

# Numerical Instability Caused by Cancellation

Group 2 - Ferschl Martin, Reiter Roman, Zenkic Mirza

December 12, 2025

## 1 Task specification

We aim to evaluate the function

$$f(x) = \exp(x) - 1,$$

on a computer (in double precision arithmetic) for very small values of the argument, i.e., for  $0 < |x| \ll 1$ .

The exercise asks us to:

1. Perform a numerical experiment in double precision using values  $x = 10^{-k}$  for  $k = 1, 2, 3, \dots$ . We may assume that  $\exp(x)$  itself is implemented correctly in double precision. As an “almost exact” reference value for  $f(x)$ , we are asked to use the Taylor expansion of  $\exp(x)$  about  $x = 0$ , for example truncated after the 10th degree. We shall then plot the relative error between the direct evaluation  $\exp(x) - 1$  and the Taylor reference on a logarithmic scale.
2. Explain why the direct evaluation of  $f(x)$  in the form  $\exp(x) - 1$  is numerically unstable for  $|x| \rightarrow 0$ .

## 2 Method

Using the Taylor expansion of the exponential function,

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \Rightarrow \quad \exp(x) - 1 = \sum_{n=1}^{\infty} \frac{x^n}{n!},$$

we approximate  $f(x)$  by truncating the series after  $N$  terms:

$$f_{\text{ref}}(x) := \sum_{n=1}^N \frac{x^n}{n!}.$$

In our computations we use  $N = 10$  as a brute-force reference for small  $|x|$ .

The relative error we report is

$$\text{rel. error}(x) = \frac{|f_{\text{direct}}(x) - f_{\text{ref}}(x)|}{|f_{\text{ref}}(x)|},$$

where  $f_{\text{direct}}(x) = \exp(x) - 1$  uses the built-in exponential function in double precision, and  $f_{\text{ref}}(x)$  is the Taylor reference.

### 3 Implementation

Below is the Python code used to generate the numerical results and the plot. It computes the direct value  $\exp(x) - 1$ , the Taylor reference, prints a table, and saves the figure as `relative_error.png`.

---

```
import numpy as np
import matplotlib.pyplot as plt

def taylor_exp_minus_one(x, n_terms=10):
    x = np.array(x, dtype=np.float64)
    s = np.zeros_like(x)
    term = None
    for k in range(1, n_terms + 1):
        term = x if k == 1 else term * x / k
        s += term
    return s

#  $x = 10^{-k}$ ,  $k = 1..15$ 
k_vals = np.arange(1, 16)
x = 10.0 ** (-k_vals)

f_direct = np.exp(x) - 1.0
f_ref = taylor_exp_minus_one(x, n_terms=10)
rel_err = np.abs(f_direct - f_ref) / np.abs(f_ref)

print(" k      x      direct      taylor      rel. error")
for k, xv, fd, ft, err in zip(k_vals, x, f_direct, f_ref, rel_err):
    print(f"{k:3d}  {xv:8.1e}  {fd: .3e}  {ft: .3e}  {err: .3e}")

plt.figure()
plt.loglog(x, rel_err, "o-")
plt.gca().invert_xaxis()
plt.xlabel(r"$x = 10^{-k}$")
plt.ylabel(r"relative error")
plt.title(r"Relative error of $\exp(x)-1$ vs Taylor reference")
plt.grid(True, which="both", ls="--")
plt.tight_layout()
plt.savefig("relative_error.png", dpi=200)
plt.close()
```

---

## 4 Results

### 4.1 Numerical table

Table 1 shows a sample of the numerical results for representative  $k$  (Values are produced by the Python script).

Table 1: Direct evaluation vs. Taylor reference for  $x = 10^{-k}$ .

$k$	$x$	direct	Taylor ref.	rel. error
1	$1.000 \times 10^{-1}$	$1.052 \times 10^{-1}$	$1.052 \times 10^{-1}$	$7.917 \times 10^{-16}$
2	$1.000 \times 10^{-2}$	$1.005 \times 10^{-2}$	$1.005 \times 10^{-2}$	$1.087 \times 10^{-14}$
3	$1.000 \times 10^{-3}$	$1.001 \times 10^{-3}$	$1.001 \times 10^{-3}$	$4.291 \times 10^{-14}$
4	$1.000 \times 10^{-4}$	$1.000 \times 10^{-4}$	$1.000 \times 10^{-4}$	$4.327 \times 10^{-13}$
5	$1.000 \times 10^{-5}$	$1.000 \times 10^{-5}$	$1.000 \times 10^{-5}$	$9.702 \times 10^{-12}$
6	$1.000 \times 10^{-6}$	$1.000 \times 10^{-6}$	$1.000 \times 10^{-6}$	$3.798 \times 10^{-11}$
7	$1.000 \times 10^{-7}$	$1.000 \times 10^{-7}$	$1.000 \times 10^{-7}$	$5.663 \times 10^{-10}$
8	$1.000 \times 10^{-8}$	$1.000 \times 10^{-8}$	$1.000 \times 10^{-8}$	$1.108 \times 10^{-8}$
9	$1.000 \times 10^{-9}$	$1.000 \times 10^{-9}$	$1.000 \times 10^{-9}$	$8.224 \times 10^{-8}$
10	$1.000 \times 10^{-10}$	$1.000 \times 10^{-10}$	$1.000 \times 10^{-10}$	$8.269 \times 10^{-8}$
11	$1.000 \times 10^{-11}$	$1.000 \times 10^{-11}$	$1.000 \times 10^{-11}$	$8.274 \times 10^{-8}$
12	$1.000 \times 10^{-12}$	$1.000 \times 10^{-12}$	$1.000 \times 10^{-12}$	$8.890 \times 10^{-5}$
13	$1.000 \times 10^{-13}$	$9.992 \times 10^{-14}$	$1.000 \times 10^{-13}$	$7.993 \times 10^{-4}$
14	$1.000 \times 10^{-14}$	$9.992 \times 10^{-15}$	$1.000 \times 10^{-14}$	$7.993 \times 10^{-4}$
15	$1.000 \times 10^{-15}$	$1.110 \times 10^{-15}$	$1.000 \times 10^{-15}$	$1.102 \times 10^{-1}$

