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

Table of Contents

I. Notation……………………........................................................................................................................2

II. Theoretical introduction......................................................................................................................3

III. Formulation of problem..................................................................................................................... 6

IV. Results.................................................................................................................................................8

V. Discussion..........................................................................................................................................14

VI. References........................................................................................................................................17

**Notation**

This section is devoted to definitions of all symbols and acronyms, which can be found throughout the assignment report.  
  
***Basic variables*** *– general purpose independent variable*

*– general purpose dependent variable*

*– general purpose function, often:*

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

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

*– a function, which describes propagation of representation error across a scalar function of a scalar variable*

*– a function, which describes propagation of rounding error across a scalar function of a scalar variable****Modifications of basic variables***

*– pure, error-free value of a general purpose independent variable*

*– error-corrupted value*

**The variables associated with are generated in analogous way.  
  
*Operators***

*– operator for determining the* ***relative error***,

*– operator for computing exponential function, e.g.****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 .  
  
There are 2 general ways to obtain   
  
 **1. Method of symbolic differentiation**  
 In general case, can be computed the following way:

**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 as error-corrupted variable , expressed as:

where denotes error-free interpretation of and represents the error.  
  
Next step is to compute . By eliminating common elements, we shall simplify it to:

and hence, obtain

**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 .  
  
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]](#footnote-1), also known as **standard form**. After transformation, we obtain a value in form , where is in 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]](#footnote-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.  
  
**4. Rules for epsilon calculus**Computing both rounding and data errors, one must remember that they are significantly small (). Thus, some special rules are applicable, namely:

**Formulation of problem**

**Task #1**

Determine the function 𝑇 (𝑥) characterising the propagation of the relative error corrupting the

values of to the result of computing the following expression:

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

of 𝑥 stored using the single-precision floating-point representation, where:

Proceed as follows:

a) compute the values :

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

b) compute the values

where denotes the result of storing 𝑥𝑛 using the single-precision representation (single);

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

c) compute the relative errors according to the formula:

Compare the computed errors with the worst-case estimate , where 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 and , characterising

the propagation of the relative errors caused by rounding the intermediate results of computing

according to the following algorithms:

Task #4

Determine the relative errors which corrupt the values when the results of all intermediate operations are stored using the single-precision representation. Proceed as follows:

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

b) by means of the algorithms and , compute the approximate values , ..., and , ..., 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:

and

for

Compare the computed errors with the worst-case estimates: , .

Task #5

Compare the functions , and for   
**Results**Task 1The idea of task is to compare acquired by operating with methods mentioned in Theoretical introduction of this report.  
Firstly, we shall use method of symbolic differentiation

For the sake of clarity, is computed separately:  
  
Substituting the obtained value, we get:  
  
To use Epsilon calculus in order to receive , proceed as follows:

By using the rules of epsilon calculus[[3]](#footnote-3), we get:

*Note:*

Taking into consideration the notation , we conclude:

and thus, **proving the identity** of 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* |

Task 3  
  
Here, we shall compare the efficiency of computing using 2 algorithms.  
The idea is to compute for both algorithms to propagation errors.  
Starting with the first algorithm, one shall gradually imply propagation errors

*Note:*

From this definition, we derive:

The estimation oftotal error can be assessed the following way:

So, the final function is

By following similar steps, we compute as follows:

*Note:*

Thus,  
*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 computed 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.   
  
Изображение выглядит как диаграмма, текст, График, линия

Автоматически созданное описание   
 *Fig 3* *Comparative characteristic of coefficients*Observing *Fig 3* several significant points and intervals can be distinguished. All 3 errors are peaking at and , 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 . 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]](#footnote-4) to illustrate the most ‘interesting’ part of it.

Изображение выглядит как текст, линия, График, число

Автоматически созданное описание  
 *Fig 4* *A frame of the plot of*From *Fig 4* it is clear, that absolute value of denominator reaches its *infimum* at points and , which are exactly the values at which maximum errors are reached.

Speaking of differences, the behavior of the 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 . By plotting (*Fig 5*) a nominator of , we prove that this is exactly the value at which it reaches its *infimum*, causing the error value to drop.  
  
Изображение выглядит как линия, диаграмма, текст, График

Автоматически созданное описание  
 *Fig 5* *A frame of the plot of*   
  
 At , both K-functions plateau, maintaining an error value of , unlike the T function, which increases linearly over this interval.   
  
In conclusion, is more efficient than due to lower maximum error in , but with exception of higher error at peak values. 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.

1. https://en.wikipedia.org/wiki/Scientific\_notation [↑](#footnote-ref-1)
2. https://learn.microsoft.com/en-us/cpp/build/ieee-floating-point-representation?view=msvc-170 [↑](#footnote-ref-2)
3. Formulas (4) – (7) [↑](#footnote-ref-3)
4. https://www.desmos.com/calculator [↑](#footnote-ref-4)