

William Santosa
wsantosa@ucsc.edu
24 January 2021

CSE13S Winter 2021
Assignment 2 : A Small Numerical Library
Design Document

DESCRIPTION

The math.h header from the C library contains many different functions for a variety of computational needs. They include, but are not limited to, sin, cos, tan, exp, and log. In this lab, students aim to implement these functions ourselves to better understand how they work and deepen our comprehension of complex computations on computers.

Files:

- mathlib.h
 - Contains prototypes of the math functions
- mathlib.c
 - Contains implementations of the math functions
- mathlib-test.c
 - Contains main() and has command-line options to print results of my math functions
- Makefile
 - Runs program
- README.md
 - Information
- DESIGN.pdf
 - Describes purpose, covers the layout, clear description of program parts, pseudo code
- WRITEUP.pdf
 - Discussion of results for the tests

TOP LEVEL DESIGN / PSEUDOCODE

mathlib.c

```
#include "mathlib.h"
```

```
#define EPS 1e-14 // 1x 10-14
```

```
Sin(x) // From Slides 8 - Numerical Computation  
{  
    Sum = x
```

```

    Numerator = x
    Denominator = 1
    for(K < EPS)
    {
        Add 2 to the numerator power and invert sign
        Add 2 to the factorial of denominator
        Add new (numerator/denominator) to sum
    }
    return sum
}

```

```

Cos(x)
{
    Sum = x
    Numerator = x
    Denominator = 0
    for(K < EPS)
    {
        Add 2 to the numerator power and invert sign
        Add 2 to the factorial of denominator
        Add new (numerator/denominator) to sum
    }
    return sum
}

```

```

Tan(x, y)
{
    return Sin(x) / Cos(y)
}

```

```

Exp(x) // From Slides 8 - Numerical Computation
{
    term = 1
    sum = term
    for(Abs(term) > EPS)
    {
        Multiply term by (x / k)
        Add term to sum
    }
    return sum
}

```

```

Log(x) // From Slides 8 - Numerical Computation
{

```

```

y = 1
p = Exp(y) // p = e^y
while (Absolute value of (New exponent subtracted by x) > EPSILON)
{
    Set y equal to y plus (x minus p) divided by p
    p = Exp(y)
}
return y
}

```

```

Abs(x) // From Eugene's Lecture (1/19/21 23:20)
{
    if (x < 0)
    {
        Multiply x by -1
    }
    return x
}

```

```

Sqrt(y) // From Eugene's Lecture (1/19/21 23:20)
{
    term = 1
    old = 0
    while(Absolute value of (term subtracted by old) > EPSILON)
    {
        Set Old to term
        Set term to (0.5 * (Term + y / Term))
    }
    Return term
}

```

mathlib-test.c

```

Set opt to 0
Boolean for all functions
Obtain command line input
switch(opt){
    Do all cases and make boolean true for that function if in opt
    break
}

```

```

For each boolean function
if(true){
    Print top part
}

```

Print numbers

}

switch(opt)

- Sets boolean var to true if case is done for it
- Breaks so it doesn't go to all the other stuff

For each boolean function

- Top part should use `printf("x____Sin____Library____Difference\n-----")`
- Print numbers should use for loop or while loop to iterate over all the numbers required
 - N, Sin(N), sin(N), (Sin(N) - sin(N))

DESIGN PROCESS / MODIFICATIONS

Haven't done any yet, but I expect that I will have to do a whole bunch of revisions to the design document...

- I'm stupid. Tan(x) is supposed to only have one input, not two, since it's $\sin(x) / \cos(x)$
 - Changed from Tan(x, y) to Tan(x)
- Don't need a term variable. Just used numerator / denominator instead.
- Added pseudocode for mathlib-test.c cause I didn't do that apparently
- Had to change cosine as well, starts at 1 instead of x