# Computational Physics Laboratory report

Stefano Ge

Winter semester 2025

## Contents



## 1 Error analysis

#### 1.0.1 Approximating exponentials

**Results**

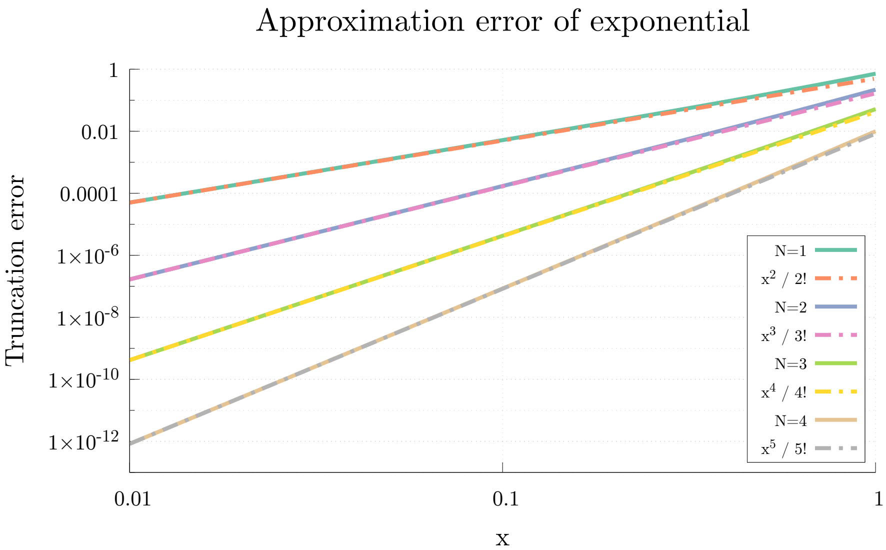


Figure 1: Truncation errors when approximating $e^x$ through the truncated power series $\Sigma_0^N$. $N$ is the degree of the polinomyal used for the approximation. The plot of $x^{N+1}/(N+1)$ has been added for reference.

**Remarks**

- Horner's method was used in polynomial evaluation, because it is faster and reduces roundoff errors. A naive implementation of a polynomial computation requires more operations ($O(N^2)$) and, for small values of $x$, adds together numbers of order 1 and $x^N$, while Horner's method is $O(N)$ and keeps the additions only between terms of order 1 and x.

- By Taylor's theorem, if $\hat{f}(x)$ is the power series of $e^x$ truncated after the N-th term, the error is of order $O(x^{N+1})$ as x approaches 0. Computing an additional term of the series shows that:

$$|e^x - \hat{f}(x)| = \frac{x^{N+1}}{(N+1)!} + o(x^{N+1})$$

  It is then not surprising that the approximation error in Figure 1 is indistinguishable from $x^{N+1}/(N+1)$ for small values of x. For larger values, the contribution of higher order terms in the series becomes ever more important, and hence the divergence from $x^{N+1}/(N+1)$.

## 1.2 Floating-point arithmetic and roundoff errors

### 1.2.1 Computing the Basel problem

**Results**

**Remarks**

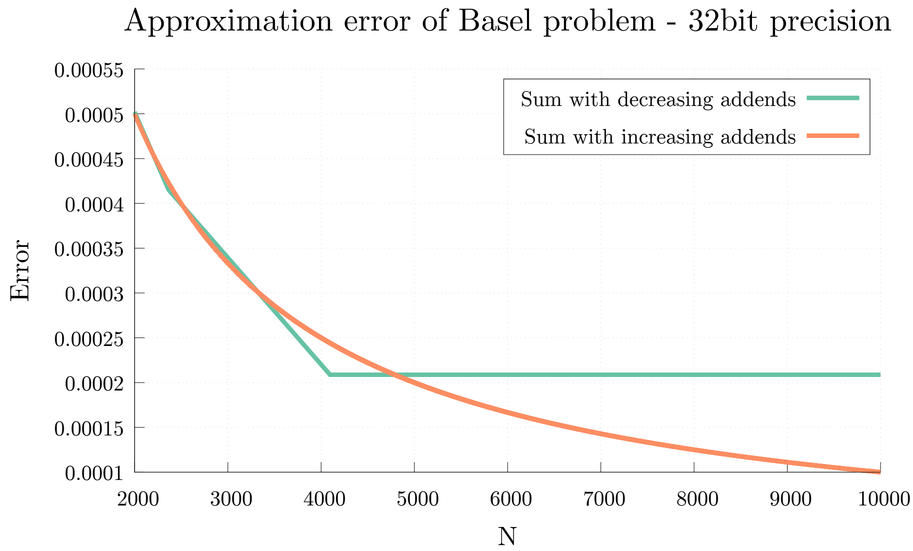


Figure 2: Approximation error when computing $\pi^2/6$ with the Basel problem truncated series $\Sigma_0^N \frac{1}{n^2}$, with single precision floating variables. Here N is the number of terms used when calculating the series.

