Warsaw University of Technology

FACULTY OF
ELECTRONICS AND INFORMATION TECHNOLOGY

Assignment A: Accuracy Of Computation

Mahmoud Elshekh Ali
Student index number: 323930

April 15, 2024

Course ENUME spring semester 2023/2024

Supervisor dr inż. Jakub Wagner

Contents

1 Mathematical Symbols And Notations

x	-Error free independent scalar variable.
$\mathbf{x}$	-Error free independent vector variable.
y	-Error free dependent scalar variable.
$\mathbf{y}$	-Error free dependent vector variable.
$\widetilde{x}$	-Error corrupted independent scalar variable.
$\widetilde{\mathbf{x}}$	-Error corrupted independent vector variable.
$\widetilde{y}$	-Error corrupted dependent scalar variable.
$\widetilde{\mathbf{y}}$	-Error corrupted dependent vector variable.
$\Delta\widetilde{x} = \widetilde{x} - x$	-Absolute error (in x).
$\delta[\widetilde{x}] = \frac{\widetilde{x}-x}{x}$	-Relative error (in x)
$T(x)$	-Coefficient of data error propogation
$K(x)$	-Coefficient of rounding error propogation
ϵ	-relative error in data.
η	-relative error due to rounding.
eps	- Maximum possible relative error due to rounding.

Model of error propogation for data errors:
$\delta[\widetilde{y}] = T(x) \cdot \delta[\widetilde{x}]$

Model of error propogation for rounding errors:
$\delta[\widetilde{y}] = K(x) \cdot \delta[\widetilde{x}]$

Worst-case data error:
$\delta[\widetilde{y}] \leq |T(x)| \cdot eps$

Worst-case rounding error:
$\delta[\widetilde{y}] \leq |K(x)| \cdot eps$

2 Introduction

- The aim of this assignment is to apply developed numerical methods of error prediction and analysis to various mathematical functions. The errors considered in this document are split into two categories: data errors and rounding errors. Data errors are those which are introduced with the data itself, due to for example rounding values of the data prior to using them; as a result of this the algorithm used to solve a particular mathematical expression does not effect the propogation of these errors, the propogation depends only on the expression itself. It is important that we are able to analyze these errors so that we can know the level of precision in data required given a certain application. Rounding errors on the other hand, are errors which are introduced during the intermediate steps of an algorithm. They strongly depend on the algorithm used. For this reason, the ability to analyze error in different algorithms is necessary to choose those which are the most accurate, and most useful to us. Errors may in general be expressed in either absolute or relative form, where absolute error is simply the difference between the estimated value and the "real", or error free value, and relative error is the ratio of the absolute error to the real value. The analysis of error (regardless of whether it is data or rounding error) is in this document done by analyzing and comparing two important parameters: The maximum possible or "worst-case" error , and the actual error. The analysis of worst-case error is convenient because it allows us to condense our analysis down to a single expression which we can for example use to directly compare two algorithms to observe the range of error we are exposed to when using each one. However, this parameter alone is not enough. More often than not there are fluctuations in errors, peaks and troughs, differences in error for different data ranges, and so on, which we are interested in observing, which we cannot derive solely from the worst-case error. Lastly, to wrap up this introduction, when we analyze error, we are generally less interested in the value of the error at a given point, but rather how it propogates through a given operation or set of operations. For this purpose, in order to have a concrete way of calculating and comparing errors, we must model this propogation. The method by which we choose to model the error propogation in this document is described in detail in the following page, alongside the methodology, assumptions and limitations under which this document was prepared.

3 Methodology and results of experimentation

3.1 Methodology, assumptions, limitations, etc.

As described in the introduction, we need to model the error propogation for both data and rounding error in order to effectively analyze and compare them. In order to get to that model however, we must first make a few assumptions to simplify our analysis. Firstly, we assume that the only source of error is rounding. From this assumption we know that the maximum possible relative error (also called *eps*) is $5 \cdot 10^{-L}$, where L is the so-called *mantissa* of the floating point representation; thus the actual relative error is in the interval $[-eps, eps]$. Furthermore, we may consider that the propogation of error through the set of operations we choose to perform is always linear, i.e., that the propogation of error through any of the operations can be described by either amplifying or attenuating the error by some coefficient (this only works if the magnitude of the relative error is very small, much smaller than 1). We call this coefficient $T(x)$ in the case of data error propogation, and $K(x)$ in the case of rounding error propogation. Now that we chose a model to describe the propogation of errors, we need to find the coefficients of error propogation. To do this, we have to methods:

- **Symbolic Differentiation:** This method requires the use of the equation $T(x) = \frac{x}{y}\frac{dy}{dx}$. This equation is derived by truncating the taylor series describing the error corrupted dependent variable (which leads to the result being an approximation rather than an exact result):

$$
\begin{aligned}
y = f(x) \rightarrow \widetilde{y} = f(\widetilde{x}) \Rightarrow y + \Delta\widetilde{y} &\equiv f(x + \widetilde{x}) \\
\widetilde{y} &= f(x) + f'(x)\Delta\widetilde{x} + \frac{1}{2}f''(x)(\Delta\widetilde{x})^2 + \dots \\
&= y + x\frac{\Delta\widetilde{x}}{x}f'(x) + \frac{1}{2}f''(x)x^2(\frac{\Delta\widetilde{x}}{x})^2 + \dots \\
\Delta\widetilde{y} = \widetilde{y} - y &= xf'(x)\epsilon + \frac{1}{2}x^2f''(x)(\epsilon)^2 + \dots, \text{where } \epsilon = \frac{\Delta\widetilde{x}}{x} \\
\delta[\widetilde{y}] \equiv= \frac{x}{y}f'(x)\epsilon + \frac{1}{2}\frac{x^2}{y}f''(x)(\epsilon)^2 + \dots &\rightarrow \delta[\widetilde{y}] \cong \frac{x}{y}f'(x)\epsilon \quad (1)
\end{aligned}
$$

- **Epsilon calculus:** This method uses approximations of simple operations to simplify large algebraic expressions. Before we get into the rules of epsilon calculus, we should write our error corrupted variable in the form that is appropriate for this method. We do this by rearranging the definition of relative error:
$\epsilon = \frac{\widetilde{x}-x}{x} = \frac{\widetilde{x}}{x} - 1 \rightarrow \widetilde{x} = x(1+\epsilon)$

  The rules of epsilon calculus:

  $(1+\epsilon_1)(1+\epsilon_2) \cong (1+\epsilon_1+\epsilon_2)$
  $(1+\epsilon)^a \cong 1+a\epsilon$
  $\log(1+\epsilon) \cong \epsilon$
  $e^{1+\epsilon} \cong (1+\epsilon)e$

Now that we have derived a working definition for $T(x)$, and knowing that the maximum relative error is *eps*, we can define an expression for the maximum relative data error:
$\delta[\widetilde{y}] \leq |T(x)| \cdot eps$

When it comes to rounding errors, we can find the coefficient of rounding error propogation using the same epsilon calculus procedure described previously. The expression for the maximum relative rounding error:
$\delta[\widetilde{y}] \leq |K(x)| \cdot eps$

3.2 Goals of the experiment

For the function $f(x) = y = \frac{arctan(x)}{x^2} - x^3$:

1. Find coefficient of error propogation $T(x)$.
2. Analyze propogation of data errors using computer generated data.
3. Find coefficients of rounding error propogation, $K_{A1}(x), K_{A2}(x)$, for algorithms $A1$, $A2$:

   A1: $\big[x\big] \rightarrow \begin{bmatrix} v_1 = arctan(x) \\ v_2 = x^2 \end{bmatrix} \rightarrow \begin{bmatrix} v_3 = \frac{v_1}{v_2} \\ v_4 = x^3 \end{bmatrix} \rightarrow \big[v_5 = v_3 - v_4\big] = \big[y\big]$

   A2: $\big[x\big] \rightarrow \begin{bmatrix} v_1 = arctan(x) \\ v_2 = x^5 \end{bmatrix} \rightarrow \begin{bmatrix} v_3 = v_1 - v_2 \\ v_4 = x^2 \end{bmatrix} \rightarrow \big[v_5 = \frac{v_3}{v_4}\big] = \big[y\big]$

4. Analyze propogation of rounding error for algorithms A1, A2 using computer generated data.

5. Compare all coefficients $T(x), K_{A1}(x), K_{A2}(x)$ and discuss differences between them.

experimentation process for numeric results

- The detailed *MATLAB* scripts used to achieve the experiment goals are provided in the appendix of this document, however a quick and general description of the process by which this experiment is performed is provided here. The experiment process is split into three steps :

- **Generation of data** is done by first choosing the size of the data vector, which in turn decides the spacing of the elements of the vector, then generating the vector using built-in functions in the *MATLAB* environment. The range of values considered is $x \in [10^{-2}, 10^{2}]$ across the entire document. The experiment is performed using linearly spaced data first, then logarithmically spaced data. This variety in data is used to allow for a more informed view of the error's behaviour. For the purposes of this experiment, we assume that the default "double" representation in *MATLAB* represents "true", or error free values (in reality, this is not the case), and that the "single" representation represents error corrupted values.

