

CS 447/547: Numerical Integration

Due February 3, 11:59 PM.

(This document was last modified on Monday, January 27, 2020 at 07:31:25 AM.)

This is a warm-up assignment, intended to refresh your C, introduce you to parallel programming, and get you used to programming in a numerical context. The task is to compute a definite integral numerically. You may use a quadrature method, or Monte Carlo. Do not use Taylor series, etc. for this assignment. You should check your method with the instructor to make sure that it is okay.

You may use C++ (recommended) or C, or any mix of the two. The method of parallelization should be pthreads or C++ threads. Other languages may be allowed. If you are interested, you may contact the instructor.

The definite integral to integrate is:

$$\int_a^b \sin(x)/x \, dx$$

Your makefile should create an executable named `integrate`. The executable should take the following command-line arguments:

```
$ ./integrate a b n n_threads
```

The a and b arguments give the limits of integration. They may be any floating point number, and may be negative. The n arguments gives the total number of samples across all threads. The $n_threads$ arguments gives the number of threads. Thus, each thread should execute approximately $n/n_threads$ samples. At the end of execution, your code should print a single number as the answer. Do not print anything else.

In addition to your code, submit a graph of both speedup and efficiency. For our purposes, we will define speedup as:

$$S_p = \frac{T_1}{T_p}$$

where S_p is the speedup for p threads, T_1 is the wall-clock time for one thread, and T_p is the wall-clock time for p threads. (Note that our definition is somewhat less restrictive than the definition that is usually used. We will explain this more later in the course.)

Efficiency is defined as:

$$E_p = \frac{S_p}{p}$$

In both graphs, the x -axis should be p , the number of threads. You should also submit a brief report (no more than 1 page) explaining your results.

Evaluation

General grading criteria are [here](#).

Your submission will be graded on correctness and scalability. It should scale well up to 10 cores. You will also be graded on your report, and how well it explains your results. I encourage you to discuss the results in office hours with me.

In addition, we will have a extra credit contest for performance. The goal will be to attain the most accuracy for a given wall clock time. To enable this mode, your executable should examine the n command-line argument. If it ends with an `s`, then the argument should be interpreted as the execution time. You should terminate execution after the time limit is reached, and print your answer. The top three submissions will receive 30, 20, and 10 extra credit points, respectively.

You must have 10 commits for this assignment.

Submission

Your submission should consist of your code, your graphs (as PNGs please, don't use JPEG), your report as either a PDF, Word, or ASCII. Include a makefile that will build your code.