Numerical Methods

Elena Celledoni

Department of Mathematical Sciences, NTNU

January 7th, 2019

Practical issues

- information: through Blackboard and webpage: https://wiki.math.ntnu.no/ma2501/2019v/start
- **Lecturer**: Elena Celledoni, elena.celledoni@ntnu.no
- **Lecures**: Mondays (12:15-15 in MA24 Grønnbygget).
- **Next lecture**: Thursday 10th, in MA24, at 08:15-10.
- **Teaching assistant**: Tale Bakke Ulfsby, tale.b.ulfsby@ntnu.no;
- **Exercise classes**: Thursdays 08:15-10 in Nullrommet (central building 2, 3rd floor). First class January, 17th.
- **Project**: count for a total of 30 % of the final mark. Three parts: first part consists of two assignments one due on the 24th of January and one due on the 31st of January.
- **Book**: Süli and Mayers: An Introduction to Numerical Analysis.

Please volunteer to be part of the reference group for this course!

Tasks of the reference group

- Participate to the reference group meetings, voice the needs and defend the interests of the students.
- Write the student report about the course.
- Need two or three students for the reference group.

Presentation

- EC Italian living in Norway:
- BSc and MSc at University of Trieste.
- PhD at University of Padova: Computational Mathematics.
- Postdoc: Cambridge University, UK, Mathematical Sciences Research Institute in Berkeley, USA, and at NTNU.
- Worked in SINTEF Applied Mathematics form 2001 to 2004.
- At the Department of Mathematical Sciences, NTNU since 2004.
- Professional interests: numerical analysis, mathematics, applications.

Python

- For installation of Python see: https://wiki.math.ntnu.no/anaconda/start
- Ask the TA (Tale Bakke Ulfsby, tale.b.ulfsby@ntnu.no) if you need help.

Learning outcome

A student successfully meeting all the learning objectives of this course

1. **will have developed knowledge and general competence in numerical methods**: familiarity with selected algorithms, knowledge of how these algorithms are developed and analyzed; familiarity with central concepts such as error sources, convergence and stability.
2. **will be able to choose a suitable numerical algorithm for a given mathematical problem, to implement this method, and to critically evaluate the result.** Ability of developing (further) simple numerical algorithms and to analyze these.

The **exam** and the compulsory **assignments** are designed to test the achievement of the learning outcome.

Representation of numbers on a computer: Floating point model

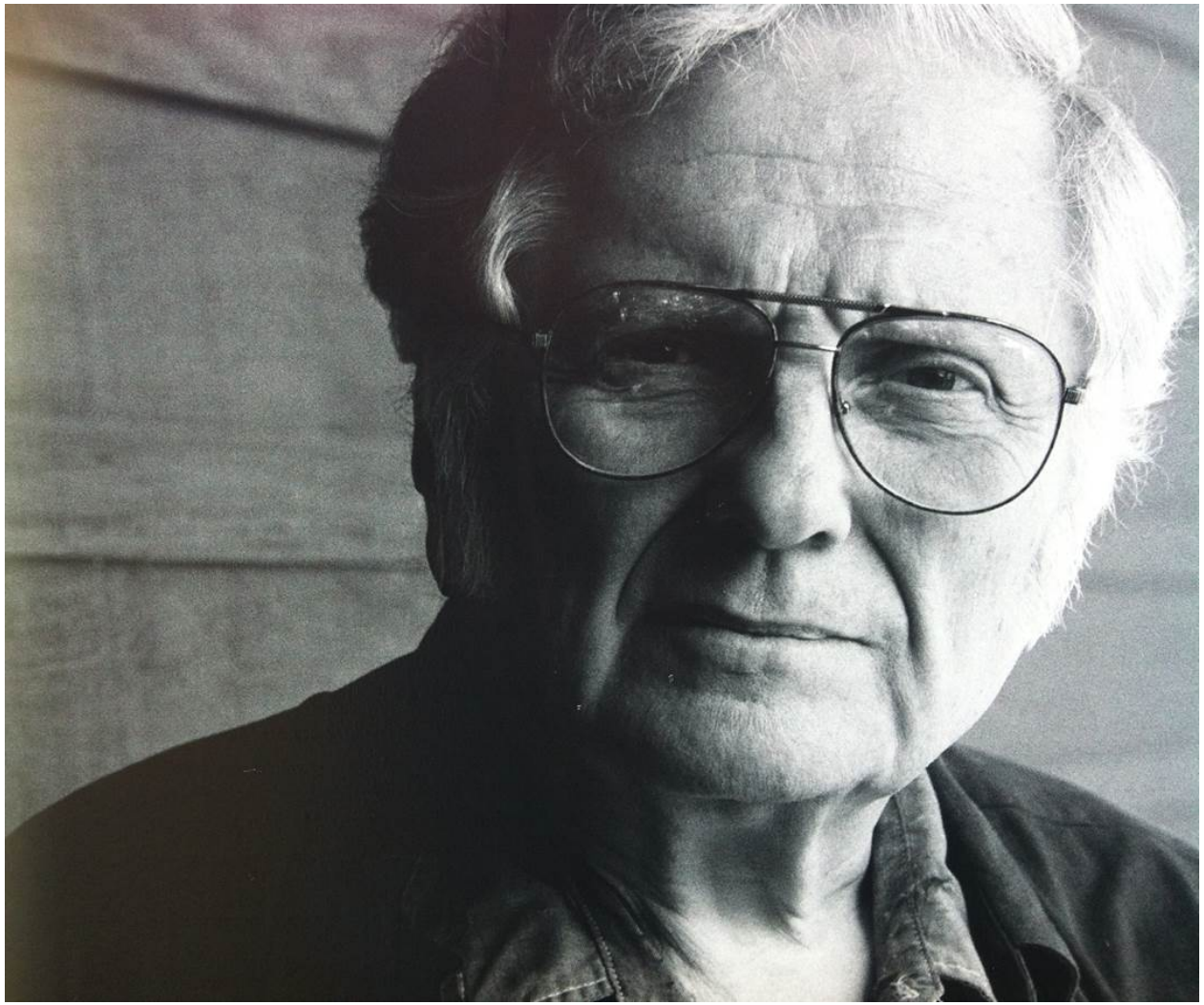
Computers have finite memory hence not every number can be represented exactly on a computer.
Examples: $\sqrt{2}$, π have infinite number of digits.
To fit in a computer, real numbers are approximated via the **floating point model**:

