ENUME

Numerical Methods

Assignment A
Project #27

Accuracy of computation

Author: Raman Kulpeksha (330240)

Advisor: Dr. Jakub Wagner

Faculty of Electronics and Information Technology

Warsaw University of Technology

Warsaw

16.04.2024

I declare that this piece of work, which is the basis for recognition of achieving learning outcomes in the Numerical Methods course, was completed on my own.

Table of Contents

Notation

This section is devoted to definitions of all symbols and acronyms, which can be found throughout the assignment report.

Basic variables

x *– general purpose independent variable*

y *– general purpose dependent variable*

$f(*)$ *– general purpose function, often:* $y = f(x)$

ε *– relative error, corrupting data due to their floating-point representation, shortly: representation error*

η *– relative error, corrupting the result of a floating-point operation, shortly: rounding error*

$T(x)$ *– a function, which describes propagation of representation error across a scalar function of a scalar variable* x

$K(x)$ *– a function, which describes propagation of rounding error across a scalar function of a scalar variable* x

Modifications of basic variables

$\dot{x}$ *– pure, error-free value of a general purpose independent variable*

$\tilde{x}$ *– error-corrupted value*

The variables associated with y are generated in analogous way.

Operators

$\delta[*]$ *– operator for determining the* ***relative error***, $\delta[y] = \frac{\tilde{y}-\dot{y}}{\dot{y}}$

$\exp(*)$ *– operator for computing exponential function, e.g.* $\exp(x) = e^x$

Miscellaneous

– MATLAB code

Theoretical introduction

In numerical methods, the accuracy of computations is crucial for obtaining reliable results. However, due to limitations in computer hardware and numerical algorithms, errors cannot avoided. Two primary sources of errors in numerical computations are **data errors** and **rounding errors**.

1. Data Errors:

Data errors are caused by inaccuracies or uncertainties in the input data used in computations. These errors can arise from various sources, namely

- Measurement inaccuracies
- Truncation of data
- Approximations made during data collection

It is essential to understand the nature and magnitude of data errors to properly assess their impact on the accuracy of the computed results.
In computations, the function used for describing data errors caused by propagation of representation error is denoted as $T(x)$.

There are 2 general ways to obtain $T(x)$

1. Method of symbolic differentiation

In general case, $T(x)$ can be computed the following way:

$$T(x) = \frac{x}{y} \cdot \frac{dy}{dx}, \tag{1}$$

2. Epsilon calculus

In some scenarios, where using the formula provided above may require excessively complex calculations, **method of epsilon calculus** could be used.

Briefly, we shall represent x as error-corrupted variable $\tilde{x}$, expressed as:

$$\tilde{x} = \dot{x}(1 + \varepsilon_x), \tag{2}$$

where $\dot{x}$ denotes error-free interpretation of x and ε_x represents the error.

Next step is to compute $\tilde{y} = f(\tilde{x})$. By eliminating common elements, we shall simplify it to:

$$\tilde{y} = \dot{y}(1 + T(x)\varepsilon_x), \tag{3}$$

and hence, obtain $T(x)$.

Both methods will be demonstrated in solution for Task 1.

2. Rounding Errors:

Rounding errors occur due to the finite precision of numerical representations in computer systems. Computers store real numbers using finite binary representations, which can lead to rounding and truncation of digits. As a result, arithmetic operations involving real numbers may introduce small errors due to the inability to represent all digits accurately.

Rounding errors can accumulate during lengthy computations and significantly affect the final results, particularly in iterative algorithms or computations involving a large number of operations.

Estimating rounding errors, similarly to representation errors, we are looking for a special function, which would illustrate it. This function is denoted as $K(x)$.

A distinctive feature of rounding errors is their accumulative nature. Since we need to store the results of intermediate calculations during the calculation, the error has a cumulative character. Practically, this entails the necessity of considering this kind of error for every intermediate operation, including +, -, *, /, [2], etc.

3. Floating point representation. IEEE 754 standard

When using computer systems in calculations, one must keep in mind the finiteness of the amount of memory that can be allocated to store a given number. Therefore, we cannot operate on the exact values of irrational numbers or numbers with a large number of digits. Otherwise, we simply would not be able to operate on them.

MATLAB is based on *IEEE 754* standard. To store a number using *IEEE 754* standard, it has to be converted to a **scientific notation**[1], also known as **standard form**. After transformation, we obtain a value in form $n \cdot 10^E$, where $|n|$ is in $(1, 10)$ and E – a real number. The former is called **mantissa** and the latter – **exponent**. A distinctive feature of IEEE 754 is the presence of two data types for storing floating point numbers, *i.e.*

Data type	Stored as
single-precision	sign bit, 8-bit exponent, 23-bit mantissa
double-precision	sign bit, 11-bit exponent, 52-bit mantissa[2]

Table 1: Data types for storing floating point numbers in IEEE 754 standard

A greater amount of bits guarantees a more accurate and reliable interpretation of value.

