SOEN6011- Software Engineering Processes

F3: sinh(x)
Github Link: https://github.com/SinglaAnkur/SOEN_6011

Submitted by:
Ankur Singla
40090208

Contents

1 Problem 1

1.1 Introduction

sinh(x) is a hyperbolic[1] sine. This function is related to a hyperbola in the same way as the trigonometric function sin(x) is related to a circle.
Consider a hyperbola: $x^2 - y^2 = 1$

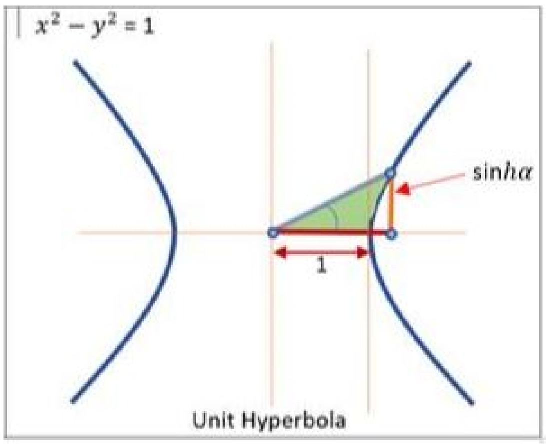

Figure:1.1-1 Hyperbola
Google Images

sinh(x) would be the length of perpendicular drawn from a vertex on hyperbola to the x-axis. The vertex is 1 unit far from the origin.[2]

$$sinh(x) = \frac{e^x - e^{-x}}{2}$$

Domain: $(-\infty, \infty)$
Co-domain: $(-\infty, \infty)$

1.2 Characteristics

1.2.1 As x increases, e^x increases quickly and e^x decreases quickly.[1]
$sinh(x) \approx \frac{e^x}{2}$

1.2.2 If x decreases, e^x decreases quickly and $-e^x$ becomes large.[1]
$sinh(x) \approx \frac{-e^{-x}}{2}$

1.2.3 sinh(x) is an odd function.[1]
$sinh(x) = \sinh(x)$

1.2.4 sinh(x)=0 for x=0.[1]
sinh(x) $\to \infty$ when x $\to \infty$.
sinh(x) $\to -\infty$ when x $\to -\infty$.

2 Problem 2

2.1 Requirements

2.1.1 The value of x shall be a Real number[1].

2.2 Assumptions

2.2.1 x shall be an independent variable[3].

2.2.2 e shall be the base of the natural log[3].

3 Problem 3

3.1 Algorithm 1

This algorithm is based on the expansion of sinh(x) function in the form of Taylor series[4]. The function has been expanded upto n^{th} term. The algorithm is divided into three functions namely: FnPower, FnFactorial, FnCalculate.
Function FnPower is used to calculate the numerator of a term. The value of $result$ variable is returned to function FnCalculate. FnFactorial is used to calculate the denominator of a term. The value of $fact$ variable is returned to function FnCalculate. Function FnCalculate calculates aggregate value of each term and adds it to the variable sum . Output of sum is our desired value of the function .

Algorithm 1: Calculate $sinh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + ... + \frac{x^{2n+1}}{(2n+1)!} + ...$

```
Function FnPower(x,n) /* Function to calculate power */
    Result: Value of x^(2n+1)
    begin
        result ← x; for i ← 1 to n − 1 do
            result ← result × i
        end
        return result
    end
end
Function FnFactorial(n) /* Function to calculate factorial */
    Result: Value of (2n+1)!
    begin
        fact ← 1; for i ← 1 to n do
            fact ← fact × i
        end
        return fact
    end
end
Function FnCalculate(n) /* Function to calculate final value */
    Data: Value of x, x ∈ R
    Result: Value of sinh(x)
    begin
        sum ← 0; for i ← 0 to 70 do
            sum ← sum + FnPower(x,((2*i)+1)) / FnFactorial((2×i)+1))
        end
        return sum
    end
end
```

3.2 Algorithm 2

This algorithm is based on the mathematical expression of $sinh(x) = \frac{e^x - e^{-x}}{2}$.
Lets expand e^x.[5]
$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... + \frac{x^n}{n!}$
$e^x = 1 + (\frac{x}{1}) \times (1 + (\frac{x}{2}) \times (1 + (\frac{x}{3})(....)))$
Using this expansion we can calculate the value of e^x.
Also e^{-x} could be calculated by inserting x as -x in the expansion. FnExponential function calculates above expansion . FnComparison function calls the FnExponential function based on the value of x.

