Random Number Generation and Simulation

Exercise 8

Area estimation using Monte Carlo method

Author
Cesare De Cal

Professor
Annie Cuyt

Assistant Professor
Ferre Knaepkens

December 4, 2019

1 Introduction

The exercise asks to approximate the area of the figure defined by

$$\begin{cases} 1 \leq x \leq 3 \\ -1 \leq y \leq 4 \\ x^3 + y^3 \leq 29 \\ y \geq e^x - 2 \end{cases}$$

using the Monte Carlo method.

2 Tools

To solve this exercise, I've used the following libraries and programming languages:

- C
- Intel Math Kernel Library (more specifically, the Vector Statistical Library)
- OpenMP
- C Math Library

I've used the following Intel MKL routines:

- `vslNewStream(&stream, brng, seed)`
- `vslLeapfrogStream(stream, k, nstreams)`
- `vsRngUniform(method, stream, nrRandomNumbers, array, start, end)`
- `vslDeleteStream(&streamToDelete)`

OpenMP provides a user-friendly interface to build multi-threading applications. I've used the following methods and procedures:

- `omp_get_max_threads()`
- `#pragma omp parallel private(nrOfThreads, threadID)`
- `omp_get_thread_num()`

To make the code more clear, I've also wrote my own function `isInsideArea(x,y)` which checks if a given pair of coordinates (x, y) is inside the area drawn by the system of inequalities. To get started with the exercise, I have used the template file `template.c` provided on the website.

To compile and run the code on my own machine with `gcc`, I created a Makefile based on the compiler options and link line specified here:
`https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor`.

3 Computation

On a high level picture, the Monte Carlo method in this exercise works by generating random (x, y) coordinates in the rectangle formed by

$$\begin{cases} 1 \leq x \leq 3 \\ -1 \leq y \leq 4 \end{cases}$$

And then checks if the points satisfy the following inequalities using the `isInsideArea(x,y)` function I wrote.

$$\begin{cases} x^3 + y^3 \leq 29 \\ y \geq e^x - 2 \end{cases}$$

The area is given by the ratio of how many points satisfied the system (in other words, are inside the figure) to the total number of iterations N. This is repeated $LOOPS$ times, and finally I calculate the average of all the areas.

To make this happen, I created two independent streams of uniformly distributed random numbers.

4 Plots

5 Observations