PHYS 514 - Computational Physics

Problem Set 1

Author:
Volkan Aydıngül (Id: 0075359)

Date: October 18, 2020

Contents

1 Problem I **2**

1.1 Calculation of $1 - \frac{\sqrt{1+x^2}}{1+\frac{1}{2}x^2}$ 2

1.2 Calculation of $\cot x^2 - \frac{1}{x^2}$ 3

2 Problem II - Roots of Quadratic Equation **5**

2.1 Alternative Form for Root Finding 5

3 Problem III and IV **6**

1 Problem I

1.1 Calculation of $1 - \frac{\sqrt{1+x^2}}{1+\frac{1}{2}x^2}$

In this problem, we are asked to calcute following function:

$$f(x) = 1 - \frac{\sqrt{1+x^2}}{1+\frac{1}{2}x^2}$$

In addition to the calculation, also the minimum precision level of 10^{-8} is required. Since we are working on 64 *bit* machines, we have always absolute error of 10^{-16}. First, we need to find the x value which corresponds to border of precision, namely 10^{-8}. To do that, one can conduct following relation:

$$Relative\ Error = \frac{Absolute\ Error}{f(x)}$$

$$10^{-8} = \frac{10^{-16}}{f(x)}$$

Above equation can be solved by *Wolfram Mathematica*, and approximate solution can be found as:

$$x \approx 10^{-2}$$

Now, we know that if the x value drops below 10^{-2}, we start to lose precision that we wanted. The basic idea behind this situation is the fact that when x is very low, quatioent part starts to approximate 1 and there happens a loss of signifance due to the $(1-1)$. To deal with this situation, we can expand power series. With help of *Wolfram Mathematica*, one can obtain following power series for $f(x)$.

$$f(x) = 0.125x^4 - 0.125x^6 + 0.101563x^8 - 0.078125x^{10} + O(x^{12})$$

When we input of x around order of magnitude of 10^{-2}, we obtain approximately following values:

$$f(x) \cong 10^{-8} + 10^{-12} + 10^{-16} + 10^{-20} + \dots$$

We know that criteria to truncate the series for a given precision is:

$$\left| \frac{a_{k+1}x_{k+1}}{a_1 x_1} \right| < Precision$$

Satisfying above inequality, we deduce that the first three term is sufficient to obtain precision of 10^{-8}, when the x is lower than 10^{-2}. Finally, we can write our function in a piece-wise manner as following:

$$f(x) = \begin{cases} 1 - \frac{\sqrt{1+x^2}}{1+\frac{1}{2}x^2} & x \geq 10^{-2} \\ 0.125x^4 - 0.125x^6 + 0.101563x^8 & x < 10^{-2} \end{cases}$$

The *loglog* plot that represents the $f(x)$ can be seen in Figure 1.