- **Performing operations on the data** is done by applying built-in and/or user-defined functions. We use the built-in *MATLAB* conversion function "*single*()", which converts data from the "double" to the "single" floating-point representation, to apply the effect of rounding, and introduce error into the data. The most important mathematical expression in the context of this document is the function

$$y = \frac{arctan(x)}{x^2} - x^3 \tag{2}$$

This is the expression that we use to apply and test all the methods derived and introduced hitherto. The purpose of this experiment is to analyze the propogation of data and rounding errors through equation 2. There are two algorithms, used for solving equation 2, which we use in this document to practice comparing the rounding error of multiple algorithms, called $A1$ and $A2$ respectively.

$$\text{A1: } \big[x\big] \rightarrow \begin{bmatrix} v_1 = arctan(x) \\ v_2 = x^2 \end{bmatrix} \rightarrow \begin{bmatrix} v_3 = \frac{v_1}{v_2} \\ v_4 = x^3 \end{bmatrix} \rightarrow \big[v_5 = v_3 - v_4\big] = \big[y\big]$$

$$\text{A2: } \big[x\big] \rightarrow \begin{bmatrix} v_1 = arctan(x) \\ v_2 = x^5 \end{bmatrix} \rightarrow \begin{bmatrix} v_3 = v_1 - v_2 \\ v_4 = x^2 \end{bmatrix} \rightarrow \big[v_5 = \frac{v_3}{v_4}\big] = \big[y\big]$$

- **Generation of figures** is done by using *MATLAB*'s extensive plotting capabilities to plot the data and the results of various operations. These plots are also regularly used to compare results and show trends in data in an informative and easy-to-understand way.

3.3 Results of experimentation

3.3.1 Finding $T(x)$

In this section, our aim is to find the coefficient of data error propogation $T(x)$. The following is the derivation of $T(x)$ using both symbolic differentiation and epsilon calculus:

1. **Using Symbolic Differentiation:**

$$\begin{aligned} T(x) = \frac{x}{y}\frac{dy}{dx} &\equiv \frac{x}{\frac{arctan(x)}{x^2} - x^3}(\frac{arctan(x)}{x^2} - x^3)' \\ &= \frac{x^3}{arctan(x) - x^5}(\frac{\frac{1}{1+x^2}x^2 - 2xarctan(x)}{x^4} - 3x^2) \\ &= \frac{1}{arctan(x) - x^5}(\frac{x}{1+x^2} - 2arctan(x) - 3x^5) \quad (3) \end{aligned}$$

2. **Using Epsilon Calculus:**

$$y = f(x) = \frac{arctan(x)}{x^2} - x^3 \rightarrow \widetilde{y} = \frac{arctan(x(1+\epsilon))}{(x(1+\epsilon))^2} - (x(1+\epsilon))^3 \rightarrow$$

$$\widetilde{y} = \frac{arctan(x(1+\epsilon))}{x^2(1+2\epsilon)} - x^3(1+3\epsilon) \rightarrow$$

using symbolic differentiation to find the propogation of error through $arctan(x)$:

$$T_{arctan(x)}(x) = \frac{x}{arctan(x)}(\frac{1}{1+x^2})$$

$$\widetilde{y} = \frac{arctan(x)(1+\frac{x}{arctan(x)}\frac{1}{1+x^2}\epsilon)}{x^2(1+2\epsilon)} - x^3(1+3\epsilon) \rightarrow$$

$$\widetilde{y} = \frac{arctan(x)}{x^2}(1+\frac{x}{arctan(x)}\frac{1}{1+x^2}\epsilon)(1-2\epsilon) - x^3(1+3\epsilon) \rightarrow$$

$$\widetilde{y} = \frac{arctan(x)}{x^2}(1+\frac{x}{arctan(x)}\frac{1}{1+x^2}\epsilon - 2\epsilon) - x^3(1+3\epsilon) \rightarrow$$

$$\widetilde{y} = \frac{arctan(x)}{x^2}(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 2)\epsilon) - x^3(1+3\epsilon) \rightarrow$$

$$\widetilde{y} = (y+x^3)(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 2)\epsilon) - x^3(1+3\epsilon) \rightarrow$$

$$\widetilde{y} = y(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2}-2)\epsilon)+x^3(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2}-2)\epsilon)-x^3(1+3\epsilon) \rightarrow$$

$$\widetilde{y} = y(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2}-2)\epsilon)+x^3((\frac{x}{arctan(x)}\frac{1}{1+x^2}-2)\epsilon-3\epsilon) \rightarrow$$

$$\widetilde{y} = y(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 2)\epsilon) + x^3(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 5)\epsilon \rightarrow$$

$$\widetilde{y} = y(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 2)\epsilon) + (\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3)\epsilon \rightarrow$$

$$\widetilde{y} = y(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 2)\epsilon + \frac{1}{y}(\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3)\epsilon) \rightarrow$$

$$\widetilde{y} = y(1+(\frac{x}{arctan(x)}\frac{1}{1+x^2} - 2 + \frac{1}{y}(\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3))\epsilon) \Rightarrow$$

$$T(x) = \frac{x}{arctan(x)}\frac{1}{1+x^2} - 2 + \frac{1}{y}(\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3)$$

$$= \frac{x}{arctan(x)}\frac{1}{1+x^2} - 2 + \frac{1}{\frac{arctan(x)}{x^2} - x^3}(\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3)$$

$$= \frac{x}{arctan(x)}\frac{1}{1+x^2} - 2 + \frac{1}{\frac{arctan(x)-x^5}{x^2}}(\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3)$$

$$
\begin{aligned}
&= \frac{x}{arctan(x)}\frac{1}{1+x^2} - 2 + \frac{x^2}{arctan(x)-x^5}(\frac{x^4}{arctan(x)}\frac{1}{1+x^2} - 5x^3) \\
&= \frac{x}{arctan(x)}\frac{1}{1+x^2} - 2 + \frac{x^6}{(arctan(x)-x^5)(arctan(x))(1+x^2)} - \frac{5x^5}{arctan(x)-x^5} \\
&= \frac{x}{arctan(x)}\frac{1}{1+x^2}(1+\frac{x^5}{arctan(x)-x^5}) - 2 - \frac{5x^5}{arctan(x)-x^5} \\
&= \frac{x}{arctan(x)}\frac{1}{1+x^2}(\frac{arctan(x)-x^5+x^5}{arctan(x)-x^5}) - 2 - \frac{5x^5}{arctan(x)-x^5} \\
&= \frac{x}{arctan(x)}\frac{1}{1+x^2}(\frac{arctan(x)}{arctan(x)-x^5}) - 2 - \frac{5x^5}{arctan(x)-x^5} \\
&= \frac{x}{(1+x^2)(arctan(x)-x^5)} - 2 - \frac{5x^5}{arctan(x)-x^5} \\
&= \frac{1}{arctan(x)-x^5}(\frac{x}{1+x^2} - 2(arctan(x)-x^5) - 5x^5) \\
&= \frac{1}{arctan(x)-x^5}(\frac{x}{1+x^2} - 2arctan(x) + 2x^5 - 5x^5) \\
&= \frac{1}{arctan(x)-x^5}(\frac{x}{1+x^2} - 2arctan(x) - 3x^5)
\end{aligned} \tag{4}
$$

