Numerical Linear Algebra II: 01 Introduction

What Does Numerical Mean?

Jakub Kruzik
jakub.kruzik@vsb.cz

September 14, 2025
Ostrava

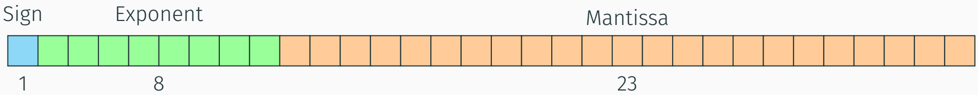
Representation of Numbers on Computers - Binary Floating Point (IEEE 754)

Representation:

$$(-1)^{\text{sign}} \times (1.\text{mantissa}) \times 2^{\text{exponent}-\text{bias}}$$

Single precision (32-bit):

- Sign: 1 bit
- Exponent: 8 bits (bias = 127)
- Mantissa: 23 bits



Special values:

- Exponent all 0s $\Rightarrow$ denormals / 0
- Exponent all 1s $\Rightarrow \infty$ / NaN

Example: Convert 5.75 to Single Precision (IEEE 754)

Step 1: Convert to binary

$$5.75_{10} = 101.11_2$$

Step 2: Normalize to scientific notation (base 2)

$$101.11_2 = 1.0111_2 \times 2^2$$

Step 3: Determine sign, exponent, and mantissa

- Sign = 0 (positive number)
- Exponent = $2 + 127 = 129 = 10000001_2$
- Mantissa = digits after the leading 1: 01110000000000000000000

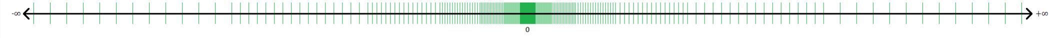
Step 4: Combine into 32-bit IEEE 754 single precision

$$\underbrace{0}_{\text{sign}}\ \underbrace{10000001}_{\text{exponent}}\ \underbrace{01110000000000000000000}_{\text{mantissa}}$$

FP Formats Properties

Format	Precision	Digits (bin)	Digits (dec)	Exp. range	Max value	Min normal
binary16	Half	11	3.31	$[-14, +15]$	6.55×10^{4}	6.10×10^{-5}
binary32	Single	24	7.22	$[-126, +127]$	3.40×10^{38}	1.18×10^{-38}
binary64	Double	53	15.95	$[-1022, +1023]$	1.80×10^{308}	2.23×10^{-308}
binary128	Quadruple	113	34.02	$[-16382, +16383]$	1.19×10^{4932}	3.36×10^{-4932}

Representable numbers:



Beware Rounding:

$$0.1 + 0.2 == 0.3?$$

Beware Associativity:

$$(10^{16} - 10^{16}) + 1.0 = 1.0 \qquad 10^{16} + (-10^{16} + 1.0) = 0.0$$

$$(10^{300} \times 10^{300}) \times 10^{-300} = +\infty \qquad 10^{300} \times (10^{300} \times 10^{-300}) = 10^{300}$$

Forward vs Backward Error (General)

Given $f : \mathbb{R}^n \to \mathbb{R}^n$

$$\boldsymbol{y} = f(\boldsymbol{x})$$

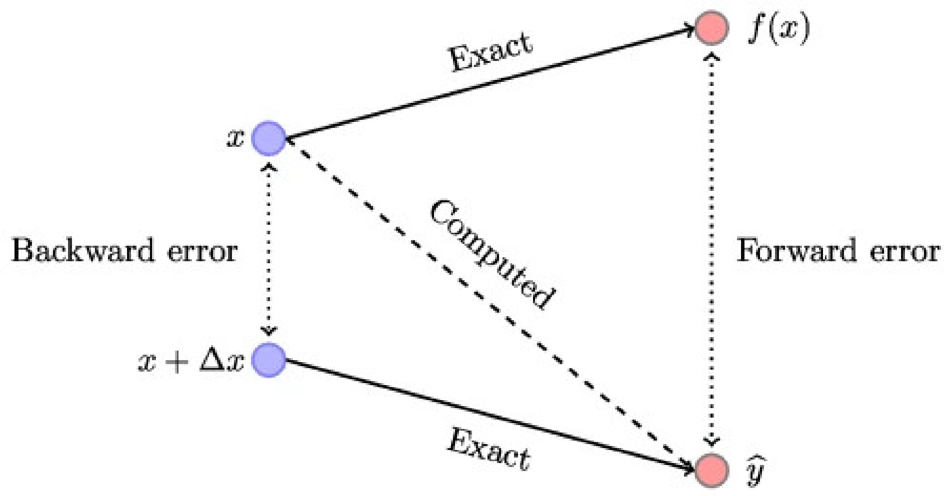
- **Forward Error:** Difference between computed solution $\widehat{\boldsymbol{y}}$ and true solution $\boldsymbol{y}$:

