1.00 \cdots 1 & \cdot\, 2^0 & \neq 1 \end{array}$$

double precision: $\varepsilon \approx 2 \cdot 10^{-16}$

Rounding

$a \in \mathbb{R}$, $a \neq 0$, then the floating point number $\hat{a}$ which represents a satisfies

$$|a - \hat{a}| \leq \frac{\varepsilon}{2} |a|$$

$$\Leftrightarrow \quad \frac{|a-\hat{a}|}{|a|} \leq \frac{\varepsilon}{2}$$

relative error of $\hat{a}$

IEEE 754 arithmetic

satisfies the following:

for any arithmetic operation $\circ \in \{+, -, \cdot, /\}$ and any floating point numbers a, b:

$$\boxed{\frac{|a \circ b - a \mathbin{\hat{\circ}} b|}{|a \circ b|} \leq \frac{\varepsilon}{2}.}$$

where $a \mathbin{\hat{\circ}} b = \widehat{a \circ b}$

1.2. Conditioning of problems

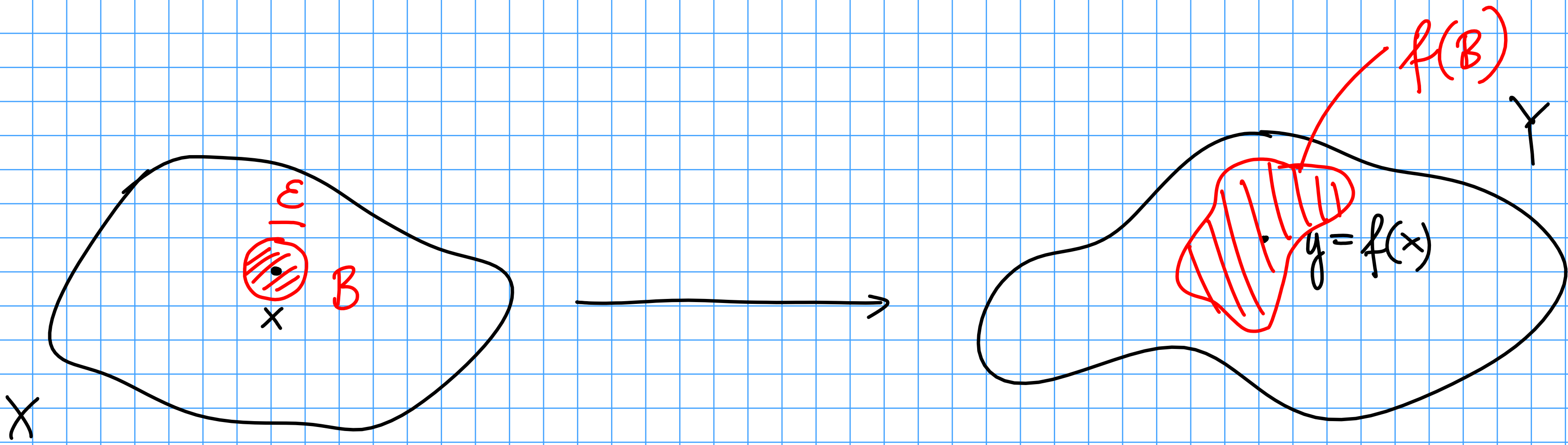
problem (in a mathematical sense): $f : X \to Y$
input output

example: find the zeros of some quadratic polynomial

$$x^2 - 2px - q$$

input : $(p,q) \in \mathbb{R}^2 = X$

output : $(x_1, x_2) \in \mathbb{R}^2 = Y$ s.t. $x_{1,2}^2 - 2p x_{1,2} - q = 0$



condition number of f (at x) is the smallest number $K_f > 0$ s.t.

$$[\![f(x) - f(\hat{x})]\!] \leq K_f [\![x - \hat{x}]\!] \qquad (*)$$

where $[\![\cdot]\!]$ measures the error, typically this the relative error

$$[\![x - \hat{x}]\!] = \frac{|x - \hat{x}|}{|x|}$$

example: $f : (x, y) \in \mathbb{R}^2 \longmapsto x + y$

let's compute the condition of f:

$$\frac{|(x+y) - (\hat{x} + \hat{y})|}{|x+y|} = \frac{|(x - \hat{x}) + (y - \hat{y})|}{|x+y|}$$