[1] https://en.wikipedia.org/wiki/Scientific_notation
[2] https://learn.microsoft.com/en-us/cpp/build/ieee-floating-point-representation?view=msvc-170

4. Rules for epsilon calculus

Computing both rounding and data errors, one must remember that they are significantly small ($<<$ 1). Thus, some special rules are applicable, namely:

$$(1 + \varepsilon_1)(1 + \varepsilon_2) \cong 1 + \varepsilon_1 + \varepsilon_2, \tag{4}$$

$$(1 + \varepsilon)^a \cong 1 + a\varepsilon \text{ for } |a| \ll eps^{-1}, \tag{5}$$

$$\ln(1 + \varepsilon) \cong \varepsilon, \tag{6}$$

$$e^{1+\varepsilon} \cong (1 + \varepsilon)e, \tag{7}$$

Formulation of problem

Task #1

Determine the function $T\ (x)$ characterising the propagation of the relative error corrupting the values of $x\ \in\ [10^{-2}, 10^{2}]$ to the result of computing the following expression:

$$y\ =\ f\ (x) \equiv\ \frac{\exp(x)}{x^{2}} - x^{3}$$

Compare the formulae of the function $T\ (x)$, determined using the method of epsilon calculus and the method of symbolic differentiation.

Task #2

Determine the relative errors corrupting the values of y computed on the basis of the values $\{x_1, \ldots, x_N\}$ of x stored using the single-precision floating-point representation, where:

$$10^{-2} = x_1 \leq x_2 \leq \ldots \leq x^N = 10^2, N \in \mathbb{N}$$

Proceed as follows:

a) compute the values $\{y_1, \ldots, y_N\}$:

$$y_n\ =\ f\ (x_n)\ \ for\ n\ =\ 1, \ldots, N$$

using the double-precision representation (default in MATLAB);

b) compute the values $\{\tilde{y}_1, \ldots, \tilde{y}_N\}$:

$$\tilde{y}_n\ =\ f\ (\tilde{x}_n)\ \ for\ n\ =\ 1, \ldots, N$$

where $\tilde{x}_n$ denotes the result of storing x_n using the single-precision representation (single);

perform the computation using the double-precision representation (double);

c) compute the relative errors according to the formula:

$$\delta[\tilde{y}_n] = \frac{\tilde{y}_n - y_n}{y_n}\ \ for\ n\ =\ 1, \ldots, N$$

Compare the computed errors with the worst-case estimate $|T\ (x)|eps_{single}$, where eps_{single} denotes the upper bound of the relative errors corrupting numeric values stored using the single-precision representation.

Task #3

Using the method of epsilon calculus, determine the functions $K_{A_1}(x)$ and $K_{A_2}(x)$, characterising the propagation of the relative errors caused by rounding the intermediate results of computing $\{y_1, \ldots, y_N\}$ according to the following algorithms:

$$A_1: [x] \rightarrow \begin{bmatrix} v_1 = \exp(x) \\ v_2 = x^2 \end{bmatrix} \rightarrow \begin{bmatrix} v_3 = \dfrac{v_1}{v_2} \\ v_4 = x^3 \end{bmatrix} \rightarrow [v_5 = v_3 - v_4] \rightarrow [y]$$

$$A_2: [x] \rightarrow \begin{bmatrix} v_1 = \exp(x) \\ v_2 = x^5 \end{bmatrix} \rightarrow \begin{bmatrix} v_3 = v_1 - v_2 \\ v_4 = x^2 \end{bmatrix} \rightarrow \left[v_5 = \frac{v_3}{v_4}\right] \rightarrow [y]$$

Task #4

Determine the relative errors which corrupt the values $\{y_1, \dots, y_N\}$ when the results of all intermediate operations are stored using the single-precision representation. Proceed as follows:

a) compute $\{y_1, \dots, y_N\}$using the double-precision representation (default in MATLAB);

b) by means of the algorithms A_1 and A_2, compute the approximate values $\tilde{y}^{\mathrm{A1}}{}_1$, ..., $\tilde{y}^{\mathrm{A1}}{}_N$ and $\tilde{y}^{\mathrm{A2}}{}_1$, ..., $\tilde{y}^{\mathrm{A1}}{}_N$by performing the intermediate operations one by one and storing their results

using single-precision representation (single);

c) compute the relative errors according to the formulae:

$$\delta\left[\tilde{y}^{\mathrm{A1}}{}_n\right] = \frac{\tilde{y}^{\mathrm{A1}}{}_n - y_n}{y_n}$$

and

$$\delta\left[\tilde{y}^{\mathrm{A2}}{}_n\right] = \frac{\tilde{y}^{\mathrm{A2}}{}_n - y_n}{y_n}$$

for $n = 1, \dots, N$

Compare the computed errors with the worst-case estimates: $K_{A_1}(x)eps_{single}$, $K_{A_2}(x)eps_{single}$.

Task #5

Compare the functions $T(x)$, $K_{A_1}(x)$ and $K_{A_2}(x)$ for $x \in [10^{-2}, 10^2]$

Results

Task 1

The idea of task is to compare $T(x)$ acquired by operating with methods mentioned in Theoretical introduction of this report.

Firstly, we shall use method of symbolic differentiation

$$T(x) = x \cdot \left(\frac{e^x - x^5}{x^2}\right)^{-1} \cdot \frac{d}{dx}\left(\frac{e^x}{x^2} - x^3\right) = \frac{x^3}{e^x - x^5} \cdot \frac{d}{dx}\left(\frac{e^x}{x^2}\right) - \frac{d}{dx}(x^3) = \frac{x^3}{e^x - x^5} \cdot \frac{d}{dx}\left(\frac{e^x}{x^2}\right) - 3x^2, \quad (8)$$

For the sake of clarity, $\frac{d}{dx}\left(\frac{e^x}{x^2}\right)$ is computed separately:

$$\frac{d}{dx}\left(\frac{e^x}{x^2}\right) = \frac{\frac{d}{dx}(e^x)x^2 - e^x \cdot \frac{d}{dx}(x^2)}{(x^2)^2} = \frac{e^x x^2 - 2e^x x}{x^4} = \frac{e^x x - 2e^x}{x^3}, \quad (9)$$

