An Introduction to Numerical Analysis

Elena Celledoni

Department of Mathematical Sciences, NTNU

21st august 2012

Practical issues

- webpage: http://wiki.math.ntnu.no/tma4215/2012h/start
- **Lecturer**: Elena Celledoni, elenac@math.ntnu.no
- **Lecures**: Tuesdays (14-16 in S4) and Thursdays (8-10 in S7).
- **Exercise and project lecturers**:
  Geir Bogfjellmo, bogfjell@stud.ntnu.no;
  Markus Eslitzbichler, m.eslitzbichler@gmail.com
- **Exercise classes**: Mondays 18-19 in EL2 . We want to schedule also another time. When?
- **Project**: count for a total of 30 % of the final mark. Three deadlines (9th September, 12th October, 16th November)
- **Book**: E. Süli and D. Mayers, An introduction to Numerical Analysis, Cambridge University Press (2003).

Summary

- Presentation
- Computational science
- What is numerical analysis good for?
- Floating point numbers and stability

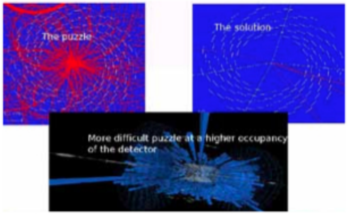
Presentation

- 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.

Computational science

Domains of computational science

- Solution of equations which we do not know how to solve with known analytical tools. Example: **Navier-Stokes equations**.
- Analysis of large amounts of data: image analysis. Examples: data from high energy physics and astrophysics.
- CS can be used to plan experiments.



Pattern recognition ATLAS CERN

We can replace real experiments with *virtual (numerical) experiments* in situations where the real experiments would be to expensive, take too much time, or be unethical.

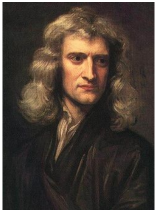
Computational mathematics

Mathematics is the language of science. It is used to describe scientific problems in a *simple* and rigorous way using *mathematical models*, that is equations (linear eq., nonlinear eq., differential equations, integral equations).

To find solutions to these equations our tools are:

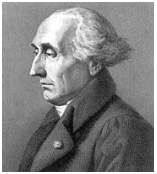
- mathematical methods:
    1. prove that there exists a solution and that it is unique;
    2. find solutions using the tools of analysis;
    3. explore if there are qualitative properties of the solution: invariance under symmetry, conservation properties.
- numerical approximations:
    1. find **good** approximations to the solutions;
    2. find out what happens with the solutions and the approximations when the inputadata is slightly changed (**stability**).

Mathematics, physics and computations.


Isaac Newton (1643-1727)

Leonard Euler(1707-1783)

Carl Friedrich Gauss(1777-1855)


Joseph-Louis Lagrange(1736-1813)

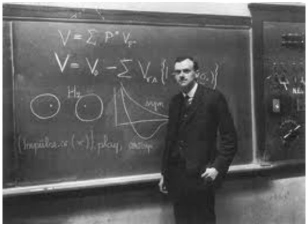
(William R. Hamilton1805-1865)

Carl Gustav Jacobi(1804-1851)

Plan: find solutions of all the differential equations related to mechanical systems.

Not all equations are easily solvable

Paul Dirac understood that it was impossible to solve all differential equations describing mechanical systems.



(1903-1984)

The underlying physical laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known and the difficulty is only that the exact application of these laws leads to equations that are much too complicated to be soluble.

History: modern computational science starts with

John von Neumann (1903-1957)

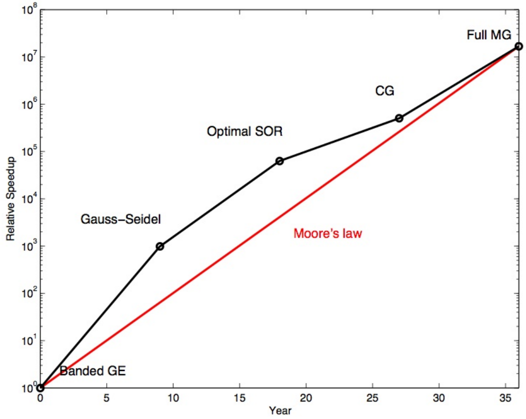
ENIAC (1942-1955)

- Hungarian mathematician, moves to the USA under World War II.
- Contributes to pure and applied mathematics, numerical analysis and computer science.
- Famous for the theory on **von Neuman-architecture** for the design of computers.
- Works with Electronic Numerical Integrator and Computer

Mathematics vs computer power

Computational science has solid mathematical fundations: method development has been at least as important to science as the improved speed of computers in the last 35 years.

Development of computing speed for the solution of the Poisson equation.



red: speed-up due to Moore's law (saying that the speed of computers doubles every 18 months)

black: speed-up due to improved numerical algorithms.

Better computers enable to solve more complex models

As the computers become faster and faster we can solve more advanced and faithful models of the reality (complex systems):

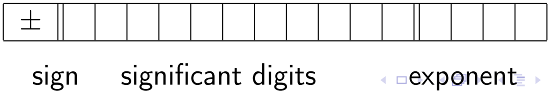
1. important to get a mathematical understanding of these complex systems;
2. important to quantify the uncertainties: comparison of simulation and experimental data (validation).

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 $\alpha_0, \dots, \alpha_k \in \{0,1\}$, $\alpha_k \neq 0$.
- a **fixed** amount of memory is allocated to represent each number:

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



Floating point numbers

Double precision IEEE 745

1 bit	52 bits	11 bits
sign	significant digits	exponent

- **Rounding**: $r \to fl(r)$. (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 of the extent of the propagation of the error with respect to perturbation 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 we can choose an unstable algorithm and experience bad propagation of the error.

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 we can choose an unstable algorithm and experience bad propagation of the error.

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.