- binary system is used:
  $r = \pm(\alpha_k\, 2^k + \alpha_{k-1}\, 2^{k-1} + \alpha_{k-2}\, 2^{k-2} + \cdots + \alpha_0\, 2^0)$ where $\alpha_0, \ldots, \alpha_k \in \{0, 1\}$, $\alpha_k \neq 0$.
- a **fixed** amount of memory is allocated to represent each number:

$$\begin{aligned} r &= \pm\, 0.\, \alpha_k\, \alpha_{k-1}\, \ldots \alpha_{k-m-1}\, \alpha_{k-m} \ldots\, \alpha_0 \quad \cdot 2^{k+1} \\ fl(r) &= \pm\, 0.\, \alpha_k\, \alpha_{k-1}\, \ldots \alpha_{k-m-1}\, \tilde{\alpha}_{k-m} \quad \cdot 2^{E} \end{aligned}$$

sign significant digits exponent

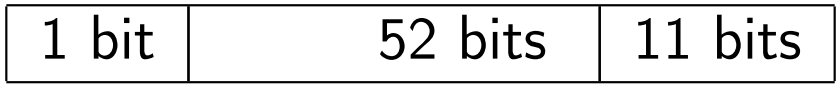
William M. Kahan



Kahan, Turing award in 1989, was the primary architect behind the IEEE 754-1985

Floating point numbers

Double precision IEEE 745



sign significant digits exponent

- **Rounding**: $r \rightarrow fl(r)$. (Alternative: chopping.)
- **Roundoff error** is $r - fl(r)$.
- **Machine epsilon**: ϵ is the smallest floating point number such that

$$1 + \epsilon \neq 1$$

in the computer.

- **Loss of significant digits**: loss of precision due to subtraction of floating point numbers very close to each other.

Loss of significant digits

Given the two real numbers

$$x = 0.3721478693$$

$$y = 0.37202300572$$

their difference is

$$x - y = 0.0001248121$$

We perform **rounding** at 5 digits, this gives

$$fl(x) = 0.37215$$

$$fl(y) = 0.37202$$

now the difference of the two floating point numbers is

$$fl(x) - fl(y) = 0.00013$$

in memory we can store 5 digits for $fl(x) - fl(y)$ but we really know only two of them, the others are lost.

Avoid propagation of roundoff error

Stability: study how the error propagates due to perturbations in the initial data.

- stability of the problem
- stability of the algorithm

Example (**Problem**: find x such that $ax + b = c$ where a, b, c are given numbers and $a \neq 0$.)

Alg 1: 1. divide by a: $x + \frac{b}{a} = \frac{c}{a}$; 2. subtract $\frac{b}{a}$: $x = \frac{c}{a} - \frac{b}{a}$

Alg 2: 1. subtract b: $ax = c - b$; 2. divide by a $x = \frac{c-b}{a}$

Stability of the problem answers the question: What happens to the solution of $ax + b = c$ if $a \to a(1 + \delta_a)$, $b \to b(1 + \delta_b)$, $c \to c(1 + \delta_c)$?

Stability of the algorithm answers the question: What happens to the output of the algorithm if $a \to a(1 + \delta_a)$, $b \to b(1 + \delta_b)$, $c \to c(1 + \delta_c)$?

Stability and condition numbers

A problem is stable when the relative error in the output solution is of the same size of the relative error in the input data.
Given a stable problem only if we choose a stable algorithm to solve it we get errors in the output which are proportional to the errors in the input.

DEF: **Condition numbers** are constants giving the amplification of the error in the output by means of the error in the input.

Stability and condition numbers

A problem is stable when the relative error in the output solution is of the same size of the relative error in the input data.
Given a stable problem only if we choose a stable algorithm to solve it we get errors in the output which are proportional to the errors in the input.

DEF: **Condition numbers** are constants giving the amplification of the error in the output by means of the error in the input.

Example (Stability of the arithmetic operation "+")

Let $x > 0$ and $y > 0$ real. Let $fl(x) = x(1+\delta_x)$, $fl(y) = y(1+\delta_y)$ with $|\delta_x| \leq \epsilon$ and $|\delta_y| \leq \epsilon$. Look at the relative error:

$$\left|\frac{x+y-(fl(x)+fl(y))}{x+y}\right| = \left|\frac{x+y-(x+x\delta_x+y+y\delta_y)}{x+y}\right|$$

$$= \left|-\frac{x}{x+y}\delta_x - \frac{y}{x+y}\delta_y\right| \leq C \cdot \bar{\delta}$$

where $C = \max\{\frac{x}{x+y}, \frac{y}{x+y}\}$ and $\bar{\delta} = 2 \cdot \max\{|\delta_x|, |\delta_y|\} \leq 2\epsilon$

"+" is a stable operation. C is the CONDITION NUMBER.