Substituting the obtained value, we get:

$$T(x) = \frac{x^3}{e^x - x^5} \cdot \left(\frac{e^x x - 2e^x}{x^3} - 3x^2\right) = \frac{x^3}{e^x - x^5} \cdot \frac{e^x x - 2e^x - 3x^5}{x^3} = \frac{e^x x - 2e^x - 3x^5}{e^x - x^5} \quad (10)$$

To use Epsilon calculus in order to receive $T(x)$, proceed as follows:

$$\tilde{y} = f(\tilde{x}), \quad (11)$$

$$\tilde{x} = x(1 + \varepsilon), \quad (12)$$

$$\tilde{y} = \frac{e^{x(1+\varepsilon)}}{x^2(1+\varepsilon)^2} - x^3(1+\varepsilon)^3, \quad (13)$$

By using the rules of epsilon calculus[3], we get:

$$\tilde{y} = \frac{(1 + x\varepsilon) \cdot e^x}{x^2(1+2\varepsilon)} - x^3(1+3\varepsilon), \quad (14)$$

$$\tilde{y} = \frac{e^x \cdot (1 + x\varepsilon) \cdot (1+2\varepsilon)^{-1}}{x^2} - x^3 - 3x^3\varepsilon, \quad (15)$$

$$\tilde{y} = \frac{e^x \cdot (1 + x\varepsilon) \cdot (1-2\varepsilon)}{x^2} - x^3 - 3x^3\varepsilon, \quad (16)$$

$$\tilde{y} = \frac{e^x \cdot (1 + x\varepsilon - 2\varepsilon)}{x^2} - x^3 - 3x^3\varepsilon, \quad (17)$$

$$\tilde{y} = \frac{e^x + e^x x\varepsilon - 2e^x\varepsilon}{x^2} - x^3 - 3x^3\varepsilon, \quad (18)$$

[3] Formulas (4) – (7)

$$\tilde{y} = \frac{e^x}{x^2} - x^3 + \frac{e^x x\varepsilon - 2e^x\varepsilon - 3x^5\varepsilon}{x^2}, \tag{19}$$

Note: $\dot{y} = \frac{e^x}{x^2} - x^3$

$$\tilde{y} = \dot{y}\left(1 + \frac{e^x x\varepsilon - 2e^x\varepsilon - 3x^5\varepsilon}{x^2} \cdot \frac{x^2}{e^x - x^5}\right), \tag{20}$$

$$\tilde{y} = \dot{y}\left(1 + \frac{e^x x - 2e^x - 3x^5}{e^x - x^5}\varepsilon\right), \tag{21}$$

Taking into consideration the notation $\tilde{y} = \dot{y}(1 + T(x)\varepsilon_x)$, we conclude:

$$T(x) = \frac{e^x x - 2e^x - 3x^5}{e^x - x^5}, \tag{22}$$

and thus, **proving the identity** of $T(x)$ computed by both methods.

The correctness of estimation can be proved by the following MATLAB script ***(Task 2, file task2.m)****:*

```
x = logspace(-2, 2, 1000);
y = (exp(x) ./ x.^2 ) - (x.^3);

% Computing maximum error
Tx = abs(x .* exp(x) - 2 .* exp(x) - 3 * x.^5) ./
(exp(x) - x.^5);
max_abs_error = abs(y .* Tx .* double(eps(single(1))) /
2);
max_rel_error = abs(max_abs_error ./ y) ;

% Computing real error
y_dist = (exp(double(single(x)))./double(single(x)).^2
)-(double(single(x)).^3);
abs_error = abs(y - y_dist);
rel_error = abs(abs_error ./ y);


% Plotting the results
plot(x, rel_error, 'Color', 'red', 'LineWidth', 1.5);
hold on;
plot(x, max_rel_error, 'Color', 'blue', 'LineWidth',
1.5);
set(gca, 'XScale', 'log', 'YScale', 'log');
legend('real error', 'maximum error'), xlabel('x'),
ylabel('δ'), title('Task 2');
```

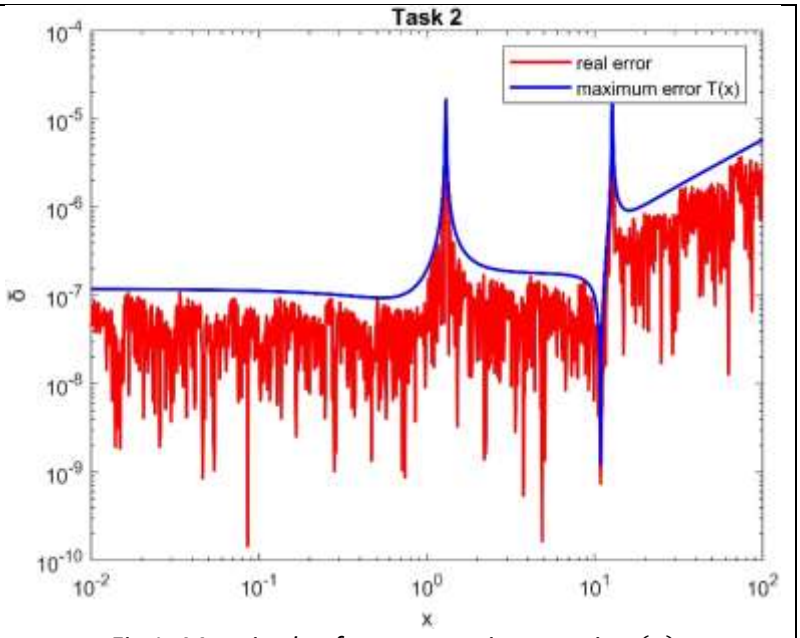

Fig 1. *Magnitude of representative error in* $y(x)$

Task 3

Here, we shall compare the efficiency of computing $y = \frac{\exp(x)}{x^2} - x^3$ using 2 algorithms.
The idea is to compute $K(x)$ for both algorithms to propagation errors.

