# EE231002 Introduction to Programming

Lab 21. Trigonometric Functions

**Due: None**

In this lab, we will practice calculating the values of trigonometric functions using infinite series. There are multiple parts in this lab and it is recommended to complete a part as efficiently as you can before moving on to the next part.

**Part 1.**

It is known that sin and cos functions can be represented by the following infinite series.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \tag{21.1}$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \tag{21.2}$$

Using these formulas please write the following two functions to approximate the values of $\sin x$ and $\cos x$.

```
double sinE(double x, double eps);
double consinE(double x, double eps);
```

These two functions approximate Eq. (21.1) and (21.2), respectively. The additional parameter `eps`, which is a small number, is used to terminate the summation of the series. Note that in calculating either $\sin x$ or $\cos x$ a new term in the form of $\frac{x^k}{k!}$ is added or subtracted each time. Thus, we can terminate the calculation if the new term is less than `eps`. In this way, the error in `sinE` or `cosinE` is guaranteed to be smaller than `eps`. As always, please write these two functions as efficient as possible.

To verify your implementation of these two functions, please write a `main` function that calls `sinE` and `cosinE` 100 times to generate a trigonometric table shown at the end of this PDF file. You should try different values of `eps` to see if it would impact the execution time or the accuracy of the results.

**Part 2.**

In real applications, we want to calculate these functions as accurately as possible. Thus, `eps` should be set to 0. However, if $x$ is not small, the summation can take many terms. We know that the floating number system for any computer has limited number of significant digits and `eps` smaller than the resolution of the floating number is really meaningless. Therefore, we can change the termination condition to be following.

For $k$'th iteration we have the summation operation as

$$sum_{k+1} = sum_k \pm \frac{x^{(2k)}}{(2k)\,!} \tag{21.3}$$

If

$$|sum_{k+1} - sum_k| = 0 \tag{21.4}$$

then the calculation can be terminated without loss of accuracy given the floating number system of the computer.

Please write the following functions using Equations (21.3) and (21.4) as the core of the operations.

```
double sine(double x);
double consine(double x);
```

Modify the `main` function of Part 1 to call the new functions. Please compare the efficiency of these two sets of functions in case of similar accuracy.

**Part 3.**

With those two basic trigonometric functions implemented efficiently, one can complete the remaining trigonometric functions.

$$\tan x = \frac{\sin x}{\cos x} \tag{21.5}$$

$$\cot x = \frac{\cos x}{\sin x} \tag{21.6}$$

$$\sec x = \frac{\sin x}{\cos x} \tag{21.7}$$

$$\csc x = \frac{\cos x}{\sin x} \tag{21.8}$$

Thus, implement the following functions as efficiently as you can.

```
double tagent(double x);
double cotagent(double x);
double secant(double x);
double cosecant(double x);
```

Also, modify the `main` function to verify your implementation and check the accuracy as well as efficiency of your implementation.

**Notes.**

1. This lab is for you own practice. You do not need to submit your program. But your are encouraged to do it. This is to let me know, indeed, there are people doing this lab. If there are a good number of submissions, then the next practice lab might be announced. As before, use the following command to submit your program.

   ```
   $ ~ee2310/bin/submit lab21 lab21.c
   ```

2. The main objective of this lab is the efficiency. But, please remember the class coding guidelines. Please maintain the good coding habit, it would benefit you in the future.

---

Example of program output for Part 1.

```
   x              sin             cos
0              0               1
0.0314159265   0.0314107591    0.9995065604
0.0628318531   0.0627905195    0.9980267284
0.0942477796   0.0941083133    0.9955619646
0.1256637061   0.1253332336    0.9921147013
0.1570796327   0.1564344650    0.9876883406
0.1884955592   0.1873813146    0.9822872507
     ...            ...             ...
     ...            ...             ...
     ...            ...             ...
2.9845130209   0.1564344650   -0.9876883406
3.0159289474   0.1253332336   -0.9921147013
3.0473448740   0.0941083133   -0.9955619646
3.0787608005   0.0627905195   -0.9980267284
3.1101767271   0.0314107591   -0.9995065604
3.1415926536   0.0000000000   -1.0000000000
```

---