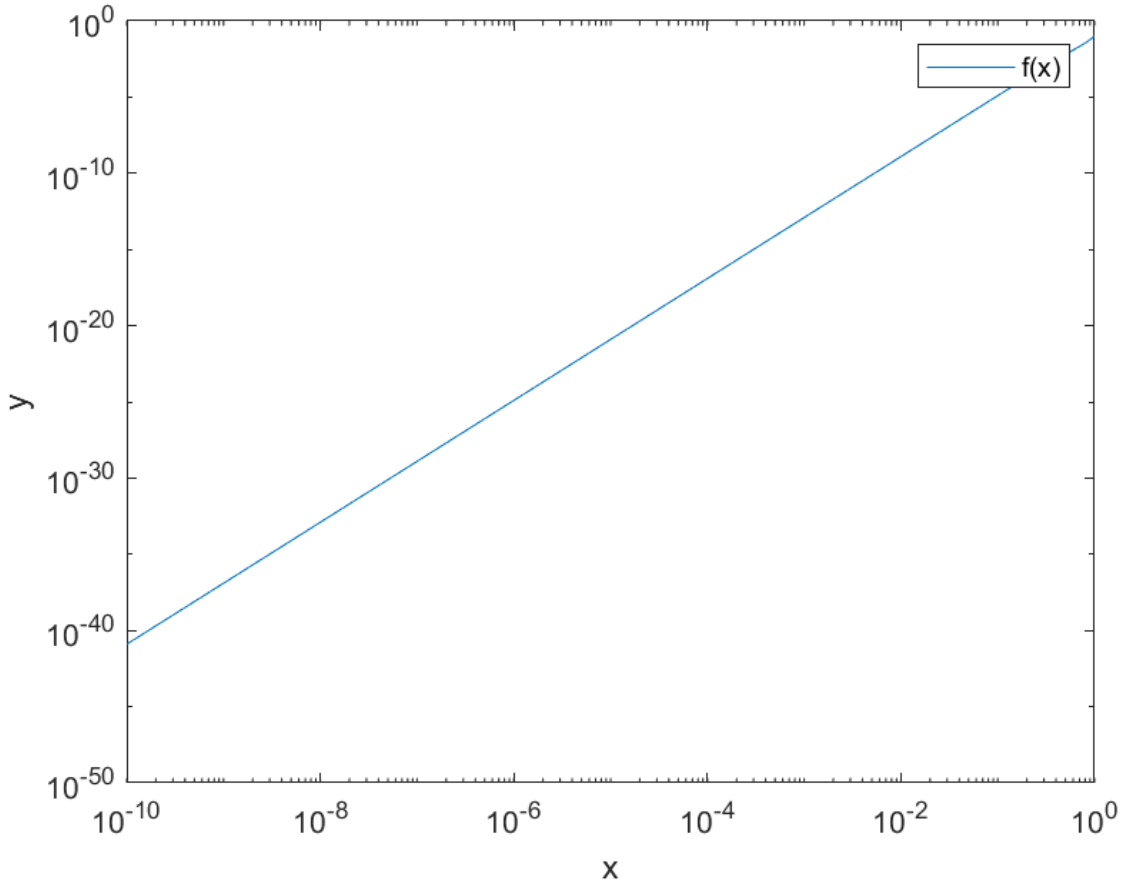

Figure 1: Loglog plot of $f(x)$

1.2 Calculation of $\cot x^2 - \frac{1}{x^2}$

Again, in this problem, we will exactly apply above procedure. In this case, our function is:

$$f(x) = \cot x^2 - \frac{1}{x^2}$$

First, we need to find the x that corresponds to loss of precision of 10^{-8}. Again, by using *Wolfram Mathematica*, we find that:

$$Relative\ Error = \frac{Absolute\ Error}{f(x)}$$

$$10^{-8} = \frac{10^{-16}}{f(x)}$$

$$x \approx 10^{-4}$$

Let's expand power series with help of *Wolfram Mathematica*:

$$f(x) = -\frac{x^2}{3} - \frac{x^6}{45} - \frac{2x^{10}}{945} - \frac{x^{14}}{4725} - \frac{2x^{18}}{93555} + O\left(x^{21}\right)$$

When we input of x around order of magnitude of 10^{-4}, we obtain approximately following values:

$$f(x) \approxeq 10^{-8} + 10^{-24} + 10^{-40} + 10^{-56} + 10^{-80} + \dots$$

We know that criteria to truncate the series for a given precision is:

$$\left|\frac{a_{k+1}x_{k+1}}{a_1 x_1}\right| < Precision$$

Satisfying above inequality, we deduce that the first term is sufficient to obtain precision of 10^{-8}, when the x is lower than 10^{-4}. Finally, we can write our function in a piece-wise manner as following:

$$f(x) = \begin{cases} \cot x^2 - \frac{1}{x^2} & x \geq 10^{-4} \\ -\frac{x^2}{3} & x < 10^{-4} \end{cases}$$

The *loglog* plot that represents the $f(x)$ can be seen in Figure 2.