$$A_1: [x] \to \begin{bmatrix} v_1 = \exp(x) \\ v_2 = x^2 \end{bmatrix} \to \begin{bmatrix} v_3 = \frac{v_1}{v_2} \\ v_4 = x^3 \end{bmatrix} \to [v_5 = v_3 - v_4] \to [y]$$

Starting with the first algorithm, one shall gradually imply propagation errors

$$\tilde{y}_1 = [v_3 - v_4](1 + \eta_1), \tag{23}$$

$$\tilde{y}_1 = \left[\frac{v_1}{v_2}(1 + \eta_2) - x^3(1 + \eta_3)\right](1 + \eta_1), \tag{24}$$

$$\tilde{y}_1 = \left[\frac{\exp(x)\,(1 + \eta_4)}{x^2(1 + \eta_5)}(1 + \eta_2) - x^3(1 + \eta_3)\right](1 + \eta_1), \tag{25}$$

$$\tilde{y}_1 = \left[\frac{\exp(x)}{x^2}(1 + \eta_2 + \eta_4 - \eta_5) - x^3(1 + \eta_3)\right](1 + \eta_1), \tag{26}$$

$$\tilde{y}_1 = \left[\frac{\exp(x)}{x^2} - x^3 + \frac{\exp(x)}{x^2}(\eta_2 + \eta_4 - \eta_5) - x^3\eta_3\right](1 + \eta_1), \tag{27}$$

Note: $\dot{y}_1 = \frac{e^x}{x^2} - x^3$

$$\tilde{y}_1 = \left(\dot{y}_1 + \frac{\exp(x)}{x^2}(\eta_2 + \eta_4 - \eta_5) - x^3\eta_3\right), \tag{28}$$

$$\tilde{y}_1 = \dot{y}_1\left(1 + \left[\frac{\exp(x)}{x^2}(\eta_2 + \eta_4 - \eta_5) - x^3\eta_3\right]\left(\frac{x^2}{\exp(x) - x^5}\right)\right)(1 + \eta_1), \tag{29}$$

$$\tilde{y}_1 = \dot{y}_1\left(1 + \left[\frac{\exp(x)}{\exp(x) - x^5}(\eta_2 + \eta_4 - \eta_5) - \frac{x^5}{\exp(x) - x^5}\eta_3 + \eta_1\right]\right), \tag{30}$$

$$\tilde{y}_1 = \dot{y}_1\left(1 + \left[\frac{\exp(x)}{\exp(x) - x^5}(\eta_2 + \eta_4 - \eta_5) - \frac{x^5}{\exp(x) - x^5}\eta_3 + \eta_1\right]\right), \tag{31}$$

From this definition, we derive:

$$|K_1| = 1, \qquad |K_2| = |K_4| = |K_5| = \left|\frac{\exp(x)}{\exp(x) - x^5}\right|, \quad |K_3| = \left|\frac{x^5}{\exp(x) - x^5}\right|, \tag{32}$$

The estimation of total error can be assessed the following way:

$$|\delta[\tilde{y}_1]| \le |K_1|eps + 3|K_2|eps + |K_3|eps, \tag{33}$$

$$|\delta[\tilde{y}_1]| \le \left(1 + 3\left|\frac{\exp(x)}{\exp(x) - x^5}\right| + \left|\frac{x^5}{\exp(x) - x^5}\right|\right)eps, \tag{34}$$

So, the final function is

$$K_{A_1} = 1 + 3\left|\frac{\exp(x)}{\exp(x) - x^5}\right| + \left|\frac{x^5}{\exp(x) - x^5}\right|, \tag{35}$$

By following similar steps, we compute K_{A_2} as follows:

$$A_2\colon [x] \to \begin{bmatrix} \nu_1 = \exp(x) \\ \nu_2 = x^5 \end{bmatrix} \to \begin{bmatrix} \nu_3 = \nu_1 - \nu_2 \\ \nu_4 = x^2 \end{bmatrix} \to \left[\nu_5 = \frac{\nu_3}{\nu_4}\right] \to [y]$$

$$\tilde{y}_2 = \left(\frac{\nu_3}{\nu_4}\right)(1+\eta_1), \tag{36}$$

$$\tilde{y}_2 = \left(\frac{\nu_1 - \nu_2(1+\eta_2)}{x^2(1+\eta_3)}\right)(1+\eta_1), \tag{37}$$

$$\tilde{y}_2 = \left[\frac{\left(\exp(x)\,(1+\eta_4) - x^5(1+\eta_5)\right)(1+\eta_2)}{x^2(1+\eta_3)}\right](1+\eta_1), \tag{38}$$

$$\tilde{y}_2 = \left[\frac{\left(\exp(x)\,(1+\eta_4) - x^5(1+\eta_5)\right)(1+\eta_2)}{x^2(1+\eta_3)}\right](1+\eta_1), \tag{39}$$

$$\tilde{y}_2 = \left[\frac{\left(\exp(x)\,(1+\eta_4) - x^5(1+\eta_5)\right)(1+\eta_2-\eta_3)}{x^2}\right](1+\eta_1), \tag{40}$$

$$\tilde{y}_2 = \left[\frac{\exp(x) - x^5 + x^5\eta_5 + \exp(x)\,\eta_4}{x^2}\right](1+\eta_1+\eta_2-\eta_3), \tag{41}$$

$$\tilde{y}_2 = \left[\frac{\exp(x) - x^5}{x^2} + \frac{x^5\eta_5 + \exp(x)\,\eta_4}{x^2}\right](1+\eta_1+\eta_2-\eta_3), \tag{42}$$