The fact that the two results are equivalent is not a coincidence. It is infact a good confirmation of our results as both methods always return equivalent expressions. Finally, we have the following expression for $T(x)$, which we will be using to approximate the propogation of relative data errors in equation 2:

$$
T(x) = \frac{1}{arctan(x)-x^5}(\frac{x}{1+x^2} - 2arctan(x) - 3x^5) \tag{5}
$$

$$
\delta[\widetilde{y}] \cong T(x) \cdot \delta[\widetilde{x}] \tag{6}
$$

3.3.2 Analyzing data error using $T(x)$ and computer generated data

Using the *MATLAB* environment, we generate 2 vectors of data, $\mathbf{x}_1, \mathbf{x}_2$, containing 5000 elements in the range of $[10^{-2}, 10^2]$ each. $\mathbf{x}_1$ being of data that is linearly spaced, $\mathbf{x}_2$ being of data that is logarithmically spaced. We save 2 versions of each vector, one that is saved in the "double" representation (representing error-free data $\mathbf{x}_1$, $\mathbf{x}_2$), and one that is in the "single" representation (representing error-corrupted data $\widetilde{\mathbf{x}_1}, \widetilde{\mathbf{x}_2}$). We then

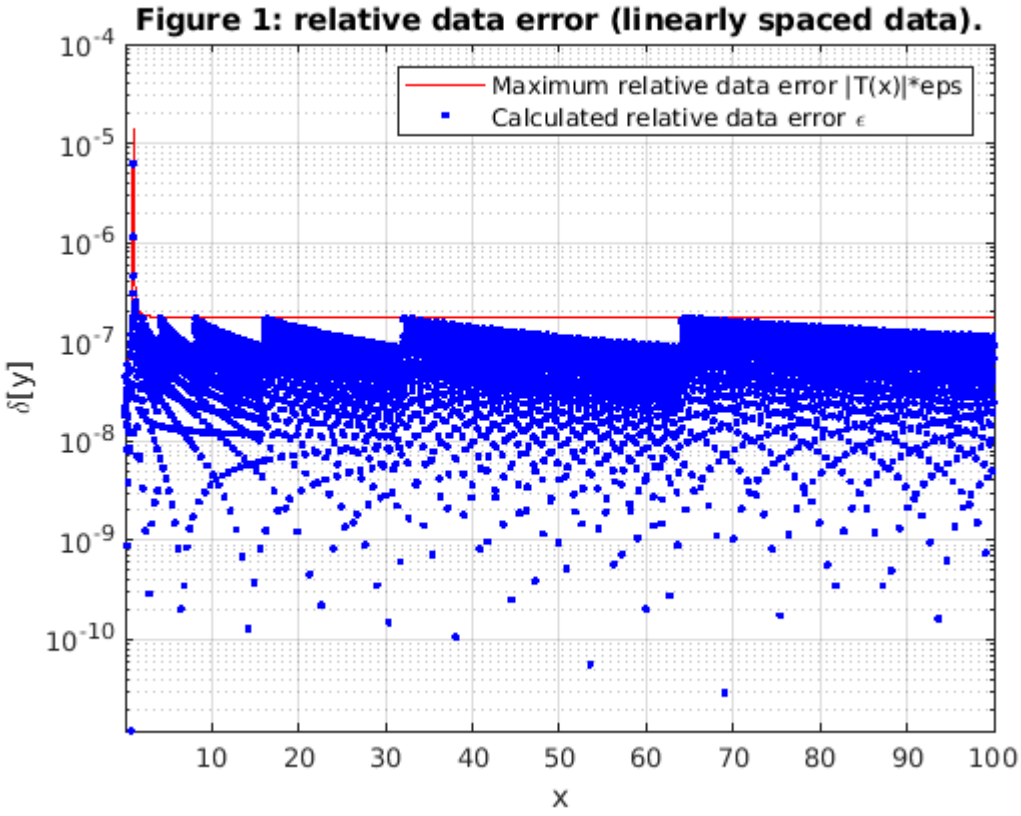

Figure 1: maximum relative data error plotted alongside calculated relative data errors in result of computation for linearly spaced data.

apply equation 2 to both vectors of data. We can calculate the relative error in the result using the definition of relative error $\delta[\widetilde{\mathbf{y}}] = \frac{\widetilde{\mathbf{y}}-\mathbf{y}}{\mathbf{y}}$. Figures 1, 2 illustrate our results.

3.3.3 Finding K_{A1}, K_{A2}

To find the coefficient charactarizing the propogation of rounding error for algorithms $A1$, $A2$, we can use epsilon calculus similarly to how we used it to find $T(x)$:

$$\text{A1: } [x] \to \begin{bmatrix} v_1 = arctan(x) \\ v_2 = x^2 \end{bmatrix} \to \begin{bmatrix} v_3 = \frac{v_1}{v_2} \\ v_4 = x^3 \end{bmatrix} \to [v_5 = v_3 - v_4] = [y]$$

$$\text{A2: } [x] \to \begin{bmatrix} v_1 = arctan(x) \\ v_2 = x^5 \end{bmatrix} \to \begin{bmatrix} v_3 = v_1 - v_2 \\ v_4 = x^2 \end{bmatrix} \to [v_5 = \frac{v_3}{v_4}] = [y]$$

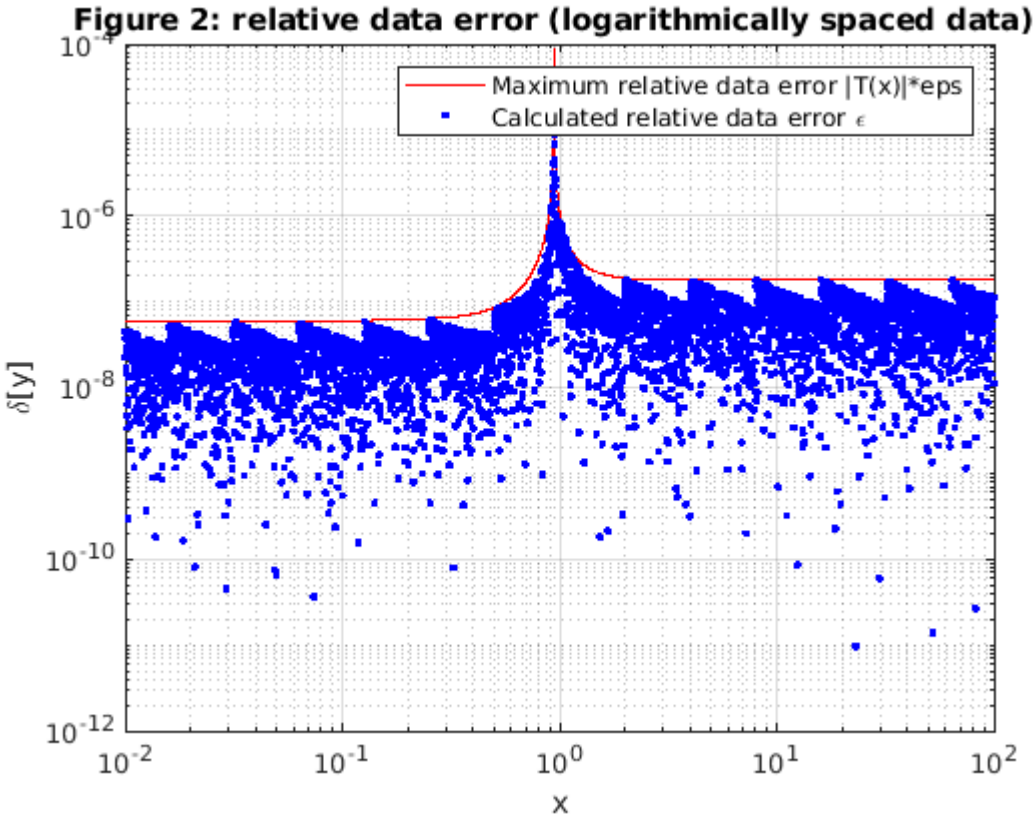

Figure 2: maximum relative data error plotted alongside calculated relative data errors in result of computation for logarithmically spaced data.