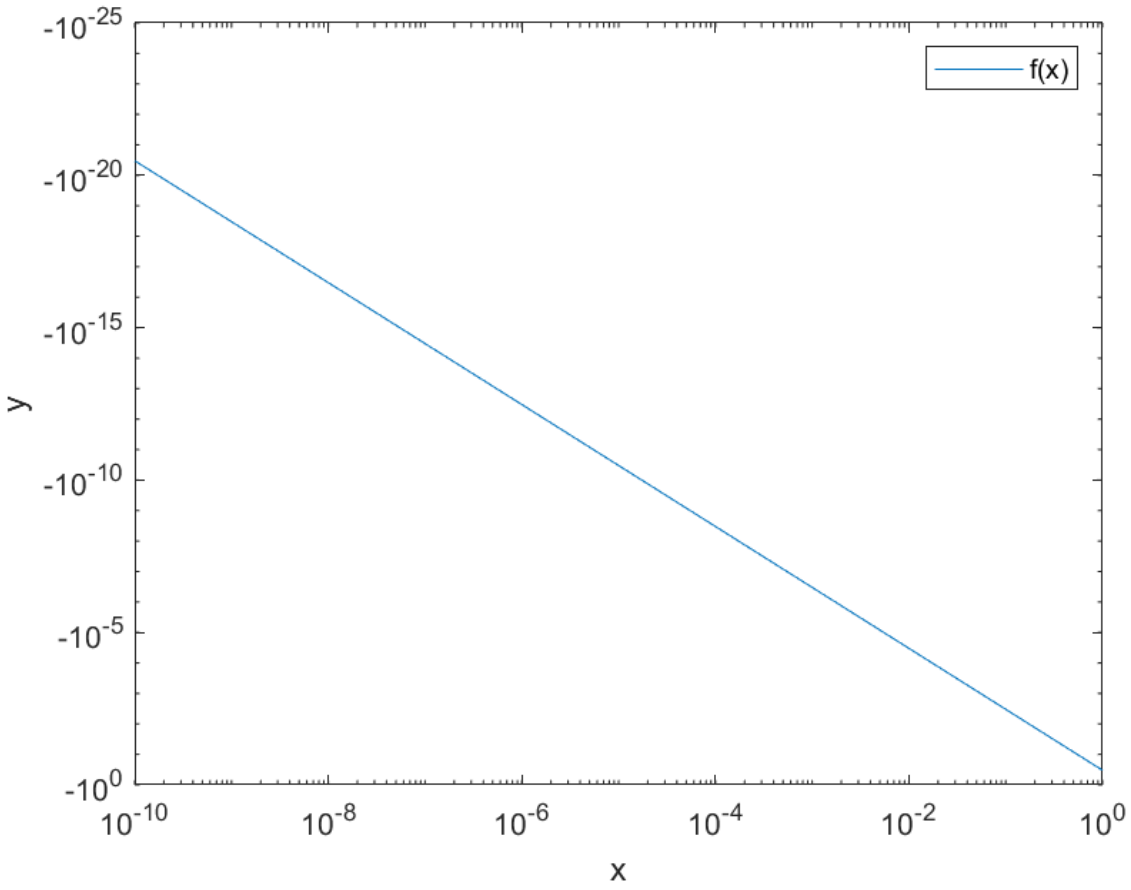

Figure 2: Loglog plot of $f(x)$

2 Problem II - Roots of Quadratic Equation

Let's illustrate the quadratic equation,

$$ax^2 + bx + c = 0$$

In this case, the roots of the above equation can be defined as follows:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

However, when we use above form for root finding, some problems may occur. One of these problems is that when the difference between the terms b^2 and $4ac$ is very large, the loss of significance may take place. For example, when c is very small, depending on the sign of the b, the aforamentioned loss of significance can reduce the accuracy dramatically. In essence, the more stable algorithm which eliminates the usage of substraction is needed.