## 4.2 Error plot

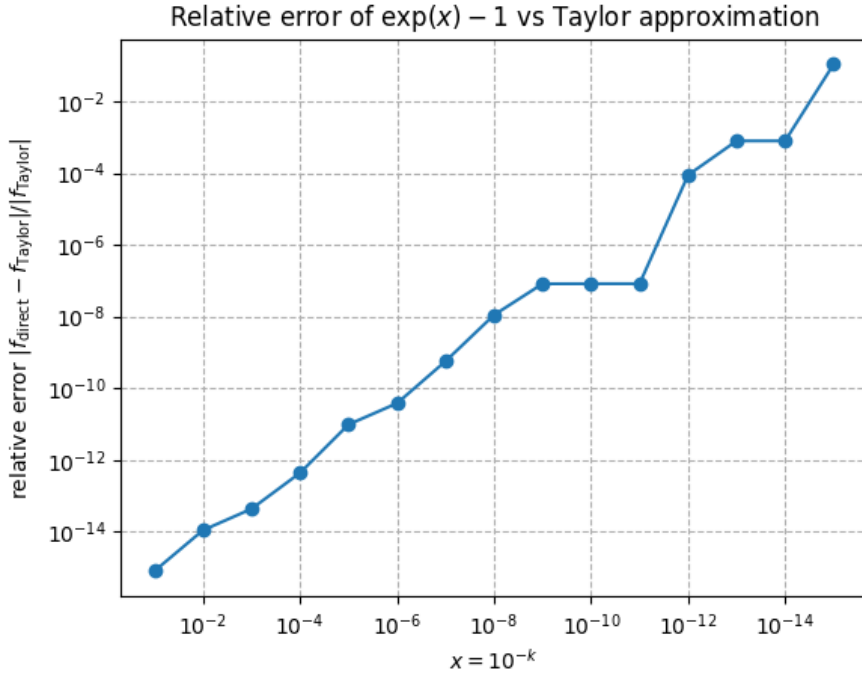


Figure 1: Relative error of the direct evaluation  $\exp(x) - 1$  against the Taylor reference, for  $x = 10^{-k}$ . Both axes use logarithmic scaling (the  $x$ -axis is shown decreasing to the right).

## 5 Discussion

For very small  $x$ , the exponential function satisfies

$$\exp(x) = 1 + x + \frac{x^2}{2} + \cdots.$$

In exact arithmetic, subtracting 1 would give  $\exp(x) - 1 = x + \mathcal{O}(x^2)$ , so the result is of order  $|x|$ .

In floating point arithmetic, however, we can model the situation as follows. For small  $|x|$ , the correctly rounded value of  $\exp(x)$  can be written as

$$\text{fl}(\exp(x)) = 1 + x + \delta,$$

where  $\delta$  is a round-off error of order  $\mathcal{O}(\varepsilon)$ , with  $\varepsilon$  the machine precision. When we form

$$\text{fl}(\exp(x) - 1) = \text{fl}((1 + x + \delta) - 1) \approx x + \delta,$$

the large leading terms 1 cancel, and we are left with the much smaller difference  $x + \delta$ . The absolute error is still on the order of  $|\delta| \approx \varepsilon$ , but the exact value  $\exp(x) - 1 \approx x$  has magnitude  $\mathcal{O}(|x|)$ . Hence the relative error behaves like

$$\frac{|(x + \delta) - x|}{|x|} = \frac{|\delta|}{|x|} \sim \frac{\varepsilon}{|x|},$$

which grows rapidly as  $|x| \rightarrow 0$ . For sufficiently small  $x$ ,  $\exp(x)$  may even round to exactly 1, so the computed value of  $\exp(x) - 1$  is zero, while the exact value is nonzero.

This is an example of *catastrophic cancellation*: subtracting nearly equal numbers eliminates most significant digits from the result and amplifies the effect of rounding errors, even though the underlying mathematical problem (computing  $\exp(x) - 1$ ) is well-conditioned.

## 6 Conclusion

The direct evaluation of  $\exp(x) - 1$  in double precision is numerically unstable for small  $|x|$  because of cancellation of leading digits. Although  $\exp(x)$  itself is computed accurately, the subtraction of 1 causes the relative error in the final result to grow like  $\varepsilon/|x|$  as  $|x| \rightarrow 0$ .

A numerically stable alternative near zero is to use a series expansion such as the truncated Taylor series, or to call a dedicated routine like `expm1(x)` which is implemented to avoid cancellation in this regime.