$$A1 : \widetilde{y} = (\frac{arctan(x)(1+\eta_1)}{x^2(1+\eta_2)}(1+\eta_3) - x^3(1+\eta_4))(1+\eta_5) \rightarrow$$

$$\widetilde{y} = (\frac{arctan(x)}{x^2}(1+\eta_1-\eta_2+\eta_3) - x^3(1+\eta_4))(1+\eta_5) \rightarrow$$

$$\widetilde{y} = \frac{arctan(x)}{x^2}(1+\eta_1-\eta_2+\eta_3+\eta_5) - x^3(1+\eta_4+\eta_5) \rightarrow$$

$$\widetilde{y} = (y+x^3)(1+\eta_1-\eta_2+\eta_3+\eta_5) - x^3(1+\eta_4+\eta_5) \rightarrow$$

$$\widetilde{y} = y(1+\eta_1-\eta_2+\eta_3+\eta_5) + x^3(1+\eta_1-\eta_2+\eta_3+\eta_5) - x^3(1+\eta_4+\eta_5) \rightarrow$$

$$\widetilde{y} = y(1+\eta_1-\eta_2+\eta_3+\eta_5) + x^3(\eta_1-\eta_2+\eta_3-\eta_4) \rightarrow$$

$$\widetilde{y} = y(1+\eta_1-\eta_2+\eta_3+\eta_5+\frac{x^3}{y}(\eta_1-\eta_2+\eta_3-\eta_4)) \rightarrow$$

$$\widetilde{y} = y(1+(1+\frac{x^3}{y})\eta_1+(-1-\frac{x^3}{y})\eta_2+(1+\frac{x^3}{y})\eta_3+(-\frac{x^3}{y})\eta_4+\eta_5)$$

$$\delta[\widetilde{y}] = (1+\frac{x^3}{y})\eta_1 + (-1-\frac{x^3}{y})\eta_2 + (1+\frac{x^3}{y})\eta_3 + (-\frac{x^3}{y})\eta_4 + \eta_5 \leq$$
$$|1+\frac{x^3}{y}|eps+|1+\frac{x^3}{y}|eps+|1+\frac{x^3}{y}|eps+|\frac{x^3}{y}|eps+eps = \underbrace{(3\cdot|1+\frac{x^3}{y}|+|\frac{x^3}{y}|+1)}_{K_{A1}}eps$$
$$A2: \delta[\widetilde{y}] = \frac{(arctan(x)(1+\eta_1)-x^5(1+\eta_2))(1+\eta_3)}{x^2(1+\eta_4)}(1+\eta_5) \rightarrow$$
$$\widetilde{y} = \frac{arctan(x)(1+\eta_1)-x^5(1+\eta_2)}{x^2(1+\eta_4)}(1+\eta_3+\eta_5) \rightarrow$$
$$\widetilde{y} = \frac{arctan(x)(1+\eta_1)-x^5(1+\eta_2)}{x^2}(1+\eta_3-\eta_4+\eta_5) \rightarrow$$
$$\widetilde{y} = \frac{arctan(x)(1+\eta_1+\eta_3-\eta_4+\eta_5)-x^5(1+\eta_2+\eta_3-\eta_4+\eta_5)}{x^2} \rightarrow$$
$$\widetilde{y} = \frac{arctan(x)-x^5+arctan(x)(\eta_1+\eta_3-\eta_4+\eta_5)-x^5(\eta_2+\eta_3-\eta_4+\eta_5)}{x^2} \rightarrow$$
$$\widetilde{y} = \frac{arctan(x)-x^5}{x^2}+\frac{arctan(x)(\eta_1+\eta_3-\eta_4+\eta_5)-x^5(\eta_2+\eta_3-\eta_4+\eta_5)}{x^2} \rightarrow$$
$$\widetilde{y} = y+\frac{arctan(x)(\eta_1+\eta_3-\eta_4+\eta_5)-x^5(\eta_2+\eta_3-\eta_4+\eta_5)}{x^2} \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)(\eta_1+\eta_3-\eta_4+\eta_5)-x^5(\eta_2+\eta_3-\eta_4+\eta_5)}{x^2y}) \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)(\eta_1+\eta_3-\eta_4+\eta_5)}{x^2y}-\frac{x^3(\eta_2+\eta_3-\eta_4+\eta_5)}{y}) \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)}{x^2y}\eta_1+(\frac{-x^3}{y})\eta_2+(\frac{arctan(x)}{x^2y}-\frac{x^3}{y})(\eta_3-\eta_4+\eta_5)) \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)}{x^2y}\eta_1+(\frac{-x^3}{y})\eta_2+\frac{1}{y}(\frac{arctan(x)}{x^2}-x^3)(\eta_3-\eta_4+\eta_5)) \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)}{x^2y}\eta_1+(\frac{-x^3}{y})\eta_2+\frac{1}{y}(\frac{arctan(x)-x^5}{x^2})(\eta_3-\eta_4+\eta_5)) \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)}{x^2y}\eta_1+(\frac{-x^3}{y})\eta_2+\frac{1}{y}y(\eta_3-\eta_4+\eta_5)) \rightarrow$$
$$\widetilde{y} = y(1+\frac{arctan(x)}{x^2y}\eta_1+(\frac{-x^3}{y})\eta_2+\eta_3-\eta_4+\eta_5)$$
$$\delta[\widetilde{y}] = \frac{arctan(x)}{x^2y}\eta_1+(\frac{-x^3}{y})\eta_2+\eta_3-\eta_4+\eta_5 \leq$$
$$|\frac{arctan(x)}{x^2y}|eps+|\frac{x^3}{y}|eps+eps+eps+eps = \underbrace{(|\frac{arctan(x)}{x^2y}|+|\frac{x^3}{y}|+3)}_{K_{A2}}eps \quad (7)$$

Finally, we have the following expression for K_{A1}, K_{A2}, which we will be using to approximate the propogation of rounding data errors in equation 2 for algorithm $A1$, $A2$ repectively:

$$K_{A1} = 3 \cdot |1 + \frac{x^3}{y}| + |\frac{x^3}{y}| + 1 \tag{8}$$

$$K_{A2} = |\frac{arctan(x)}{x^2 y}| + |\frac{x^3}{y}| + 3 \tag{9}$$

3.3.4 Analyzing rounding error using K_{A1}, K_{A2} and computer generated data

Using the same vectors of data and procedure as section 3.3.2, we obtain figures 3,4,5,6.