Note: $\dot{y}_2 = \frac{e^x - x^5}{x^2}$

$$\tilde{y}_2 = \left[\dot{y}_2 + \frac{x^5\eta_5 + \exp(x)\,\eta_4}{x^2}\right](1+\eta_1+\eta_2-\eta_3), \tag{43}$$

$$\tilde{y}_2 = \dot{y}_2\left[1 + \left(\frac{x^5\eta_5 + \exp(x)\,\eta_4}{x^2}\cdot\frac{x^2}{\exp(x) - x^5}\right) + \eta_1 + \eta_2 - \eta_3\right], \tag{44}$$

$$\tilde{y}_2 = \dot{y}_2\left[1 + \frac{x^5\eta_5 + \exp(x)\,\eta_4}{\exp(x) - x^5} + \eta_1 + \eta_2 - \eta_3\right], \tag{45}$$

$$\tilde{y}_2 = \dot{y}_2\left[1 + \frac{x^5}{\exp(x) - x^5}\eta_5 + \frac{\exp(x)}{\exp(x) - x^5}\eta_4 + \eta_1 + \eta_2 - \eta_3\right], \tag{46}$$

$$|K_1| = |K_2| = |K_3| = 1, \qquad |K_4| = \left|\frac{\exp(x)}{\exp(x) - x^5}\right|, \quad |K_5| = \left|\frac{x^5}{\exp(x) - x^5}\right|, \tag{47}$$

$$|\delta[\tilde{y}_1]| \le 3|K_1|eps + |K_2|eps + |K_3|eps, \tag{48}$$

$$|\delta[\tilde{y}_1]| \leq \left(3 + \left|\frac{\exp(x)}{\exp(x) - x^5}\right| + \left|\frac{x^5}{\exp(x) - x^5}\right|\right) eps, \tag{49}$$

Thus,

$$K_{A_2} = 3 + \left|\frac{\exp(x)}{\exp(x) - x^5}\right| + \left|\frac{x^5}{\exp(x) - x^5}\right|, \tag{50}$$

The correctness of estimations can be proved by the following MATLAB scripts ***(Task 4, file task4.m)****:*

```
% Clearing the results of previously executed scripts
clf, clc, clearvars, close all

x = logspace(-2, 2, 100000);
y = (exp(x)./x.^2 )-(x.^3);

% Maximum error, algorithm 1
Ka1 = 1 + 3.*abs(exp(x)./(exp(x) - x.^5)) + abs(x.^5 ./ (exp(x) - x.^5));
max_abs_error_a1 = abs(y .* Ka1 .* double(eps(single(1))) ./ 2);
max_rel_error_a1 = abs(max_abs_error_a1 ./ y);

% Real error algortihm 1
v1 = double(single(exp(x)));
v2 = double(single(x.^2));
v3 = double(single(v1 ./ v2));
v4 = double(single(x.^3));
ys_a1 = double(single(v3 - v4));
abs_error_a1 = abs(y - ys_a1);
rel_error_a1 = abs(abs_error_a1 ./ y);

% Maximum error, algorithm 2
Ka2 = 3 + abs(exp(x) ./ (exp(x) - x.^5))+ abs(x.^5 ./ (exp(x) - x.^5));
max_abs_error_a2 = abs(y .* Ka2 .* double(eps(single(1))) ./ 2);
max_rel_error_a2 = abs(max_abs_error_a2 ./ y);

% Real error algortihm 2
v1 = double(single(exp(x)));
v2 = double(single(x .^5));
v3 = double(single(v1 - v2));
v4 = double(single(x .^2));
ys_a2 = double(single(v3 ./ v4));
abs_error_a2 = abs(y - ys_a2);
rel_error_a2 = abs(abs_error_a2 ./ y);

sgtitle('Task 4')
% Plot errors for A1
subplot(1,2,1)
hold on;
plot(x, rel_error_a1, 'Color', 'red', 'LineWidth', 1.5);
plot(x, max_rel_error_a1, 'Color', 'blue', 'LineWidth',1.5);
set(gca, 'XScale', 'log', 'YScale', 'log');
legend('real error', 'maximum error'), xlabel('x'), ylabel('δ'), title('A1')
hold off

% Save A1 plot as a JPEG file
exportgraphics(gcf, 'Task4A1Plot.jpg', 'Resolution', 1000);

% Plot errors for A2
subplot(1,2,2)
hold on;
plot(x, rel_error_a2, 'Color', 'red', 'LineWidth', 1.5);
plot(x, max_rel_error_a2, 'Color', 'blue', 'LineWidth',1.5);
set(gca, 'XScale', 'log', 'YScale', 'log');
legend('real error', 'maximum error'), xlabel('x'), ylabel('δ'), title('A2')
hold off

% Save A2 plot as a JPEG file
exportgraphics(gcf, 'Task4A2Plot.jpg', 'Resolution', 1000);
```