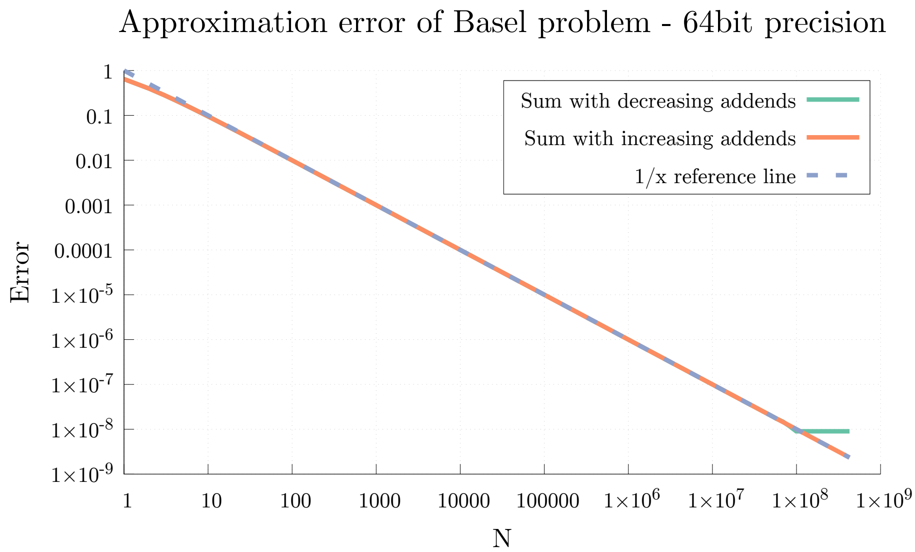


Figure 3: Same as above, but now using double precision floating variables. the plot of $1/x$ has been added for reference.

- The algorithm has been implemented with two mathematically equivalent forms, which, however, resulted in remarkably different numerical results. In the first the terms of the series are summed in increasing order of the summation index, i.e. decreasing of the value of the addends. Since small addends are summed to the accumulating partial sum, it gets to a point where the next addend is smaller than $\epsilon_{mach} \cdot sum \approx \epsilon_{mach}$, so the addition of it and subsequent values does not change the value of the partial sum.
  This can be seen in the error plots in Figure 2 and 3, since the error stops decreasing and remains constant after $1/\mathrm{N}^2 = \epsilon_{\mathrm{mach}}$, which is $2^{-24}$ for single precision ($\mathrm{N} = 4096$) and $2^{-53}$ for double precision ($\mathrm{N} \approx 9.5 \times 10^7$).

  In the second implementation the terms of the series are summed in decreasing order of the summation index, i.e. increasing the value of the addends. Therefore, the aforementioned effect is not seen, and the numerical results correspond to the mathematical expectation, in the range considered.

- The truncation error of this algorithm is asymptotically equivalent to $k/N$, with k real constant. This can be from the integral associated to $\Sigma_{n=0}^{\infty}$:

  $$\int_{N}^{\infty} \frac{1}{n^2} = \frac{1}{N}$$

  therefore,

  $$\Sigma_{n=N}^{\infty} \sim_{N \to \infty} \frac{1}{N}$$

  which is confirmed from the agreement between the computed values and the $1/x$ reference line in Figure 3.

### 1.3 Error propagation and condition number

#### 1.3.1 Computing statistical momenta

#### 1.3.2 Condition number: study of a simple algorithm

## 2 Linear systems

test text

### 2.1 Forward- and back-substitution

### 2.2 LUP Decomposition

This is test text

### 2.3

# 3 Interpolation

# 4 Roots of nonlinear equations

# 5 Numerical integration

## 5.1 Newton-Cotes formula

### 5.1.1 Trapezoidal rule

test

### 5.1.2 Simpson's rule

test

## 5.2 Free-nodes integration

### 5.2.1 Nodes and weights of Gauss-Legendre rule

something

#### Remarks

### 5.2.2 Integrals with Gauss-Legendre rule

## 5.3 Advanced topics in integration

# 6 Ordinary differential equations