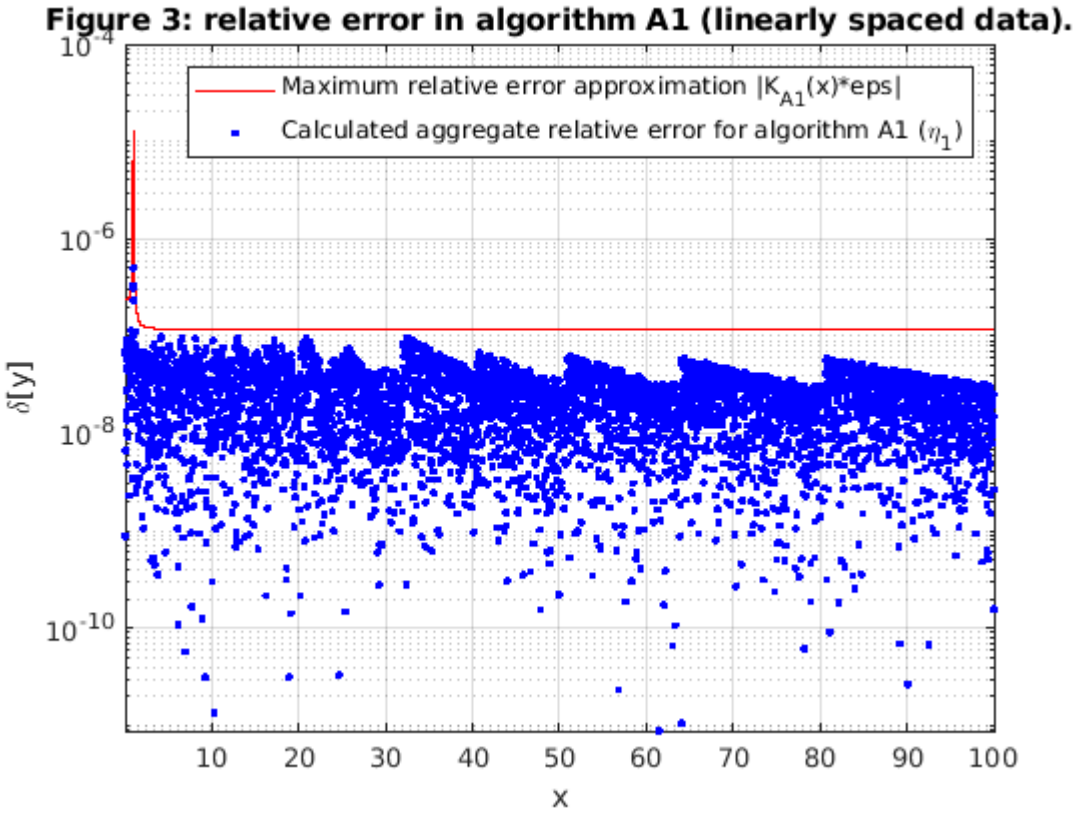

Figure 3: maximum relative rounding error plotted alongside calculated relative rounding errors in result of computation for algorithm $A1$, using linearly spaced data.

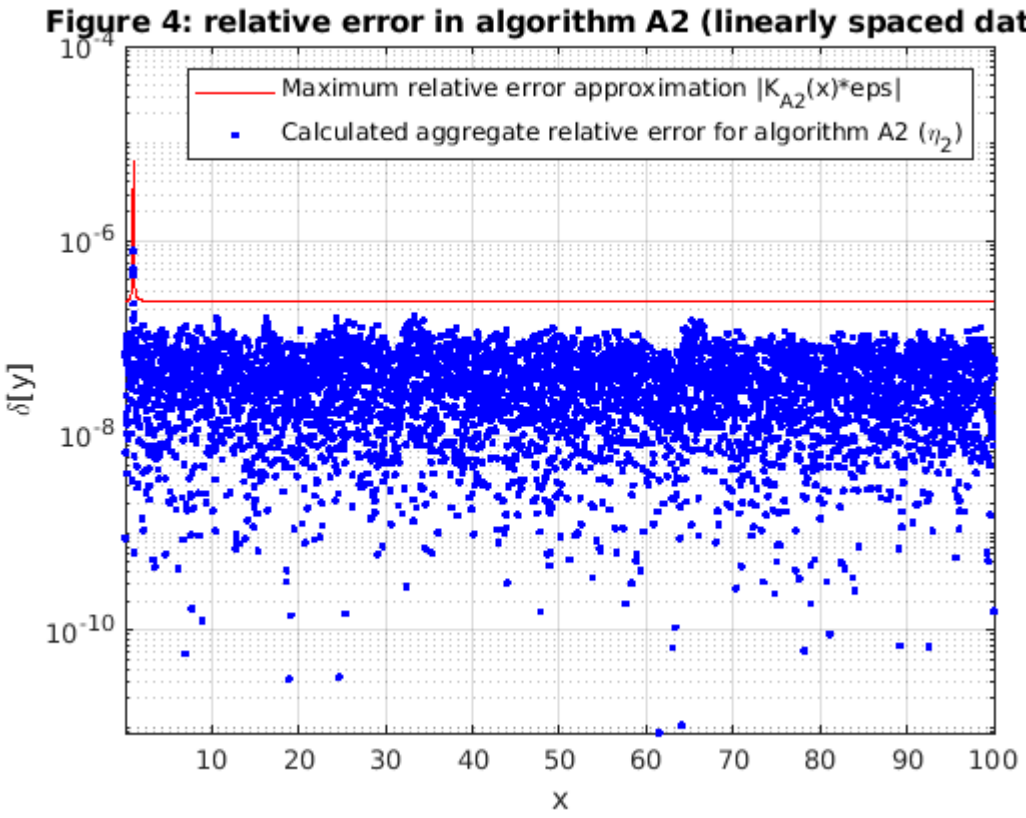

Figure 4: maximum relative rounding error plotted alongside calculated relative rounding errors in result of computation for algorithm $A2$, using linearly spaced data.

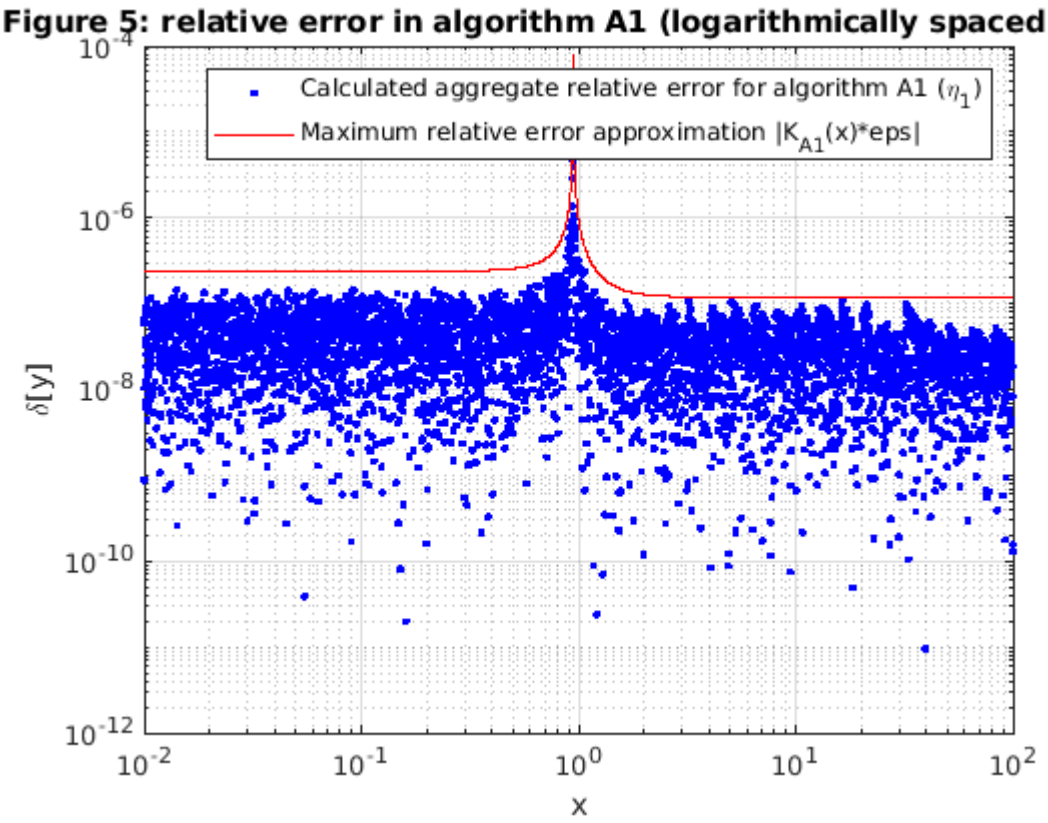

Figure 5: maximum relative rounding error plotted alongside calculated relative rounding errors in result of computation for algorithm $A1$, using logarithmically spaced data.