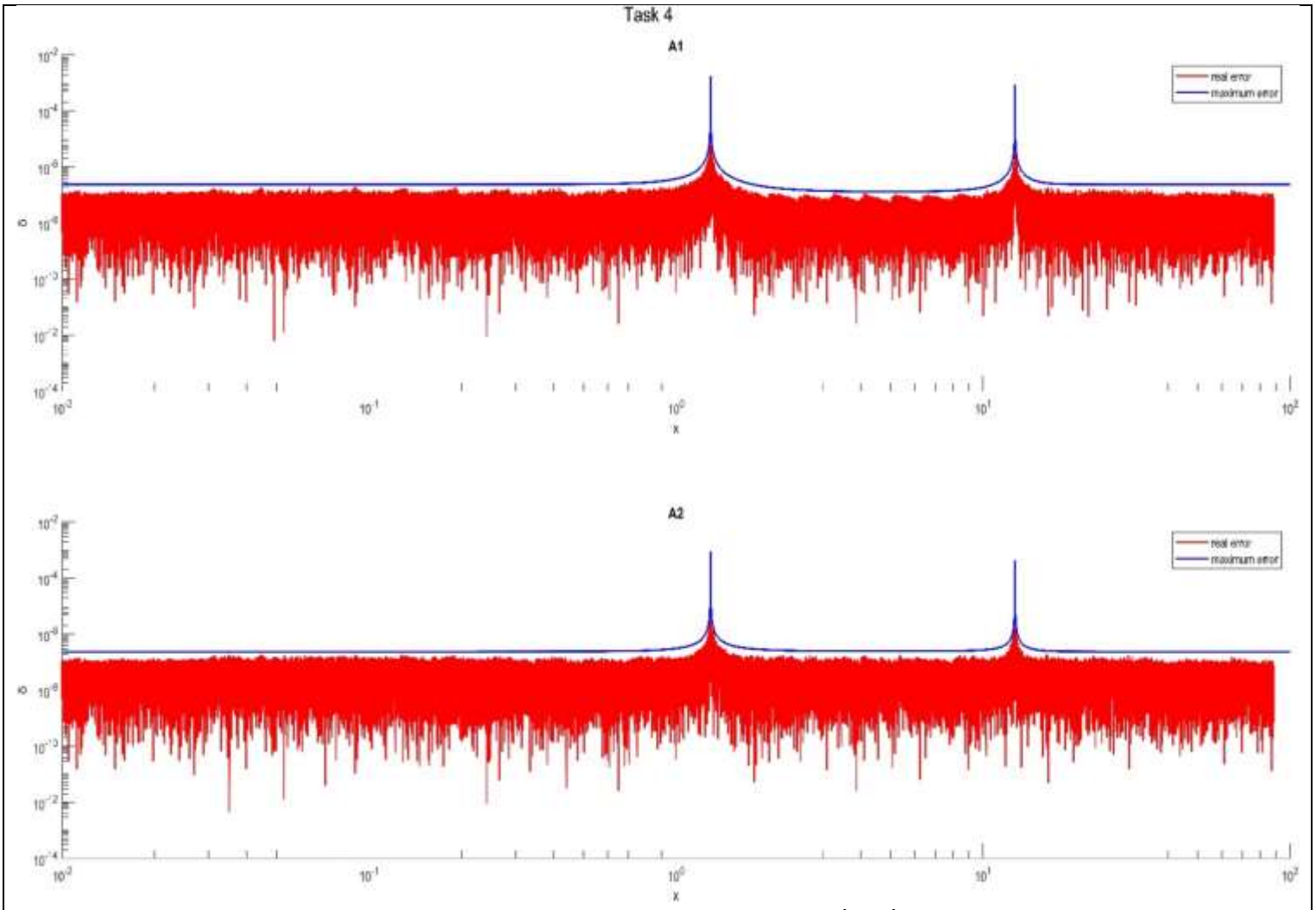

Fig 2.1, 2.2 Magnitude of rounding error in $y(x)$ computed A_1, A_2 using respectively

Discussion

In this report, I have decided to unite parts dedicated to Task 5 with discussion, as the aim of former is to compare previously obtained functions.

