CS 450: Numerical Anlaysis[1]

Introduction to Scientific Computing

University of Illinois at Urbana-Champaign

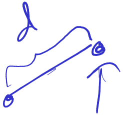
[1] *These slides have been drafted by Edgar Solomonik as lecture templates and supplementary material for the book "Scientific Computing: An Introductory Survey" by Michael T. Heath (slides).*

Scientific Computing Applications and Context

- **Mathematical modelling for computational science** *Typical scientific computing problems are numerical solutions to PDEs*
  - *Newtonian dynamics: simulating particle systems in time*
  - *Fluid and air flow models for engineering*
  - *PDE-constrained numerical optimization: finding optimal configurations (used in engineering of control systems)*
  - *Quantum chemistry (electronic structure calculations): many-electron Schrödinger equation*
- **Linear algebra and computation**
  - *Linear algebra and numerical optimization are building blocks for machine learning methods and data analysis*
  - *Computer architecture, compilers, and parallel computing use numerical algorithms (matrix multiplication, Gaussian elimination) as benchmarks*

Example: Mechanics[2]

- Newton's laws provide incomplete particle-centric picture
- Physical systems can be described in terms of *degrees of freedom* (DoFs)
  - A piston moving up and down requires ____1____ DoFs
  - 1-particle system requires ____3____ DoFs
  - 2-particle system requires ____6____ DoFs
  - 2-particles at a fixed distance require ____5____ DoFs



- N-particle system *configuration* described by $3N$ DoFs

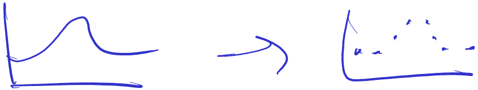
· trajectories in R^{3N}, describe free energy configuration

· basis functions

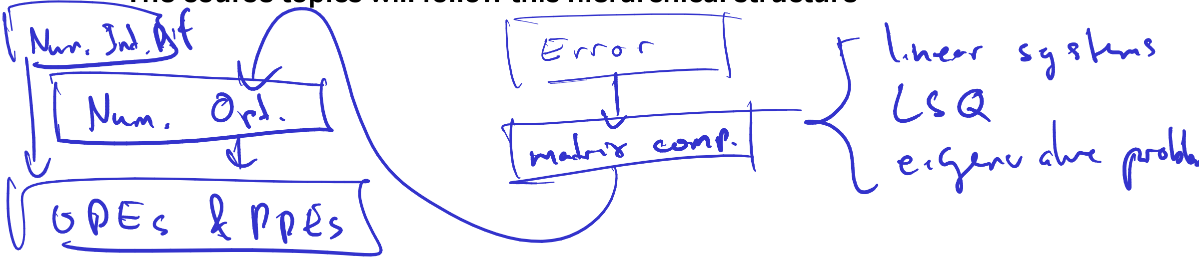
[2] *Variational Principles of Mechanics*, Cornelius Lanczos, Dover Books on Physics, 1949.

Course Structure

- **Complex numerical problems are generally reduced to simpler problems**
  - discretization
  - nonlinear → linear
  - complicated functions → polynomials
  - diff. eqs. → linear equations



- **The course topics will follow this hierarchical structure**



Numerical Analysis

- **Numerical Problems involving Continuous Phenomena:**

$X \in \mathbb{R}^n$, $y = f(x)$

domain

- well-posed if the solution exists and is unique
  it is continuous w.r.t. input
- otherwise ill-posed, e.g., $f'(x) \neq \infty$

- **Error Analysis:**

$y = f(x)$, instead computed $\hat{y} = \hat{f}(x)$

absolute (forward) error

$$\Delta y = \hat{y} - y$$

relative error $\Delta y / |y|$, $\dfrac{\Delta y}{\|y\|}$

Demo: *Floating Point vs Program Logic*

Sources of Error

- **Representation of Numbers:** cannot represent π

  floating point – uniform precision

  · in practice, 32, 64-bit   8-16-bit

  scientific notation, 2.10^{3} mol/ $\times 10^{-23}$

- **Propagated Data Error:** $x \to fl(x)$, $|fl(x) - x| / |x| \le \epsilon$

  $\hat{x} \approx x$

  $f(\hat{x}) - f(x)$

- **Computational Error =** $\hat{f}(x) - f(x)$ **= Truncation Error + Rounding Error**