Algorithm 2: Calculate $sinh(x) = \frac{e^x - e^{-x}}{2}$

Function `FnExponential(`n,x`)` `/* Function to calculate exponential */`
  Result: Value of e^x
  begin
    $sum \leftarrow 1$; **for** $i \leftarrow n-1$ **to** 1 **do**
      $sum \leftarrow 1 + x \times \frac{sum}{i}$
    end
    return sum
  end
end
Function `FnComparison(`x`)` `/* Finds value of the exponential based on the input value of x */`
  Result: Value of $sinh(x)$
  begin
    $output \leftarrow 0$; $abs\_x \leftarrow x$; **if** x==0 **then**
      return 0
    else if x<0 **then**
      $abs\_x = x \times (-1)$
    if abs_xc0 *&&* $abs\_x \leq 15$ **then**
      for $n \leftarrow 2$ **to** 1000 **do**
        $value\_e^x \leftarrow$ `FnExponential`(n, abs_x)
        $value\_e^{-x} \leftarrow$ `FnExponential`$(n, -abs\_x)$
      end
      output=$\frac{value\_e^x - value\_e^{-x}}{2}$
    else if abs_x>15 *&&* $abs\_x \leq 700$ **then**
      for $n \leftarrow 2$ **to** 1000 **do**
        $value\_e^x \leftarrow$ `FnExponential`(n, abs_x)
      end
      output=$\frac{value\_e^x}{2}$
    if x<0 **then**
      return $output \times (-1)$
    else
      return $output$
    end
  end
end

3.3 Comparison

After analyzing both algorithms, Algorithm 2 was selected as the choice. Below are the advantages and disadvantages of choosing the particular algorithm.

3.3.1 Advantages

Algorithm 1

3.1.1.1 It is easy to understand.

3.1.1.2 It is easy to maintain.

Algorithm 2

3.1.1.3 It provides more accurate values.
Example: For x=100, Algorithm 1 outputs as 1.3439570961820604E43 and Algorithm 2 outputs as 1.344058570908067E43.

3.1.1.4 It provides output for higher domain: x $\in$ (-700,700) as compared to Algorithm 1 x $\in$ (-150,150)

3.3.2 Disadvantages

Algorithm 1

3.1.2.1 It provides inaccurate values as stated in [3.1.1.4]

3.1.2.2 It results in Infinity for x>150 and -Infinity for $x< -150$.

Algorithm 2

3.1.2.4 It is hard to maintain.

3.1.2.5 It is hard to understand.

3.1.2.5 It results in Infinity for x>700 and -Infinity for $x< -700$.

4 References

[1] MathCentre, 'Hyperbolic functions. Available: 2006 http://www.mathcentre.ac.uk/resources/workbooks/m
[Accessed: 3-July-2019].
[2] LumenCollege 'Equations of Hyperbolas' https://courses.lumenlearning.com/waymakercollegealgebra/chapt
of-hyperbolas/ [Accessed: 3-July-2019]
[3] Alcyone Systems 'Mathematics reference' Available: 2019 http://www.alcyone.com/max/reference/maths/h
[Accessed: 4-July-2019]
[4] UIC 'Stat 401: Introduction to Probability' Available: 2006 http://homepages.math.uic.edu/ jyang06/stat40
[Accessed: 13-July-2019]
[5] GeeksForGeeks team 'Efficient program to calculate e^x ' Available: 2012 https://www.geeksforgeeks.org/program-to-efficiently-calculate-ex/ [Accessed: 17-July-2019]