CS770: Assignment 1

Ronghao Yang
20511820

September 25, 2017

1 Question 1

For $f(x)$, I select

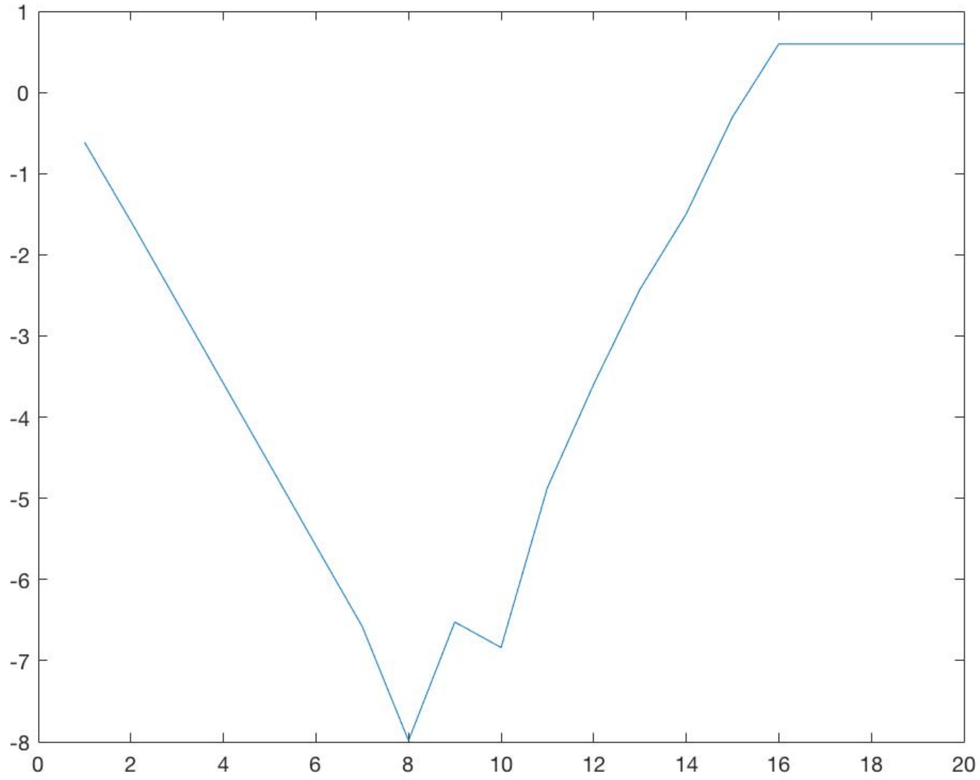
$$f(x) = (x^2 + x)sin(x)^2 \tag{1}$$

$$f(x)^{'} = (2x + 1)sin(x)^2 + (x^2 + x)sin(2x) \tag{2}$$

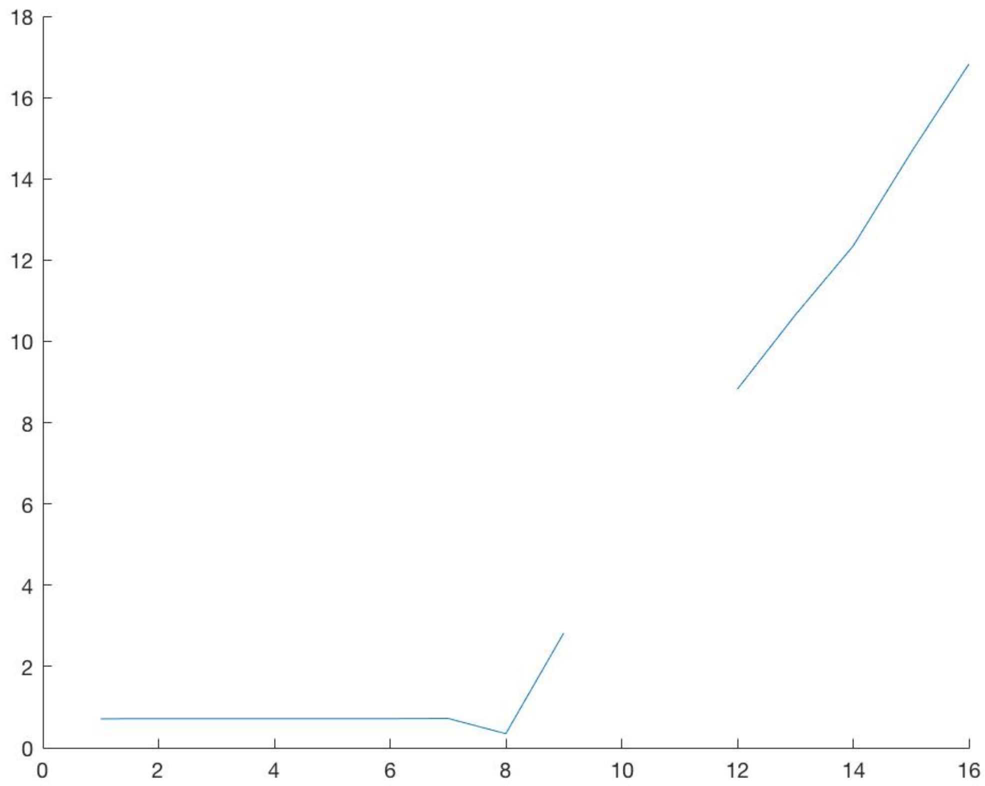
$$f(x)^{''} = 2sin(x)^2 + 2(2x + 1)sin(2x) + 2(x^2 + x)cos(2x) \tag{3}$$