  · truncation error due to $f \to \hat{f}$ (approximate algorithm) method vs true

  · rounding error $\to fl(x) - x$

Error Analysis

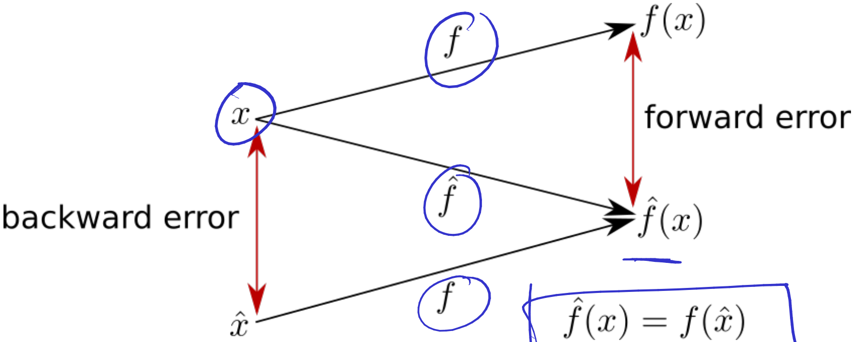
- **Forward Error:**

  $\hat{f}(x) - f(x)$

- **Backward Error:**

  We computed $\hat{f}(x)$, is there some $\hat{x}$ for which $f(\hat{x}) = \hat{f}(x)$, if so $\hat{x} - x$ is our backward error

Visualization of Forward and Backward Error



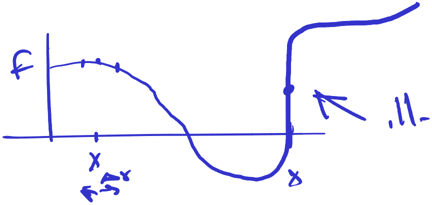
$$\hat{f}(x) = f(\hat{x})$$

Conditioning of a problem

- **Absolute Condition Number:**

$$\kappa_{abs}(f,x) = \lim_{\Delta x \to 0} \frac{|f(x+\Delta x) - f(x)|}{|\Delta x|} = f'(x)$$

$$\kappa_{abs}(f) = \max_{x \in \mathbb{R}} \kappa_{abs}(f,x)$$



ill-conditioned

- **(Relative) Condition Number:**

$$\kappa(f,x) = \lim_{\Delta x \to 0} \frac{|f(x+\Delta x) - f(x)| / |f(x)|}{|\Delta x / |x||}$$

$$\kappa(f) = \max_{x \in D} \kappa(f,x)$$

Posedness and Conditioning

- **What is the condition number of an ill-posed problem?**

Stability and Accuracy

- **Accuracy**:

- **Stability**:

Demo: *Truncation vs Rounding*

Error and Conditioning

- Two major sources of error: *roundoff* and *truncation* error.
  - roundoff error concerns floating point error due to finite precision
  - truncation error concerns error incurred due to algorithmic approximation, e.g. the representation of a function by a finite Taylor series
- To study the propagation of roundoff error in arithmetic we can use the notion of conditioning.

Floating Point Numbers

Demo: *Picking apart a floating point number*
Demo: *Density of Floating Point Numbers*

- **Scientific Notation**

- **Significand (Mantissa) and Exponent** Given x with s leading bits $x_0, \ldots, x_{s-1}$

Rounding Error

***Demo:** Floating point and the Harmonic Series*
***Demo:** Floating Point and the Series for the Exponential Function*

- **Maximum Relative Representation Error (Machine Epsilon)**

Rounding Error in Operations (I)

Demo: *Catastrophic Cancellation*
Activity: *Cancellation in Standard Deviation Computation*

- **Addition and Subtraction**

***Demo:** Polynomial Evaluation Floating Point*

Rounding Error in Operations (II)

- **Multiplication and Division**

Exceptional and Subnormal Numbers

- **Exceptional Numbers**

- **Subnormal (Denormal) Number Range**

- **Gradual Underflow: Avoiding underflow in addition**

Floating Point Number Line