Computer Arithmetic & Error Analysis

Lecture 2: Floating-Point Systems & Numerical Stability

Francisco Richter Mendoza

Università della Svizzera Italiana
Faculty of Informatics
Lugano, Switzerland

September 8, 2025

Lecture Overview

- **Floating-Point Number Systems**
  - IEEE 754 standard
  - Machine epsilon and precision limits
- **Roundoff Error Analysis**
  - Error accumulation in algorithms
  - Standard model of floating-point arithmetic
- **Catastrophic Cancellation**
  - Sources and examples
  - Avoidance strategies
- **Numerical Stability**
  - Forward vs backward error analysis
  - Algorithm design principles

Floating-Point Number System

Definition

A floating-point system $F(\beta, p, L, U)$ represents numbers as:

$$\pm d_0.d_1 d_2 \dots d_{p-1} \times \beta^e$$

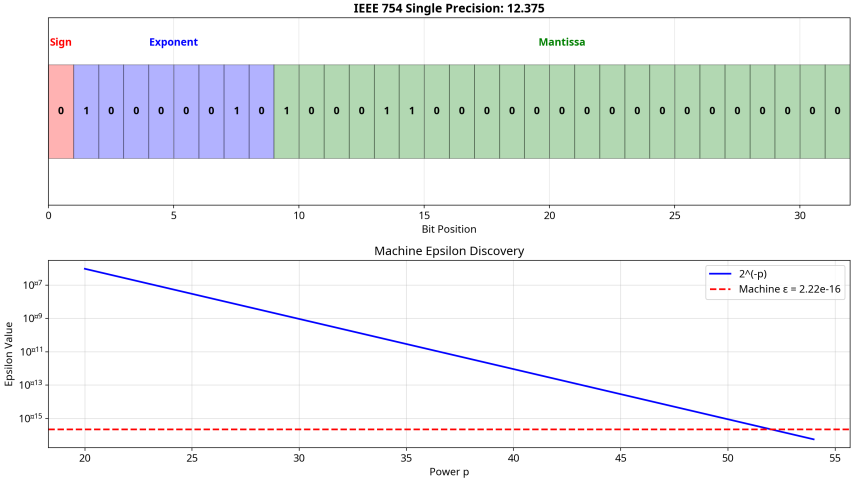
where:

- β: base (typically 2)
- p: precision (significant digits)
- $[L, U]$: exponent range
- $d_0 \neq 0$ (normalized form)

Key Properties:

- Finite representation of real numbers
- Non-uniform spacing: spacing $\approx \beta^{1-p}|x|$
- Relative precision is approximately constant

IEEE 754 Standard



- **Single precision:** 1 sign + 8 exponent + 23 mantissa bits
- **Double precision:** 1 sign + 11 exponent + 52 mantissa bits
- **Machine epsilon:** $\varepsilon_{mach} = 2^{-23} \approx 1.19 \times 10^{-7}$ (single)

Machine Epsilon

Definition (Machine Epsilon)

The machine epsilon ε_{mach} is the smallest positive number such that:

$$1 + \varepsilon_{mach} > 1 \text{ in floating-point arithmetic}$$

IEEE 754 Values:

$$\text{Single precision:} \quad \varepsilon_{mach} = 2^{-23} \approx 1.19 \times 10^{-7} \tag{1}$$

$$\text{Double precision:} \quad \varepsilon_{mach} = 2^{-52} \approx 2.22 \times 10^{-16} \tag{2}$$

Fundamental Guarantee:

$$fl(x) = x(1 + \delta), \quad |\delta| \leq \varepsilon_{mach}$$

Standard Model of Floating-Point Arithmetic

Theorem (IEEE 754 Arithmetic Model)

For floating-point numbers x, y and operation $\circ \in \{+, -, \times, \div\}$:

$$fl(x \circ y) = (x \circ y)(1 + \delta), \quad |\delta| \leq \varepsilon_{mach}$$

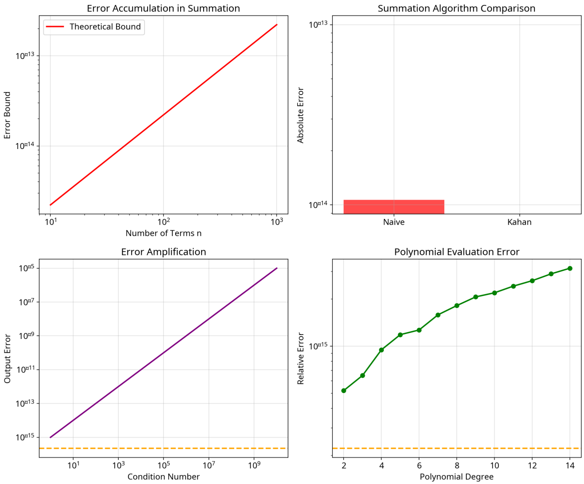
Implications:

- Each operation introduces relative error $\leq \varepsilon_{mach}$
- Errors accumulate through algorithm execution
- Need to analyze error propagation

Error Accumulation Example:

$$fl(((x_1 + x_2) + x_3) + \cdots + x_n) \text{ vs } \sum_{i=1}^{n} x_i$$

Roundoff Error Analysis



- **Error accumulation** grows with problem size
- **Kahan summation** reduces accumulated error
- **Condition number** amplifies input errors

Error Accumulation in Summation

Theorem (Summation Error Bound)

For floating-point summation $s_n = \sum_{i=1}^{n} x_i$:

$$|fl(s_n) - s_n| \leq \gamma_n \sum_{i=1}^{n} |x_i|$$

where $\gamma_n = \frac{n\varepsilon_{mach}}{1 - n\varepsilon_{mach}}$ for $n\varepsilon_{mach} < 1$.