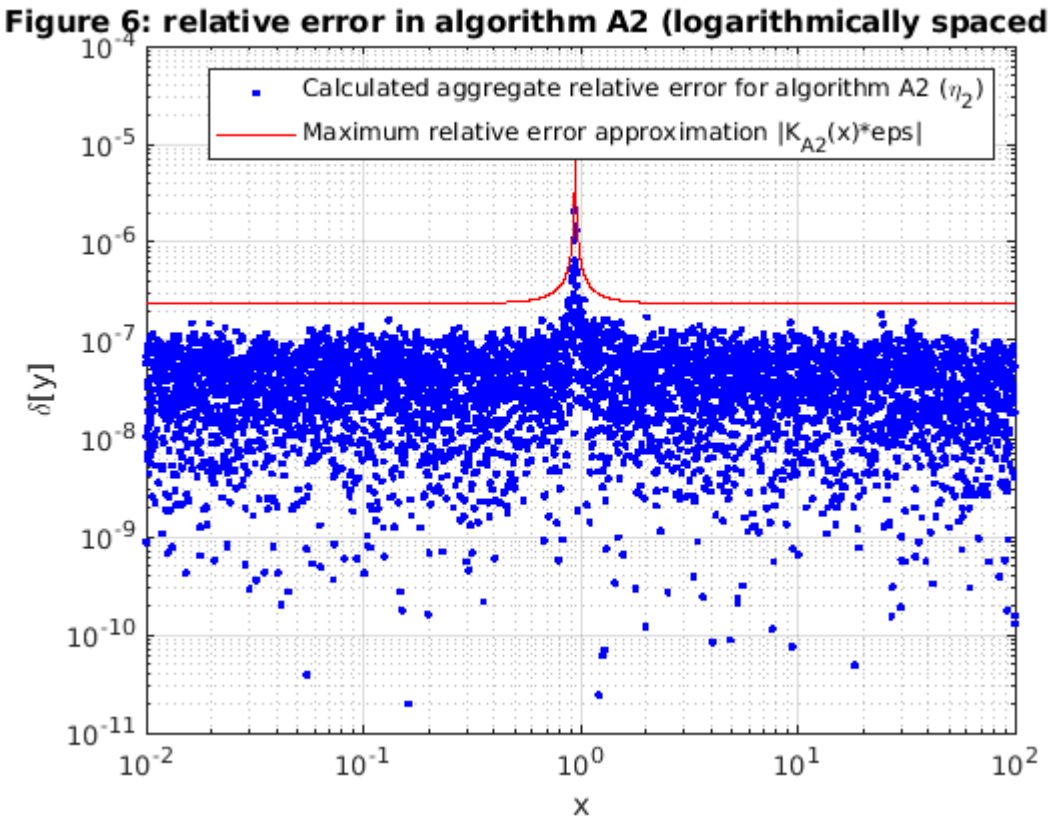

Figure 6: maximum relative rounding error plotted alongside calculated relative rounding errors in result of computation for algorithm $A2$, using logarithmically spaced data.

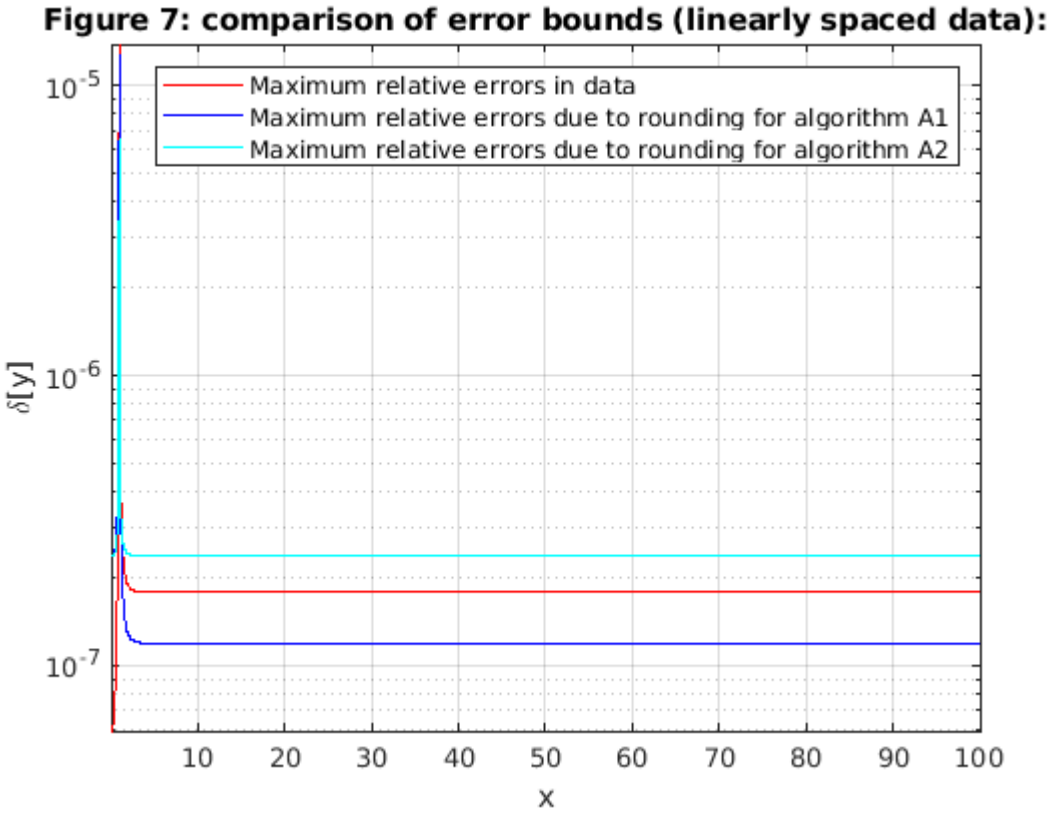

Figure 7: Comparison of maximum error bounds for data and rounding errors for linearly spaced data.

3.3.5 Comparison of error bounds for data and rounding error

Using the same vectors of data and procedure as section 3.3.2, we may plot all the expressions of maximum error hitherto obtained in a single figure to directly compare the bounds for each type of error across the range of data considered.

4 Discussion

- Let us begin our discussion of the results obtained by examining figures 1, 2. Perhaps the most striking aspects of these figures are the sawtooth-like pattern with which the relative error repeats, and the peak observed at $x \approx 0.946$. The peak can be explained by looking at equation 5,

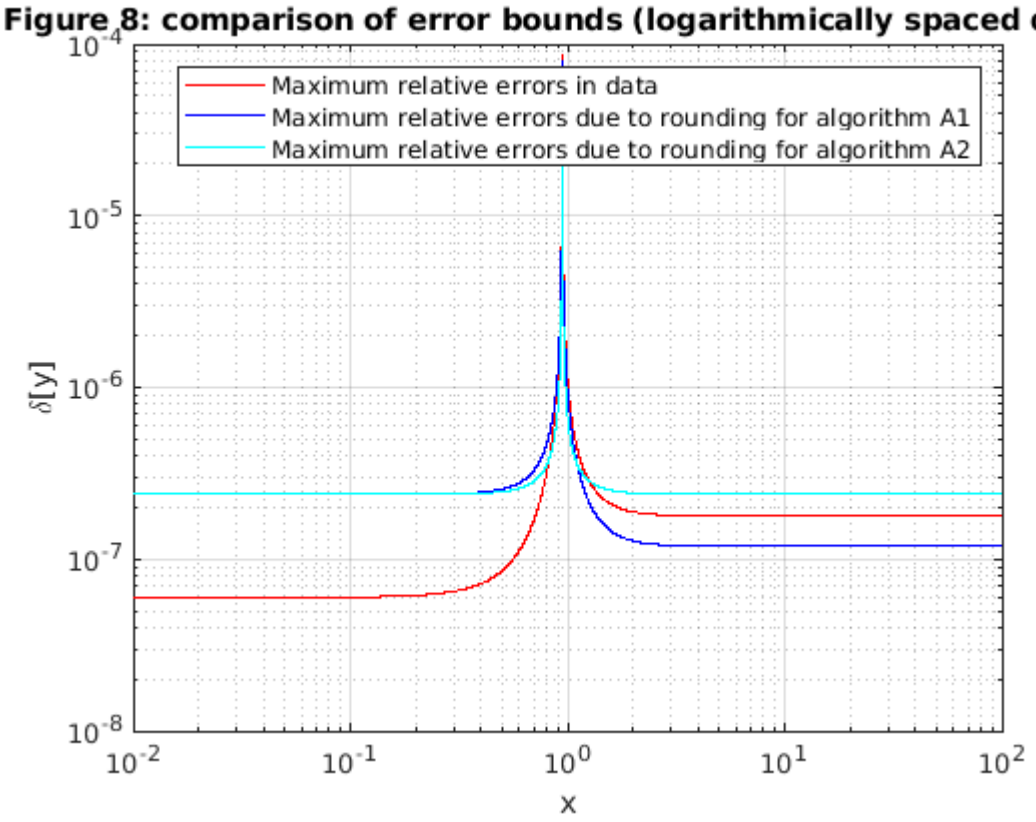

Figure 8: Comparison of maximum error bounds for data and rounding errors for logarithmically spaced data.

