Numerical analysis (1/7): Errors in numerical analysis

University of Luxembourg

Philippe Marchner

Siemens Digital Industries Software, France

October 19th, 2023

Outline

1. Introduction: types of errors
2. Computer representation of real numbers
3. Floating-point arithmetic
4. Conditioning and stability
5. Summary

Outline

Types of errors

What type of errors do we find in numerical analysis ?

1. Errors in the problem to be solved
   - errors in the mathematical model
   - errors in the input data: noise, measurement
2. Round-off errors (today's topic, computer floating-point representation)
3. Approximation errors
   - Discretization/truncation errors: going from the continuous to the discrete level
   - Convergence errors: iterative methods

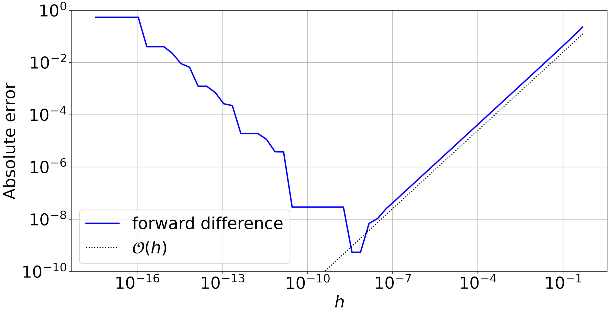
Round-off vs discretization error: example

Derivative approximation (see exercise)

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + \mathcal{O}(h)$$

The **discretization error** is linear with h

But when h is too small, we have round-off errors



Outline

1. Introduction: types of errors
2. Computer representation of real numbers
3. Floating-point arithmetic
4. Conditioning and stability
5. Summary

Floating-point representation

A computer has a finite capacity and cannot store all real numbers ($\pi \approx 3.14159...$)
Any number is approximated by a rational number, and has a finite number of digits d_i

Floating point representation of $x \in \mathbb{R}$

$$\mathrm{fl}(x) = \pm m \times b^e, \quad m = \left(\frac{d_0}{b^0} + \frac{d_1}{b^1} + \cdots + \frac{d_{p-1}}{b^{p-1}}\right)$$

m: mantissa or significand, e: exponent, b: radix or basis, p: precision

Normalized (unique) representation: $1 \leq m < b$, $d_0 \neq 0$

Examples

Let us choose $\mathrm{fl}(x) = +152853.50, \quad b = 10, p = 8, e = 5$

$$152853.50 = 1.5285350 \times 10^5 = \frac{15285350}{10^7} \times 10^5 = (1 \times 10^0 + 5 \times 10^{-1} + \cdots + 0 \times 10^{-7}) \times 10^5$$

Floating-point representation

Examples

Approximate π in base 2, $p = 24$: $\pi \approx 11001001\,00001111\,1101101\mathbf{1}$
Approximate 0.1 in base 2, $p = 16$: $0.1 \approx 11001100\,1100110\mathbf{1}$

$\text{fl}(\pi) = \left(1 + 1 \times 2^{-1} + \cdots + 1 \times 2^{-23}\right) \times 2^{1} = 3.141592\mathbf{7}$, 7 digits precision (why ?)

Floating point system

A floating point system is characterized by 4 values: (b, p, U, L)
base, precision, exponent upper and lower bound such as $L \leq e \leq U$

Double precision format (64 bit storage)

$b = 2, p = 52, L = -1022, U = 1023$
11 bit exponent, 52+1 bits for the significand (1 is implicit for the sign)

Let us look what Python uses...

Floating-point representation

Example

Single precision: $p = 23$ bits significand, 8 bit exponent, $U = 127$

The sequence of digits

$$1\ 0111\ 1110\quad 100\ 0000\ 0000\ 0000\ 0000\ 0000$$

considered as simple precision can be interpreted as follows:

- the first digit is 1, so the sign is negative
- the next 8 digits form the exponent: $(0111110)_2 = 0 \times 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 0 \times 2^0 = 126$
- the next 23 digits form the significand, where the first bit is 1 by convention

The number in decimal is:

$$-2^{(126-U)} \times (1 + 1 \times 2^{-1} + 0 \times 2^{-2} + \cdots + 0 \times 2^{-23}) = -0.5 \times 1.5 = -0.75$$

Machine precision

When rounding, two adjacent numbers are spaced by $\eta = b^{-p}/2$, it defines the *machine precision*.