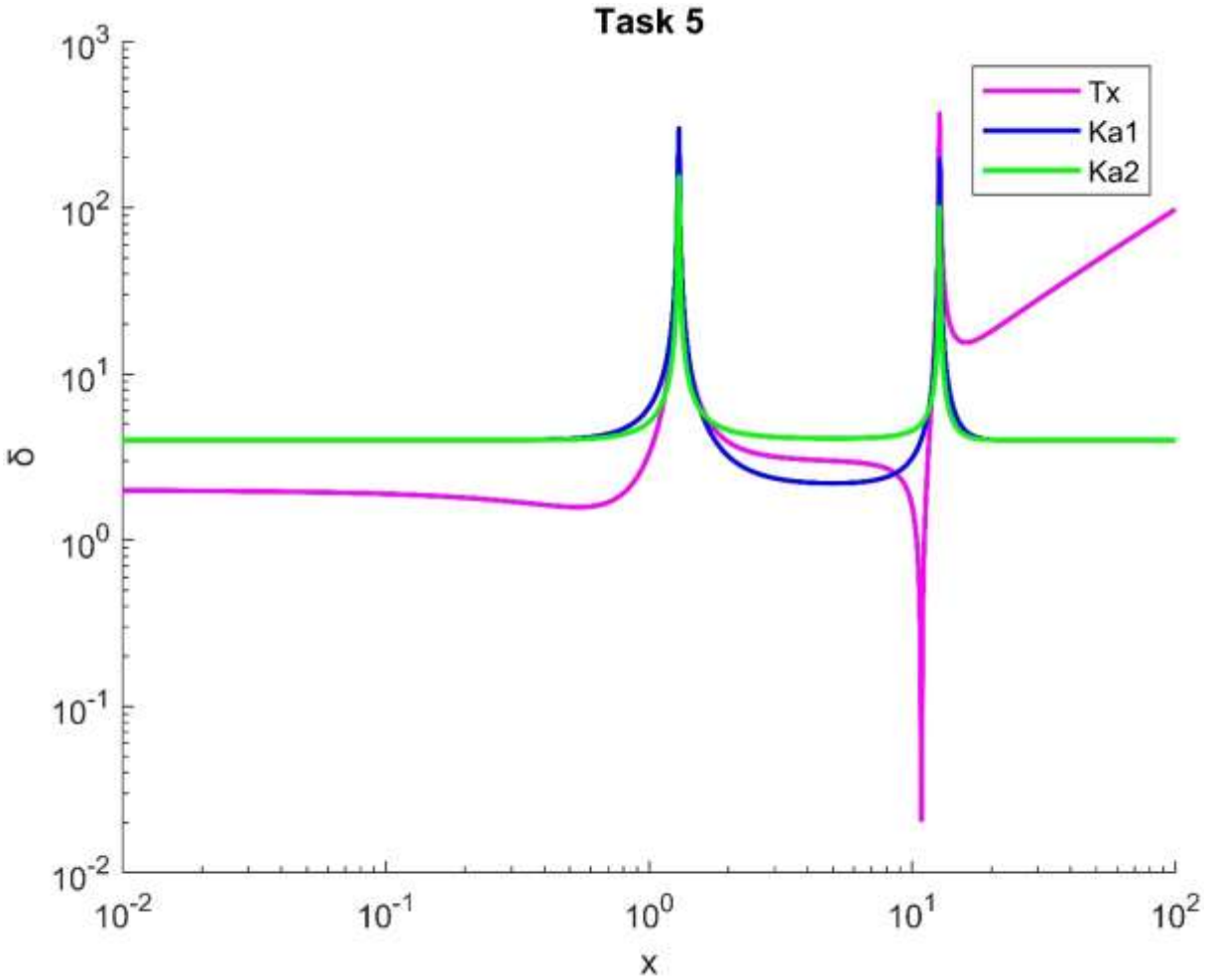

Fig 3 *Comparative characteristic of coefficients* $T(x), K_{A_1}(x), K_{A_2}(x)$

Observing *Fig 3* several significant points and intervals can be distinguished. All 3 errors are peaking at $x \cong 1.3$ and $x \cong 12.7$, their slope is also gradually changing near these values. Trying to find out the reason of such similarity, I have decided to address one common feature for all discussed functions, namely, denominator $e^x - x^5$. Due to the nature of its plot, it is impossible to include a full graph in this report, so I had to use Desmos[4] to illustrate the most 'interesting' part of it.

[4] https://www.desmos.com/calculator

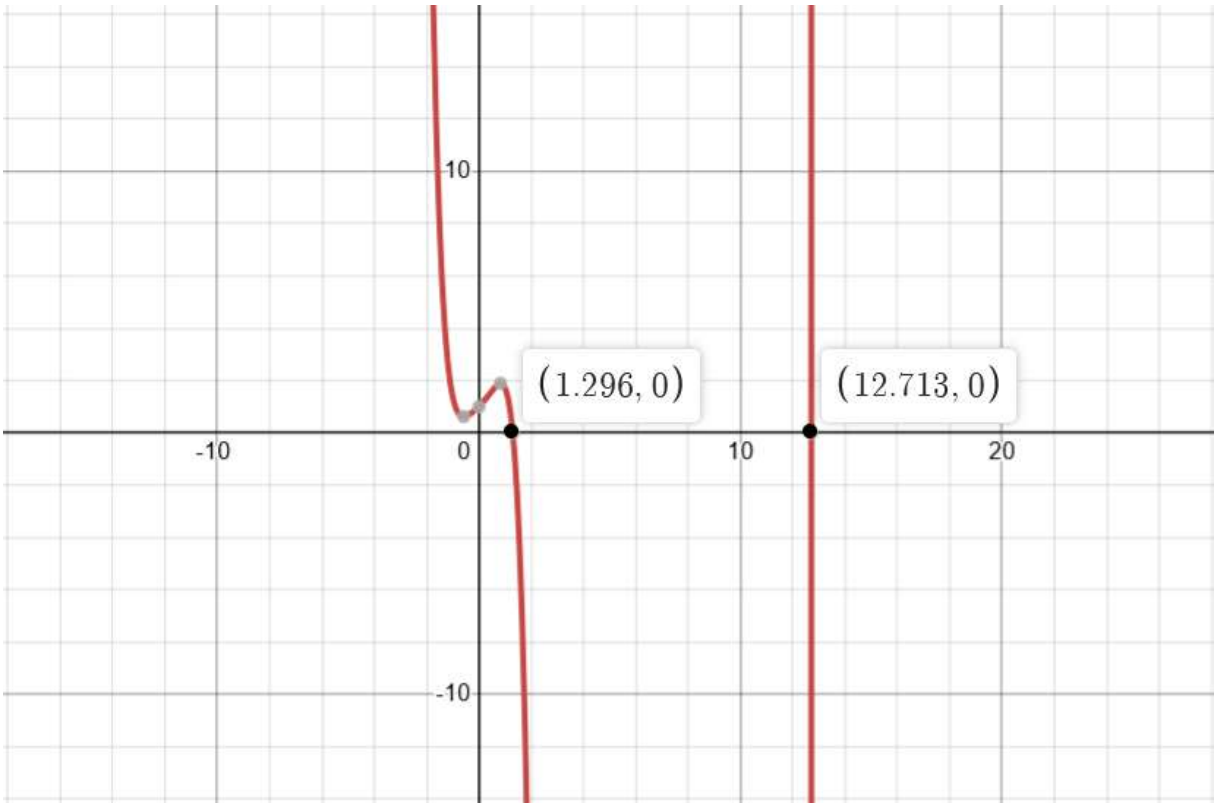

Fig 4 A frame of the plot of $y = e^x - x^5$

From *Fig 4* it is clear, that absolute value of denominator reaches its *infimum* at points $x_1 \cong 1.296$ and $x_2 \cong 12.713$, which are exactly the values at which maximum errors are reached.