$x = \hat{x} + \delta x$
$y = \hat{y} + \delta y$

$$= \frac{|\delta x + \delta y|}{|x+y|}$$

$$\leq \frac{|\delta x| + |\delta y|}{|x+y|}$$

now use that $\frac{|x-\hat{x}|}{|x|} = \frac{|\delta x|}{|x|} \leq \varepsilon$, i.e. $|\delta x| \leq \varepsilon |x|$

likewise $|\delta y| \leq \varepsilon |y|$

ie we obtain $\frac{|\delta x| + |\delta y|}{|x+y|} \leq \frac{\varepsilon |x| + \varepsilon |y|}{|x+y|}$

$$= \boxed{\underbrace{\frac{|x|+|y|}{|x+y|}}_{= \kappa_f}} \varepsilon = \kappa_f [\![x - \hat{x}]\!]$$

Case 1: x and y have the same sign : $|x| + |y| = |x+y|$

$\Longrightarrow \kappa_f = 1$

Case 2: x and y have opposite sign : $|x+y| = ||x| - |y||$

So for $|x| \approx |y|$, $|x+y| \approx 0$ and κ_f will be large !

Remark: In the output, one typically looses $\log_{10} \kappa_f$ decimal places of accuracy, i.e. $\kappa_f = 10^{10} \Rightarrow$ we loose 10 digit of accuracy

example: loss of accuracy in addition:

$$\begin{aligned} \longrightarrow \quad x &= 1.23467 \cdot 10^0 \\ y &= -1.23456 \cdot 10^0 \\ x+y &= 0.00011 \cdot 10^0 \\ &= 1.1???? \cdot 10^{-4} \end{aligned}$$

cancellation of leading digits

$$\kappa_f = \frac{|x|+|y|}{|x+y|} \approx \frac{2}{10^{-4}} \approx 10^4$$

other arithmetic operation:

op	$\cdot$	/	$\sqrt{\ }$
κ	2	2	1/2

More generally: $f: \mathbb{R}^n \to \mathbb{R}$ differentiable, $x = (x_1, \dots, x_n)^T \mapsto y$

$$\boxed{\kappa_f = \frac{\langle |\nabla f(x)|, |x| \rangle}{|f(x)|}} = \frac{\sum_{i=1}^{n} |\partial_i f(x)| |x_i|}{|f(x)|}$$

(derive by linearization: $f(x+\delta x) - f(x) = \langle \nabla f(x), \delta x \rangle$)

1.3 Stability of algorithms

problem: $x \mapsto y = f(x)$

expected error in y: $[\![y-\hat{y}]\!] \leq \kappa_f \, [\![x-\hat{x}]\!]$

algorithm: $f_n \circ \cdots \circ f_2 \circ f_1$, $f_1, f_2, \dots \in \{+, -, \cdot, /, \sqrt{\ }\}$

goal: $= f$

this sequence of elementary arithmetic operations for evaluating f is not unique!

implementation: $\hat{f} = \hat{f}_n \circ \cdots \circ \hat{f}_2 \circ \hat{f}_1 = fl \circ f_n \cdots \circ fl \circ f_2 \circ fl \circ f_1$

rounding from x to $\hat{x}$

simple case: $f = f_2 \circ f_1$

$\hat{f} = f_2 \circ fl \circ f_1$

if f_2 has a large condition number, then the error in the output will be large stable algorithm: $[\![f(x) - \hat{f}(x)]\!] = \mathcal{O}(\kappa_f \cdot \varepsilon)$

$$= C \kappa_f \varepsilon$$

C constant „not too large"

unstable algorithm: $[\![f(x) - \hat{f}(x)]\!] \gg \kappa_f \varepsilon$

example: problem: compute the zeros of a quadratic polynomial

$$x^2 - 2px - q$$

input: $(p, q) \in \mathbb{R}^2$

output: $(x_1, x_2) \in \mathbb{R}^2$

problem is well conditioned

school algorithm: $x_1 = p - \sqrt{p^2 + q}$

$$x_2 = p + \sqrt{p^2 + q}$$

observation: $\quad (p,q) \xmapsto{f_1} \begin{pmatrix} r = \sqrt{p^2+q} \\ p \end{pmatrix} \xmapsto{f_2} p - r = x_1$ ✗

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad \xmapsto{f_2} p + r = x_2$ ✓

κ_{f_2} is large (in our case), i.e. this algorithm is unstable

better algorithm: $\quad x^2 - 2px - q = (x - x_1)(x - x_2) = x^2 - (x_1 + x_2)x + x_1 x_2$

i.e. $-q = x_1 x_2 \Rightarrow x_1 = -\dfrac{q}{x_2}$

$$(p,q) \xmapsto{f_1} \begin{pmatrix} r = \sqrt{p^2+q} \\ q \\ p \end{pmatrix} \xmapsto{f_2} \begin{pmatrix} x_2 = p + r \\ q \end{pmatrix} \xmapsto{f_3} \begin{pmatrix} x_2 \\ x_1 = -\dfrac{q}{x_2} \end{pmatrix}$$