As for my chosen point, I set x to be 2.

1.1 Question 1.a



1.2 Question 1.b



The missing part in the graph is -Inf, this is from the errors being 0(therefore $log10(error) = -Inf$) when h has the value of 10^{10} and 10^{11}

2 Question 2

Inexactness of algebraic operations:

```
%% Inexactness of algebraic operations
a = 0.2;
b = 0;
for i = 1:10000
    b = b+a;
end
b == 2000
```

b is supposed to be 2000, however, due to the error generated in floating point operations, $b == 2000$ is evaluated to false.

Non-commutativity of algebraic operations:

```
%% Noncommutativeness of algebraic operations
a = 0.1;
b = 0.1;
c = 10;
(a*b)*c == a*(b*c)
```

(a*b)*c is supposed to be equal to a*(b*c), however, due to the error generated in floating point operations, $(a * b) * c == a * (b * c)$ is evaluated to false.

Cancellation of errors:

```
%% Cancellation errors

a = 0.2
a-a == 0
```

Although $a = 0.2$ is a approximated, it's error is cancelled when $a - a$ is performed, therefore, $a - a == 0$ is evaluated to true.

3 Question 3

Computation by *Matlab*, the two roots are:

$$x_1 = 9.999999999999999e + 07 \tag{4}$$

$$x_1 = 1.000000000000000e - 08 \tag{5}$$

However, computation by hand gives:

$$x_1 = 10^8 \tag{6}$$

$$x_1 = 0 \tag{7}$$

4 Question 4

4.1 Question 3.1

1) Signs and modulus:
Since we have 1 bit for sign, we have 31 bits for representing the magnitude of a number. Therefore, there are 2^{31} different magnitudes can be represented, and by adding the sign, we have $2 \times 2^{31} - 1 = 2^{32} - 1$ different numbers can be represented.(If +0 and -0 are counted as two different numbers, then there are 2^{32} different numbers)

2) 2's complement:
When using 2's complement for representing numbers, we could have representations for 2^{32} different numbers.

2's complement is the representation for zero unique.

4.2 Question 3.2

When using unsigned numbers, numbers range from 0 to $2^{16} - 1$(0 to 65535 in decimal). When using signed numbers, numbers range from -2^{15} to $2^{15} - 1$ (-32768 to 32767 in decimal).

4.3 Question 3.3

1: 00000001
10: 00001010
100: 01100100
-1: 11111111
-10: 11110110
-100: 10011100

For example, $100 + (-100)$, which in binary representation is $01100100 + 10011100 = 100000000$, since it uses 8 bits for number representation, the first bit 1 is discarded, this gives us 0.

4.4 Question 3.4

With 32 bits representation, a negative number $-x$ is represented using $2^{32} - x$, since -x is negative, -x ranges from -2^{31} to -1, then x ranges from 1 to 2^{31}, then $2^{32} - x$ ranges from 2^{31} to $2^{32} - 1$, this always leaves the first bit of the 32 bits 1. Therefore, all the negative numbers have leading bit 1.
Similarly, for a positive x, x ranges from 0 to $2^{31} - 1$, this always leaves the first bit of the 32 bits 0. Therefore, all the positive numbers have leading bit 0.

4.5 Question 3.5

The last step to complete the process is add 1 to it. Let's say we have 32 bits, x is a positive number, if we simply flip the bits without adding a 1 at the end, $x + (-x) = 2^{32} - 1$. By definition $-x = 2^{32} - x$, therefore, we should add 1 after we flip the bits.

4.6 Question 3.6

In binary representation, 50,-50,100,-100 are represented in the following:
50: 00110010
-50: 11001110

100: 01100100
-100: 10011100

1) 50+ (-100) = 00110010 + 10011100 = 11001110 which is -50 in decimal.
2) 100+ (-50) = 01100100 + 11001110 = 100110010, since the first digit is discarded, the answer is 00110010, which is 50 in decimal.
3) 50 + 50 = 00110010 + 00110010 = 01100100, which is 100 decimal, no overflow occurred.