Speaking of differences, the behavior of the $T(x)$ function is still noticeably different from the K functions. More specifically, it has a smaller value on the way to the first maximum, but then, before the second maximum, the error value suddenly drops by a factor of ten at $x \cong 10.83$. By plotting (*Fig 5*) a nominator of $T(x)$, we prove that this is exactly the value at which it reaches its *infimum*, causing the error value to drop.

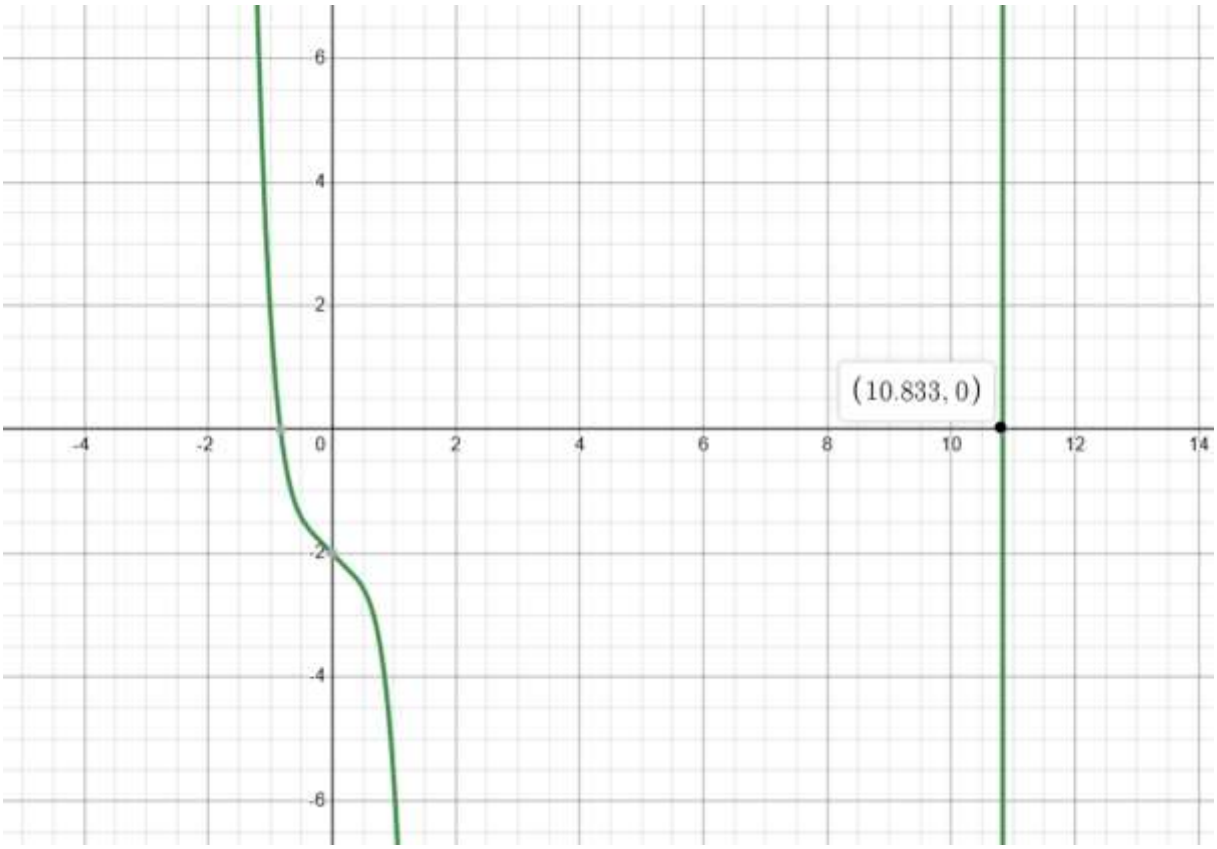

Fig 5 A frame of the plot of $y = xe^x - 2e^x - 3x^5$

At $x \gtrsim 22.3$, both K-functions plateau, maintaining an error value of ~ 4, unlike the T function, which increases linearly over this interval.

In conclusion, K_{A_1} is more efficient than K_{A_2} due to lower maximum error in $x \in (1.57, 11.5)$, but with exception of higher error at peak values. T function proved to have more unstable behaviour with an additional low-peak before and graduate increase after the second maximum.

References

[1] **Roman Z. Morawski**: *Numerical Methods, 2024L Lecture slides*

[2] ***Wikipedia*** *articles, namely*

- https://en.wikipedia.org/wiki/IEEE_754 (accessed 12.04.2024)
- https://en.wikipedia.org/wiki/Scientific_notation (accessed 15.04.2024)

[3] **R. Z. Morawski, A. Miękina**, *Solved Problems in Numerical Methods for Students of Electronics and Information Technology*, Oficyna Wydawnicza Politechniki Warszawskiej, 2021.