Key Insights:

- Error bound grows linearly with n
- Relative error depends on data magnitude
- Algorithm order matters for accuracy

Catastrophic Cancellation

Definition (Catastrophic Cancellation)

Loss of precision when subtracting nearly equal floating-point numbers.
If $x \approx y$ with p significant digits, then $x - y$ may have $\ll p$ significant digits.

Classic Example:

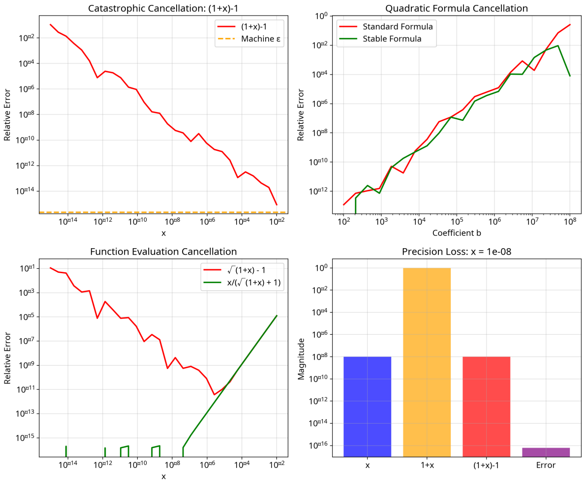
$$\text{Unstable:} \quad (1 + x) - 1 \text{ for small } x \tag{3}$$

$$\text{Stable:} \quad x \tag{4}$$

Error Amplification:

$$\frac{\text{relative error in } (x - y)}{\text{relative error in } x, y} \approx \frac{|x|}{|x - y|} \gg 1$$

Catastrophic Cancellation Examples



- **Function evaluation:** $\sqrt{1+x}-1$ vs $\frac{x}{\sqrt{1+x}+1}$
- **Quadratic formula:** Standard vs numerically stable forms
- **Precision loss:** Dramatic error growth for small differences

Quadratic Formula: Stable Implementation

Problem: Solve $ax^2 + bx + c = 0$ when $b^2 \gg 4ac$

Standard Formula (Unstable):

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Stable Alternative:

$$x_1 = \frac{-2c}{b + \text{sign}(b)\sqrt{b^2 - 4ac}} \tag{5}$$

$$x_2 = \frac{c}{ax_1} \tag{6}$$

Key Principle: Avoid subtracting nearly equal quantities

Forward vs Backward Error Analysis

Definition (Error Types)

For algorithm f computing $\tilde{y} = \tilde{f}(x)$:

- **Forward Error:** $\|f(x) - \tilde{f}(x)\|$
- **Backward Error:** $\min\{\|\Delta x\| : f(x + \Delta x) = \tilde{f}(x)\}$

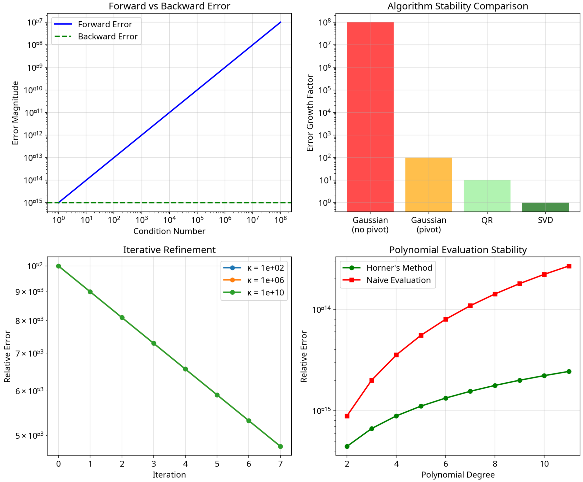
Relationship:

$$\text{Forward Error} \leq \kappa(f) \times \text{Backward Error}$$

where $\kappa(f)$ is the condition number.

Numerical Stability: Algorithm is stable if backward error is small.

Numerical Stability Analysis



- **Error amplification** depends on condition number
- **Algorithm choice** affects stability significantly
- **Iterative refinement** can improve accuracy

Algorithm Stability Comparison

Algorithm	Stability	Error Growth
Gaussian Elimination (no pivoting)	Unstable	$O(2^n)$
Gaussian Elimination (partial pivoting)	Stable	$O(n^3)$
QR Factorization	Stable	$O(n)$
SVD	Very Stable	$O(1)$

Design Principles:

- Minimize operations on ill-conditioned quantities
- Use orthogonal transformations when possible
- Implement pivoting strategies
- Consider iterative refinement

Practical Guidelines for Numerical Stability

Algorithm Design:

- Avoid subtracting nearly equal numbers
- Use mathematically equivalent but numerically stable formulations
- Implement appropriate scaling and pivoting
- Consider higher precision for critical computations

Error Control Strategies:

- Monitor condition numbers
- Use iterative refinement
- Implement compensated summation (Kahan algorithm)
- Validate results with backward error analysis

Remember: Numerical stability is as important as mathematical correctness!

Key Takeaways

1. **Floating-point arithmetic** has fundamental limitations
   - Machine epsilon bounds relative precision
   - IEEE 754 provides standardized behavior
2. **Error analysis** is essential for reliable computation
   - Roundoff errors accumulate through algorithms
   - Condition numbers amplify input errors
3. **Catastrophic cancellation** must be avoided
   - Reformulate mathematically equivalent expressions
   - Use stable algorithms and implementations
4. **Numerical stability** guides algorithm design
   - Backward error analysis provides stability measures
   - Choose algorithms based on stability properties

Next Lecture: Nonlinear Equations and Root-Finding Methods