$$\epsilon_{\text{forward}} = ||\widehat{\boldsymbol{y}} - \boldsymbol{y}||$$

- **Backward Error:** Smallest perturbation $\Delta \boldsymbol{x}$ that makes $\widehat{\boldsymbol{y}}$ exact:

$$\epsilon_b(\widehat{\boldsymbol{y}}) = \min\{\epsilon : \widehat{\boldsymbol{y}} = f(\boldsymbol{x} + \Delta \boldsymbol{x}),\ ||\Delta \boldsymbol{x}|| \leq \epsilon ||\boldsymbol{x}||\}$$

Forward vs Backward Error (General)



source: https://nhigham.com/2020/03/25/what-is-backward-error/

Forward vs Backward Error for Linear Systems

Given

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \text{ where } \boldsymbol{A} \in \mathbb{R}^{n\times n} \text{ and } \boldsymbol{b} \in \mathbb{R}^n.$$

- **Forward Error:** Difference between computed solution $\widehat{\boldsymbol{x}}$ and true solution $\boldsymbol{x}$:

$$\epsilon_{\text{forward}} = ||\Delta \boldsymbol{x}|| = ||\widehat{\boldsymbol{x}} - \boldsymbol{x}||$$

- **Residual:** How far $\widehat{\boldsymbol{x}}$ fails to satisfy the original system:

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}$$

- **Relative Backward Error:** Smallest perturbation $\Delta\boldsymbol{A}, \Delta\boldsymbol{b}$ that makes $\widehat{\boldsymbol{x}}$ exact:

$$\epsilon_b(\widehat{\boldsymbol{x}}) = \min\{\epsilon : (\boldsymbol{A} + \Delta\boldsymbol{A})\widehat{\boldsymbol{x}} = \boldsymbol{b} + \Delta\boldsymbol{b},\ ||\Delta\boldsymbol{A}|| \leq \epsilon||\boldsymbol{A}||,\ ||\Delta\boldsymbol{b}|| \leq \epsilon||\boldsymbol{b}||\}$$

It can be shown:

$$\epsilon_b(\widehat{\boldsymbol{x}}) = \frac{||\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}||}{||\boldsymbol{b}|| + ||\boldsymbol{A}||\,||\widehat{\boldsymbol{x}}||} = \frac{||\Delta\boldsymbol{A}_{\text{min}}||}{||\Delta\boldsymbol{A}||} = \frac{||\Delta\boldsymbol{b}_{\text{min}}||}{||\Delta\boldsymbol{b}||}$$

Why Backward Error?

1. **Computable without knowing the true solution**
    - Forward error needs $\widehat{\boldsymbol{x}}$ (the exact solution), which we don't know.
    - Backward error only needs $\boldsymbol{A}$, $\boldsymbol{b}$, and $\widehat{\boldsymbol{x}}$ (the computed solution).
2. **Natural measure of algorithm stability**
    - An algorithm is *backward stable* if it solves a nearby problem exactly:

    $$(\boldsymbol{A} + \Delta\boldsymbol{A})\widehat{\boldsymbol{x}} = \boldsymbol{b} + \Delta\boldsymbol{b}, \quad \|\Delta\boldsymbol{A}\|, \|\Delta\boldsymbol{b}\| \text{ small.}$$

    - This matches how floating-point arithmetic perturbs data.
3. **Separates problem difficulty from algorithm quality** (next slide)
    - Forward error $\leq \kappa(A)\cdot$ (backward error),
    - $\kappa(A)$ = sensitivity of the problem,
    - backward error = performance of the algorithm.