$T(x) \to \infty$ as $x \to 0.946$ due to the fraction $\frac{1}{arctan(x)-x^5}$. This is because $\frac{1}{arctan(x)-x^5} \to 0$ as $x \to 0.946$. As for the sawtooth-like pattern in the relative error, seen not just in figure 1 and 2 but across most of the figures in this document, a likely explanation is the effect of rounding the data, mixed with taking the absolute value to inspect the magnitude of the error, and the fact that most of the mathematical expressions mentioned hitherto, including the ones used in figures 1,2, are for the most part monotonic. Rounding the data would lead to a cyclic pattern with a peak when the error is equal to *eps*. Other noteworthy features of the figures are, for example, the fact that the maximum error is higher for values higher than $x \approx 0.946$ than for values lower than this value. Also, we can observe from the difference in the figures that the periodic nature of the sawtooth-like pattern in the error is much more consistent and pronounced for logarithmically spaced data values as compared to linearly spaced data.

- Moving on to the analysis of figures 3-6, we can make very similiar observations to those made for figures 1,2. There are peaks for the errors at the same point as for data error, $x \approx 0.946$. Furthermore, the same sawtooth-like pattern can be observed for the rounding error in algorithm $A1$, however it is much less pronounced, and it is not apparent at all for the rounding error in algorithm $A2$. Additionally, in the algorithm $A1$, the maximum rounding error is higher for values lower than $x \approx 0.946$ than it is for values higher than this value. In the algorithm $A2$, however, the maximum rounding error remains at the same level ouside the peak at $x \approx 0.946$.

- Finally, we can close the discussion of obtained results with a direct comparison of all the error bounds, using figures 7,8. As can be observed in figure 8, the data error is significantly lower than the rounding error for values below $x \approx 0.946$. The accuracy in both algorithms for this range is for the most part equivalent. the behaviour of all sources of error at and within a small window around $x \approx 0.946$ is largely the same, as are the magnitudes of error for all sources in this interval. The most interesting behaviour happens for values larger than $x \approx 0.946$. The rounding error algorithm $A2$ is The largest among all sources of error at this range, followed by the data errors, and finally the error in algorithm $A1$. We can deduce from this than the preffered algorithm to use for solving equation 2 is algorithm $A1$, due to the rounding error caused by it being overall lower than the rounding error caused by algorithm $A2$.

5 Conclusion

Our analysis of the obtained results leaves us with the following:

- For equation 2, the error in data follows a predictable pattern which we can leverage for error prediction.
- Algorithm $A1$ is in general more accurate than algorithm $A2$.
- For values of relative error which are very small in magnitude, the linear model of error propogation is a good approximation of actual error propogation.

6 References

[1] R. Z. Morawski, lecture notes for the course *Numerical Methods*, Warsaw University of Technology, Faculty of Electronics and Information Technology, spring semester 2023/24.

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

[3] J. Wagner, assignment guides for the course Numerical Methods, Warsaw University of Technology, Faculty of Electronics and Information Technology, spring semester 2023/24.

7 Appendix

7.1 *MATLAB* implementation of task 2 (Task2.m)

```
% TASK 2
% Start of script. clear any remaining variables
   from previous script runs.
clear

% Number of samples used for testing (number of
   elements of the data vector).
N =5000
% initializing vectors of data:
% x:  vector of linearly spaced data values.
```

```
9  % xl: vector of logarithmically spaced data values.
10 x = linspace(10^-2, 10^2, N);
11 xl = logspace(-2, 2, N);
12
13 % initialization of error corrupted data vectors.
       Data corruption is done
14 % by converting from "double" representaion (default
       in MATLAB), to "single" representation.
15 x_err = double(single(x));
16 xl_err = double(single(xl));
17
18 % Defining functions f(x) and T(x). f(x) is the
       function given originally in the
19 % assignment A description. It is the function for
       which we are interested
20 % in finding the coefficient of data error
       propogation T(x). T(X) was
21 % derived in task 1 of assignment A.
22 f = @(x) (atan(x)./x.^2) - x.^3;
23 T = @(x) (x./(1+x.^2) -2*atan(x) -3*x.^5)./(atan(x)-
       x.^5);
24
25 % Calculating the vector of non-error-corrupted
       values of the
26 % results of f(x) (here named y), as well as the
       vector of error-corrupted
27 % values of f(x) (here named y_err). The conversion
       of y_err to the "double"
28 % representation is done so both variables are of
       the same type (it is not
29 % necessary but is rather done for convenience. It
       does not reverse the effect of
30 % converting to the "single" representation). All
       the operations mentioned
31 % are here done for the vector of linearly spaced
       data values x.
32 y = f(x);
33 y_err = f(x_err);
34
35 % The same set of operations as the last step, but
       here done for the vector
```

```
% of logarithmically spaced data values xl.
yl = f(xl);
yl_err = f(xl_err);

% Calulating the relative error in the result using the definition for
% relative error. dy is the error in y_err, dyl is the error in yl_err. The
% reason for taking the absolute value of the error is because we are
% interested only in the error magnitude.
dy = abs((y_err-y)./y);
dyl = abs((yl_err-yl)./yl);

% Calculating the maximum relative error due to rounding (the division by
% two here is becuase the built-in eps returns the minimum space between
% two consecutive numbers, which is twice the maximum error due to rounding.
epssin = eps("single")/2;

% Calculating the vector containing N samples of the function describing
% the maxinum possible relative error approximation. t is the vector of samples
% given the linearly spaced data values, tl is the vector of samples given the
% logarithmically spaced data values.
t = abs(T(x))*epssin;
tl = abs(T(xl))*epssin;

% Code for the generation of figure 1.
figure(1);
plot(x,t,"r-");
hold on;grid on
axis([0.01,100,-inf,1e-4])
plot(x,dy,"b.");
xlabel('x');
ylabel('\delta[y]');
```

```
67 legend("Maximum relative data error |T(x)|*eps","
      Calculated relative data error \epsilon")
68 title("Figure 1: relative data error given linearly
      spaced data values.")
69 yscale log;hold off;
70
71 % Code for the generation of figure 2.
72 figure(2);
73 loglog(xl, tl, 'r-');
74 hold on;grid on;
75 loglog(xl, dyl, ".b");
76 xlabel('x');
77 ylabel('\delta[y]');
78 legend("Maximum relative data error |T(x)|*eps","
      Calculated relative data error \epsilon")
79 title("Figure 2: relative data error given
      logarithmically spaced data values.")
80 hold off;
```

