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

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

Course Structure

- **Complex numerical problems are generally reduced to simpler problems**

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

Numerical Analysis

- **Numerical Problems involving Continuous Phenomena**:

- **Error Analysis**:

Sources of Error

Demo: Floating Point vs Program Logic

- **Representation of Numbers**:

- **Propagated Data Error**:

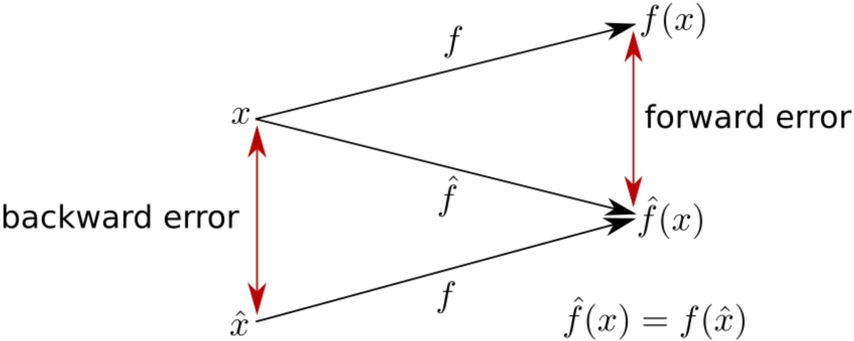
- **Computational Error** = $\hat{f}(x) - f(x)$ = **Truncation Error + Rounding Error**

Error Analysis

- **Forward Error:**

- **Backward Error:**

Visualization of Forward and Backward Error



Conditioning

- **Absolute Condition Number**:

- **(Relative) Condition Number**:

Posedness and Conditioning

- **What is the condition number of an ill-posed problem?**
  - *If the condition number is bounded and the solution is unique, we know the problem is well-posed (solution changes continously)*
  - *In fact, an alternative definition of an ill-posed problem is that f has a condition number of $\kappa_{abs}(f) = \infty$ (meaning f is not differentiable for some input)*
  - *This condition implies that the solutions to a well-posed problem f are continuous and differentiable in the given space of possible inputs to f*
  - *Geometrically, the condition number can be thought of as the reciprocal of the distance (in an appropriate geometric embedding of problem configurations) from f to the nearest ill-posed problem*

Stability and Accuracy

- **Accuracy**:

  An algorithm is accurate if $\hat{f}(x) = f(x)$ for all inputs x when $\hat{f}(x)$ is computed in infinite precision

  - *In other words, the truncation error is zero (rounding error is ignored)*
  - *More generally, an algorithm is accurate if its truncation error is negligible in the desired context*
  - *Yet more generally, the accuracy of an algorithm is expressed in terms of bounds on the magnitude of its truncation error*

- **Stability**:

  An algorithm is stable if its output in finite precision (floating point arithmetic) is always near its output in exact precision

  - *Stability measures the sensitivity of an algorithm to roundoff and truncation error*
  - *In some cases, such as the approximation of a derivative using a finite difference formula, there is a trade-off between stability and accuracy*

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*

- **Addition and Subtraction**

Rounding Error in Operations (II)

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

- **Multiplication and Division**

Exceptional and Subnormal Numbers

- **Exceptional Numbers**

- **Subnormal (Denormal) Number Range**

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

Floating Point Number Line