Round-off error

Absolute *round-off* error: $|\mathrm{fl}(x) - x| \leq \eta b^e$

Relative *round-off* error: $\left|\frac{\mathrm{fl}(x)-x}{x}\right| \leq \eta$

The spacing of a floating point system is constant in the **relative sense**

Overflow-underflow

If the exponent e is such as $e > U$, we have overflow, and if $e < L$ we have underflow

Question: what are the minimum and maximum representable numbers in double precision ?

Outline

Operations in floating point arithmetic

The usual calculus rules are altered with the floating point representation

Floating operation

$$x \bullet y = (x \bullet y)(1+\varepsilon), \quad \bullet = (+,-,\times,\div), \quad |\varepsilon| \leq \eta$$

Addition is not associative in this context ! $(a+b)+c \neq a+(b+c)$
The relative errors are $\frac{a+b}{a+b+c}\varepsilon_1(1+\varepsilon_2)+\varepsilon_2$ and $\frac{b+c}{a+b+c}\varepsilon_3(1+\varepsilon_4)+\varepsilon_4$

Examples

Derive the relative errors of the above additions
For $p=8$ try $a = 2.3371258 \times 10^{-5}$, $b = 3.3678429 \times 10^{1}$, $c = -3.3677811 \times 10^{1}$

Examples

Compute $Q = \frac{\pi - 3.1415}{10^4(\pi - 3.1515) - 0.927}$ by truncating π with p significant digits

Outline

Stability

Stability is an important concept in numerical analysis, and is found in different contexts:

- stability of numerical schemes (e.g. ODEs, finite differences),
- stability of numerical algorithms (e.g. linear systems),
- stability has also meanings at the continuous level (PDE, dynamical systems),

It is linked to the propagation of errors during the computation.

Stability

Forward and backward errors

- The forward error measures the difference between the computed and exact value $|\hat{y} - y|$
- The backward error is the error on the input: $|\hat{x} - x|$, for a perturbed output $\hat{y} = f(\hat{x})$

If the backward error is "small", we say the algorithm to be backward-stable

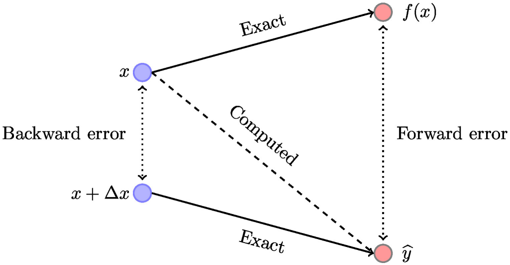


Figure: Illustration from Nicholas J. Higham blog

Conditioning

The condition number is the ratio of a relative change in the output to a relative change in the input

Condition number κ

$$\kappa = \frac{|(f(\hat{x}) - f(x))/f(x)|}{|(\hat{x} - x)/x|} = \frac{|(\hat{y} - y)/y|}{|(\hat{x} - x)/x|} = \frac{|\Delta y/y|}{|\Delta x/x|}$$

We have the relation

$$|\text{relative forward error}| = \text{Condition number } \times |\text{relative backward error}|$$

It represents the sensitivity related to the input data.

Conditioning depends on the problem, stability depends on the algorithm

Condition number of a C^1 function f

when $\hat{x} \approx x$, we have $\kappa_f \approx \left| \frac{x f'(x)}{f(x)} \right|$

Conditioning and stability - examples

Examples

Find the conditioning of the functions $\sqrt{x}$, $a - x$, $\tan(x)$

Examples

Evaluate the stability of $f(x) = \sqrt{x+1} - \sqrt{x}$ by computing $f(12345)$ with $p = 6$ significant digits

Try again using the formula $f(x) = \frac{1}{\sqrt{x+1}+\sqrt{x}}$

Compare the stability properties of the two formulas

Outline

Summary

- All machines use a floating-point representation,
- We must be very careful during computations to avoid round-off errors,
- Conditioning and stability analysis are useful theoretical tools,
- Designing numerically stable algorithms is in general not trivial