2.1 Alternative Form for Root Finding

In alternative method, the purpose is to eliminate the need for substraction, which is the main source of the loss of significance. When a or c is very small compared to the b^2, the square term is very likely to result in very close value to b. In this case, ultimately, we need to process following calculation:

$$-b + b^*$$

where b^* stands for *very close number to b*. To get rid of the catastrophic cancellation that may occur due to the above calculation, we can use below formula to calculate the one of the roots.

$$x_1 = \frac{-b - sgn(b)\sqrt{b^2 - 4ac}}{2a}$$

By doing that, we can only encounter with either summation of two positive numbers or summation of two negative numbers. Finally, the second root can be calculated as follows:

$$x_2 = \frac{c}{ax_1}$$

The formula to find second root is simply derived from the *multiplication of the roots* relation:

$$x_1 x_2 = \frac{c}{a}$$

3 Problem III and IV

When we apply the asked procedure the estimate the average round-off error of summing the all elements in an array of n elements, we obtain the following Figure 3:

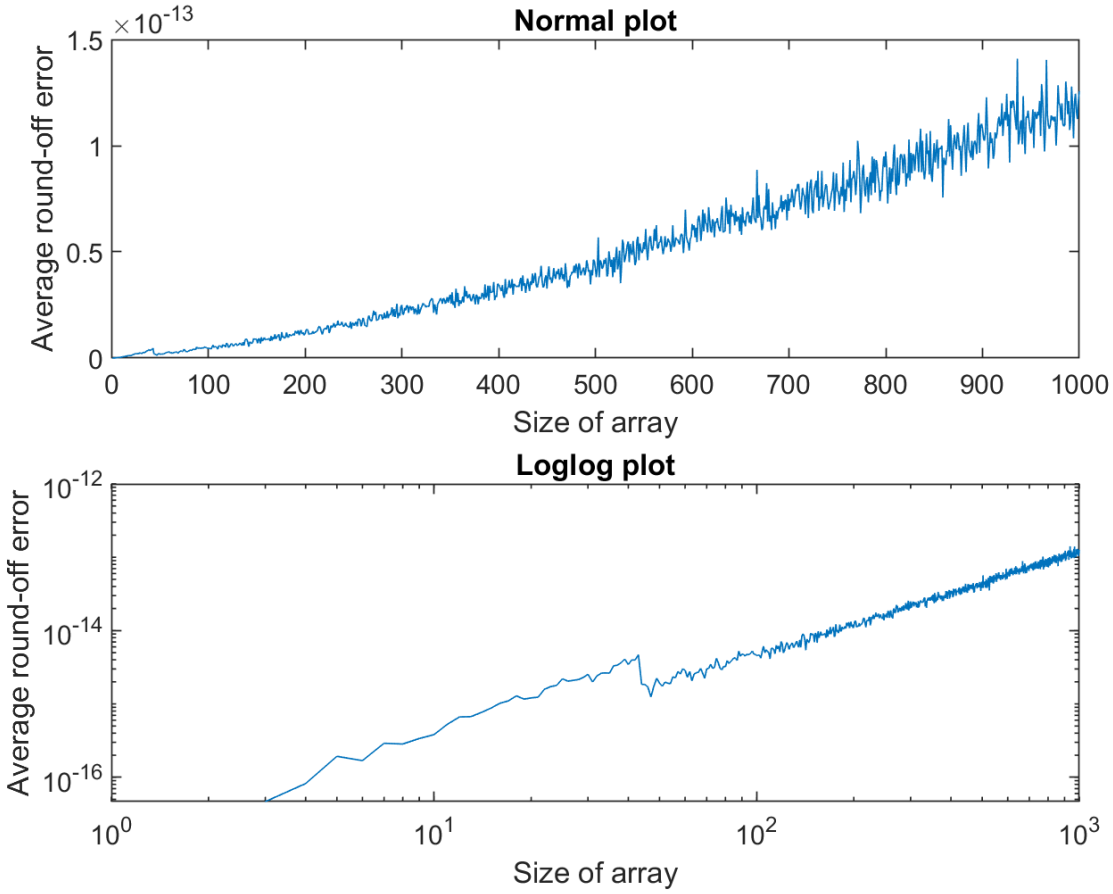

Figure 3: Normal and Loglog plot for MATLAB *sum* function