Error Analysis - Perturbation of only $\boldsymbol{b}$

$\widehat{\boldsymbol{x}}$ is the solution of the perturbed system

$$\boldsymbol{A}\widehat{\boldsymbol{x}} = \boldsymbol{A}\left(\boldsymbol{x} + \Delta\boldsymbol{x}\right) = \boldsymbol{b} + \Delta\boldsymbol{b}$$

Subtracting $\boldsymbol{A}\boldsymbol{x}(= \boldsymbol{b})$ from both sides, we get

$$\boldsymbol{A}\Delta\boldsymbol{x} = \Delta\boldsymbol{b} \quad \Rightarrow \quad \Delta\boldsymbol{x} = \boldsymbol{A}^{-1}\Delta\boldsymbol{b}$$

Take norms:

$$||\Delta\boldsymbol{x}|| = ||\boldsymbol{A}^{-1}\Delta\boldsymbol{b}|| \leq ||\boldsymbol{A}^{-1}||\,||\Delta\boldsymbol{b}|| \tag{1}$$

Use $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ again, to estimate

$$||\boldsymbol{b}|| = ||\boldsymbol{A}\boldsymbol{x}|| \leq ||\boldsymbol{A}||\,||\boldsymbol{x}|| \Rightarrow \frac{1}{||\boldsymbol{x}||} \leq ||\boldsymbol{A}||\frac{1}{||\boldsymbol{b}||} \tag{2}$$

Combining (1) and (2) gives the estimate of the relative forward error

$$\frac{||\Delta\boldsymbol{x}||}{||\boldsymbol{x}||} \leq ||\boldsymbol{A}||\,||\boldsymbol{A}^{-1}||\frac{||\Delta\boldsymbol{b}||}{||\boldsymbol{b}||}$$

Error Analysis - Perturbation of only $\boldsymbol{b}$

In this case, the unique perturbation $\Delta \boldsymbol{b}$

$$\boldsymbol{A}\widehat{\boldsymbol{x}} = \boldsymbol{b} + \Delta \boldsymbol{b} \Rightarrow \Delta \boldsymbol{b} = \boldsymbol{A}\widehat{\boldsymbol{x}} - \boldsymbol{b} = -\widehat{\boldsymbol{r}},$$

where $\widehat{\boldsymbol{r}}$ is the residual.

Then

$$\frac{||\Delta \boldsymbol{x}||}{||\boldsymbol{x}||} \leq ||\boldsymbol{A}||\,||\boldsymbol{A}^{-1}||\frac{||\Delta \boldsymbol{b}||}{||\boldsymbol{b}||} = ||\boldsymbol{A}||\,||\boldsymbol{A}^{-1}||\frac{||\widehat{\boldsymbol{r}}||}{||\boldsymbol{b}||}$$

$\kappa(\boldsymbol{A}) = ||\boldsymbol{A}||\,||\boldsymbol{A}^{-1}||$ is the condition number.

Recall that $\frac{||\Delta \boldsymbol{b}_{\text{min}}||}{||\boldsymbol{b}||}$ is, in this case, the relative backward error.

Error Analysis - General Case

$$\boldsymbol{A}\widehat{\boldsymbol{x}} = (\boldsymbol{A} + \Delta\boldsymbol{A})(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b} + \Delta\boldsymbol{b}$$

If $||\boldsymbol{A}^{-1}||_p||\Delta\boldsymbol{A}||_p < 1$ then[1]

$$\frac{||\Delta\boldsymbol{x}||_p}{||\boldsymbol{x}||_p} \leq \frac{\kappa_p(\boldsymbol{A})}{1 - \kappa_p(\boldsymbol{A})\frac{||\Delta\boldsymbol{A}||_p}{||\boldsymbol{A}||_p}}\left(\frac{||\boldsymbol{A}||_p}{||\boldsymbol{A}||_p} + \frac{||\Delta\boldsymbol{b}||}{||\boldsymbol{b}||}\right)$$

If $\kappa_p(\boldsymbol{A})$ is small, we say that the linear system is **well conditioned**.

Otherwise, we say that the linear system is **ill conditioned**.

If an algorithm is backward stable, it can be shown that

$$\frac{||\Delta\boldsymbol{x}||_p}{||\boldsymbol{x}||_p} \leq \kappa_p(\boldsymbol{A})\epsilon_M,$$

where ϵ_M is the machine precision ($2^{53} \approx 1.11\mathrm{e}{-16}$ for double).

[1] with any p-norm, i.e., $1 \leq p \leq \infty$. The same can be done on the previous slides.

Key Takeaways

- Forward error measures how far the computed solution is from the true solution.
- Residual measures how well the computed solution satisfies the original system.
- Backward error measures the minimal perturbation in the data making the computed solution exact.
- For linear systems with only $\boldsymbol{b}$ perturbations, relative backward error = norm of relative (to $\boldsymbol{b}$!) residual
- Backward error is a natural measure of algorithm stability.