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 _______ DoFs
  - 1-particle system requires _______ DoFs
  - 2-particle system requires _______ DoFs
  - 2-particles at a fixed distance require _______ DoFs
  - $N$-particle system *configuration* described by $3N$ DoFs

---

Course Structure

- Complex numerical problems are generally reduced to simpler problems
  - discretization
  - nonlinear $\rightarrow$ linear
  - complicated functions $\rightarrow$ polynomials
  - diff. eqs. $\rightarrow$ linear equations

- The course topics will follow this hierarchical structure
Numerical Analysis

- Numerical Problems involving Continuous Phenomena:
  \[ X \in \mathbb{R}^n, \quad y = f(x) \]
  - well-posed if the solution exists and is unique
  - otherwise ill-posed, e.g., \( f'(x) \neq 0 \)

- Error Analysis:
  \[ y = f(x), \text{ instead computed } \hat{y} = \hat{f}(x) \]
  - absolute (forward) error
    \[ \Delta y = \hat{y} - y \]
  - relative error
    \[ \frac{\Delta y}{|y|} = \frac{\Delta y}{y_1} = \frac{\Delta y}{|y|} \]
Sources of Error

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

  - Floating point - uniform precision
  - In practice, 32, 64-bit

  - Scientific notation, \( 2.1 \times 10^{23} \)

- **Propagated Data Error**:

  \[ x \approx x + f(x) \]

  \[ |f(x) - x| / |x| \leq \varepsilon \]

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

  - Truncation error due to finite (approximate) algorithm
  - Method vs true
  - Rounding error = \( \text{floor}(x) \)
Error Analysis

- **Forward Error:**
  \[ \hat{y}(x) - f(x) \]

- **Backward Error:**
  We computed \( \hat{f}(x) \), is there some \( x \) for which \( f(\hat{x}) = \hat{f}(x) \), if so \( \hat{x} - x \) is our backward error
Visualization of Forward and Backward Error

Forward error: $f(x)$

Backward error: $f'(x)$

$x \approx \hat{x}$

$f(x) = f(\hat{x})$
Conditioning of a problem

Absolute Condition Number:

\[ k(f, x) = \lim_{\Delta x \to 0} \frac{|f(x + \Delta x) - f(x)|}{|\Delta x|} = f'(x) \]

\[ k(f) = \max_{x \in \mathbb{R}} k(f, x) \]

(Relative) Condition Number:

\[ k(f, x) = \lim_{\Delta x \to 0} \frac{|f(x + \Delta x) - f(x)|}{|f(x)|} \frac{1}{|\Delta x|} \]

\[ k(f) = \max_{x \in D} k(f, x) \]
Posedness and Conditioning

- What is the condition number of an ill-posed problem?
Stability and Accuracy

- **Accuracy:**

- **Stability:**
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

- Scientific Notation

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

- Demo: Picking apart a floating point number
- Demo: Density of Floating Point Numbers
Rounding Error

- Maximum Relative Representation Error (Machine Epsilon)

Demo: Floating point and the Harmonic Series
Demo: Floating Point and the Series for the Exponential Function
Rounding Error in Operations (I)  

- Addition and Subtraction

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

smaller than or equal to the gap between any two floating point numbers, so

\[ fl(a - b) = 0 \iff fl(a) = fl(b) \]

normalized

\[ 1.xy \cdot 2^E, E \in [-L, L] \]