7.2 *MATLAB* implementation of task 4 (Task4.m)

```
1  % TASK 4
2  % Start of script. clear any remaining variables
      from previous script runs.
3  clear
4
5  % Number of samples used for testing (number of
      elements of the data vector).
6  N =5000
7  % initializing vectors of data:
8  % x:  vector of linearly spaced data values.
9  % xl: vector of logarithmically spaced data values.
10 x = linspace(10^-2, 10^2, N);
11 xl = logspace(-2, 2, N);
12
13 % Defining function f(x) originally given in the
      assignement A description,
14 % as well as the coefficients describing the
      propogation of rounding errors
```

```
15 % in algorithm A1 (KA1) and in algorithm 2 (KA2). Both of these
16 % coefficients were calculated in Task 3.
17 f = @(x) (atan(x)./x.^2) - x.^3;
18 KA1 = @(x) 3*abs(1+x.^3./f(x))+abs(x.^3./f(x))+1;
19 KA2 = @(x) 3+abs(x.^3./f(x))+abs(atan(x)./(x.^2.*f(x)));
20
21 % Calculating the vector of non-error-corrupted values of the
22 % results of f(x) (here named y). It is here done for linearly
23 % spaced values of the data.
24 y = f(x);
25
26 % The same operation as the last step, but here done for the vector
27 % of logarithmically spaced data values xl.
28 yl = f(xl);
29
30 % The steps of the algorith A1 performed one by one. The conversion to the
31 % 'single' representation is here used to round the values and introduce
32 % the rounding error. The rounding is done after each step. After rounding
33 % we convert back to double representation to allow for further operations.
34 % Here the algorithm is applied to linearly spaced values of the data. The result is
35 % stored in y_err1.
36 v1_1 = double(single(   atan(x)  ));
37 v2_1 = double(single(     x.^2   ));
38 v3_1 = double(single( v1_1./v2_1 ));
39 v4_1 = double(single(    x.^3    ));
40 v5_1 = double(single( v3_1-v4_1  ));
41 y_err1 = v5_1;
42
43 % The steps of the algorith A2 performed one by one. Here the algorithm
```

```
44 % is applied to linearly spaced values of the data.
     The result is
45 % stored in y_err2.
46 v1_2 = double(single(   atan(x)  ));
47 v2_2 = double(single(    x.^5    ));
48 v3_2 = double(single(  v1_2-v2_2 ));
49 v4_2 = double(single(    x.^2    ));
50 v5_2 = double(single( v3_2./v4_2 ));
51 y_err2 = v5_2;
52
53 % The steps of the algorith A1 performed one by one.
     Here the algorithm
54 % is applied to logarithmically spaced values of the
     data. The result is
55 % stored in yl_err1.
56 vl1_1 = double(single(   atan(xl)   ));
57 vl2_1 = double(single(     xl.^2    ));
58 vl3_1 = double(single( vl1_1./vl2_1 ));
59 vl4_1 = double(single(    xl.^3     ));
60 vl5_1 = double(single( vl3_1-vl4_1  ));
61 yl_err1 = vl5_1;
62
63 % The steps of the algorith A2 performed one by one.
     Here the algorithm
64 % is applied to logarithmically spaced values of the
     data. The result is
65 % stored in yl_err2.
66 vl1_2 = double(single(   atan(xl)   ));
67 vl2_2 = double(single(    xl.^5     ));
68 vl3_2 = double(single(  vl1_2-vl2_2 ));
69 vl4_2 = double(single(    xl.^2     ));
70 vl5_2 = double(single( vl3_2./vl4_2 ));
71 yl_err2 = vl5_2;
72
73 % Calulating the relative error in the result using
     the definition for
74 % relative error. dy1 is the error in y1_err1, dyl1
     is the error in yl_err1,
75 % dy2 is the error in y_err2, dyl2 is the error in
     yl_err2. The
```

```
% reason for taking the absolute value of the error is because we are
% interested only in the error magnitude.
dy1 = abs((y_err1-y)./y);
dy2 = abs((y_err2-y)./y);
dyl1 = abs((yl_err1-yl)./yl);
dyl2 = abs((yl_err2-yl)./yl);

% Calculating twice the maximum relative error due to rounding.
epssin = eps("single")/2;

% Code for the generation of figure 3.
figure(3);
plot(x,KA1(x)*epssin,"r-")
hold on;grid on;
axis([0.01,100,-inf,1e-4])
plot(x,dy1,"b.");
xlabel('x');
ylabel('\delta[y]');
legend("Maximum relative error approximation |K_{A1}(x)*eps|","Calculated aggregate relative error for algorithm A1 (\eta_{1})")
title("Figure 3: relative error in results of algorithm A1 given linearly spaced data values.")
yscale log;hold off;

% Code for the generation of figure 4.
figure(4);
plot(x,KA2(x)*epssin,"r-")
hold on;grid on;
axis([0.01,100,-inf,1e-4])
plot(x,dy2,"b.");
xlabel('x');
ylabel('\delta[y]');
legend("Maximum relative error approximation |K_{A2}(x)*eps|","Calculated aggregate relative error for algorithm A2 (\eta_{2})")
title("Figure 4: relative error in results of algorithm A2 given linearly spaced data values.")
yscale log;hold off;
```

```
109
110 % Code for the generation of figure 5.
111 figure(5);
112 loglog(xl, dyl1, 'b.');
113 hold on;grid on;
114 loglog(xl,KA1(xl)*epssin,"r-")
115 xlabel('x');
116 ylabel('\delta[y]');
117 legend("Calculated aggregate relative error for
       algorithm A1 (\eta_{1})","Maximum relative error
       approximation |K_{A1}(x)*eps|")
118 title("Figure 5: relative error in results of
       algorithm A1 given logarithmically spaced data
       values.")
119 hold off;
120
121 % Code for the generation of figure 6.
122 figure(6);
123 loglog(xl, dyl2, 'b.');
124 hold on;grid on;
125 loglog(xl,KA2(xl)*epssin,"r-")
126 xlabel('x');
127 ylabel('\delta[y]');
128 legend("Calculated aggregate relative error for
       algorithm A2 (\eta_{2})","Maximum relative error
       approximation |K_{A2}(x)*eps|")
129 title("Figure 6: relative error in results of
       algorithm A2 given logarithmically spaced data
       values.")
130 hold off;
```

7.3 *MATLAB* implementation of task 5 (Task5.m)

```
1 % Task 5
2 % Number of samples used for testing (number of
       elements of the data vector).
3 N =5000
4 % initializing vectors of data:
5 % x:  vector of linearly spaced data values.
6 % xl: vector of logarithmically spaced data values.
```

```
7  x = linspace(10^-2, 10^2, N);
8  xl = logspace(-2, 2, N);
9
10 % Calculating the maximum relative error due to
       rounding (the division by
11 % two here is becuase the built-in eps returns the
       minimum space between
12 % two consecutive numbers, which is twice the
       maximum error due to rounding.
13 epssin = eps("single")/2;
14
15 % Definition for coefficient of data error
       propogation derived in Task 1,
16 % coefficients of rounding error propogation in
       algorithm A1 (KA1) and in
17 % algorithm A2 (KA2) derived in Task 3.
18 f = @(x) (atan(x)./x.^2) - x.^3;
19 T = @(x) ((x./(1+x.^2) -2*atan(x) -3*x.^5)./(atan(x)
       -x.^5) );
20 KA1 = @(x) 3*abs(1+x.^3./f(x))+abs(x.^3./f(x))+1;
21 KA2 = @(x) 3+abs(x.^3./f(x))+abs(atan(x)./(x.^2.*f(x
       )));
22
23 % Code for generation of figure 7
24 figure(7)
25 plot(x,abs(T(x)*epssin),"r-")
26 hold on;grid on;
27 axis([0.01,100,-inf,inf])
28 plot(x,(KA1(x)*epssin),"b-")
29 plot(x,(KA2(x)*epssin),"c-")
30 xlabel('x');
31 ylabel('\delta[y]');
32 title("Figure 7: comparison of error bounds for data
       and rounding errors (linearly spaced data):")
33 legend("Maximum relative errors in data","Maximum
       relative errors due to rounding for algorithm A1
       ","Maximum relative errors due to rounding for
       algorithm A2")
34 yscale log;hold off;
35
36 % Code for generation of figure 8
```

```
figure(8);
loglog(xl, abs(T(xl)*epssin), 'r-');
hold on;grid on;
loglog(xl, (KA1(xl)*epssin), 'b-');
loglog(xl, (KA2(xl)*epssin), 'c-');
xlabel('x');
ylabel('\delta[y]');
title("Figure 8: comparison of error bounds for data
    and rounding errors (logarithmically spaced data
   ):")
legend("Maximum relative errors in data","Maximum
   relative errors due to rounding for algorithm A1
   ","Maximum relative errors due to rounding for
   algorithm A2")
hold off;
```