C++ training – project

Pierre Gruet

December 2025

This project is for the validation of the course. Below are the general instructions:

- The project is for a groupe of two students, or alternatively you may work alone if you wish;
- Please send a zip file with your code and a pdf file with your report before the 11^{th} of February, 2026 at 23:59 CET. Don't forget to give me the names of all the people in your group;
- In the pdf file, please write your report (see Section 4 for the contents). No need to include code snippets in this pdf file. But be sure to organize your files properly so that is it easy to find the code corresponding to each part of the report;
- Don't forget to write useful comments in your code.

1 General objective

The aim of this project is to develop a library to compute integrals

$$\int_a^b f(x)dx$$

numerically, for functions f that are integrable over intervals (a, b) with $a \in \mathbf{R}, b \in \mathbf{R}$. Different methods will be used.

Beyond the mathematical interest of the task, it is expected that your code shows a good understanding of the contents of the course of C++.

2 Requirements

Mathematically:

- You have to implement at least three methods summarized in Masud et al. [1];
- You also have to implement a Monte-Carlo approach: if $X_1, ..., X_n$ are n simulations of a random variable X following an uniform distribution on the interval $[a, b]$, then $\frac{b-a}{n} \sum_{i=1}^{n} f(X_i)$ converges to $\int_a^b f(x)dx$ in probability;

- Your code should be able to approximate successfully the following integrals, of which values are given for your convenience:

$$\int_0^1 x^2 \cos(x) dx = 2\cos(1) - \sin(1), \quad \int_0^1 x^{10} dx = \frac{1}{11}, \quad \int_0^1 x^{-1/2} dx = 2, \quad \int_0^1 \log(x) dx = -1.$$

  You are welcome to test your code on more functions.

From the C++ point of view:

- The mathematical procedures to approximate integrals should be implemented in different classes. All of these classes must be (possibly transitively) derived classes of one abstract class `Solver`;
- The functions should be computed in different classes. All of these classes must be (possibly transitively) derived classes of one abstract class `Function`;
- Think about the class hierarchy to avoid having to duplicate code;
- Use polymorphism and the two above abstract classes;
- Use the containers of the standard library as much as you can instead of defining your own types;
- You may use any external library, except numerical integration ones of course;
- Using CMake (or other build tools) is a plus.

3 Possible extensions

It is expected that you also address at least one of the two below extensions:

- Integrating 2-dimensional functions: try to rework your code to work on 1-dimensional functions (as above) or to approximate integrals

$$\int_a^b \int_c^d f(x, y) dy dx$$

  of functions $f : \mathbf{R}^2 \to \mathbf{R}$. Think about the architecture of your code;
- Download the GNU Scientific Library (GSL), which has numerical integration capabilities. Interface it with your code and use it to check your results on 1D function integration.

4 Expectations

In your pdf report, you should explain what methods you implemented. Briefly expose their principle, then present a benchmark of the methods based on several test functions (given above). In this benchmark, please discuss your choice of the meta-parameters such as the number of discretisation points or the number of Monte-Carlo simulations. Also explain the architecture of your code.

In a second part, discuss the extension (see Section 3) that you chose: what tests did you perform? Did you meet any difficulty?

In the last part, please explain what was your use of Artificial Intelligence for this project, if you used it.

References

[1] M Masud, Faijun Nesa Shimi, and Rathindra Chandra Gope. Numerical integration techniques: a comprehensive review. *International Journal of Innovative Science and Research Technology*, pages 2744–2755, 2024.