Observing the figures, from the *loglog* plot, we can clearly see that there is a exponential trend in the growth of error. However, due to the easily observed noise in the Figure 3, it can be deduced that estimate of the average round of error displays a random walk with respect to size of array. Ultimately, we can conclude that even though the error is small, it can accumulate and grow exponentially with the size of array.

To overcome this loss of accuracy, we can employ *Kahan summation*, which is used to minimize the loss of significance. It achieves that by storing a seperate variable which holds the cumulative error.

In this way, the round-off error can reduce significantly, which can be seen from Figure 4.

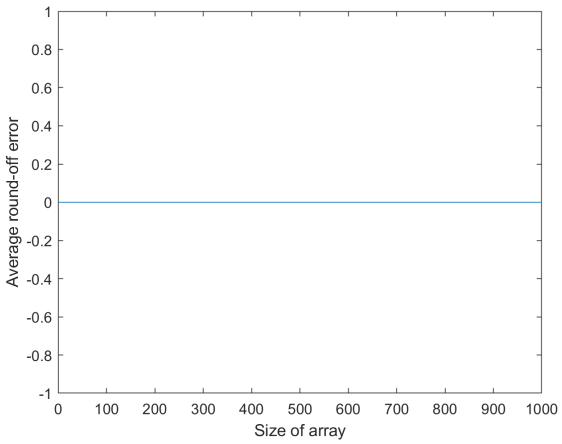

Figure 4: Normal plot for custıom $sumKahan$ function

However, even this algorithm can fail in certain ill-conditioned situations. For example, this algorithm is tend to lose its capability to have very small round-off error when the inputted array does not consists of numbers having mean of zero. In below Figure 5, we can observe that error is not stable around 0, when the input array consists of number having mean of 3 (normal distribution).

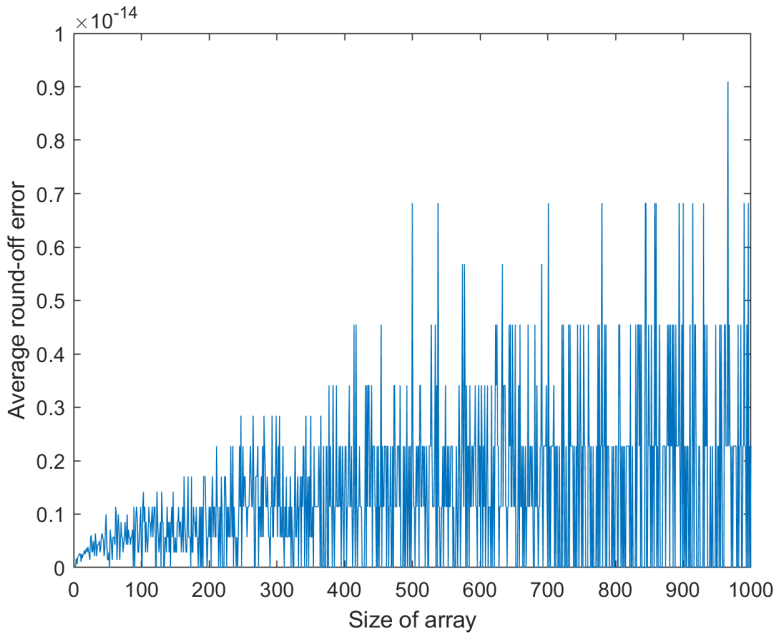

Figure 5: Ill-condition case for Kahan Summation