Documentation Monte Carlo Project

1. Overall design

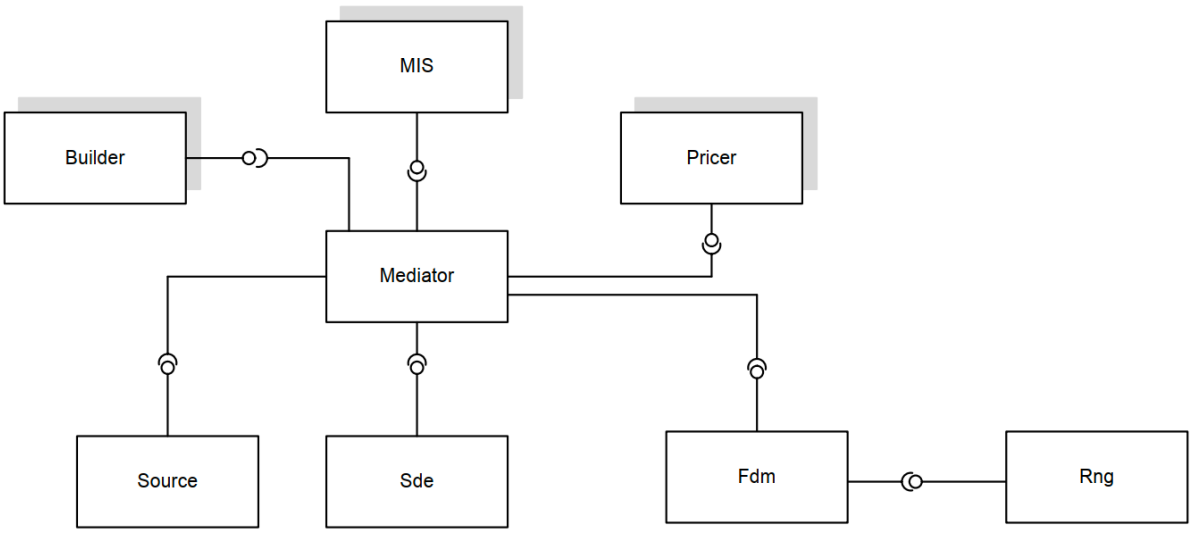

Figure 1 - Context Diagram Monte Carlo Engine

- **Source**: The system containing the data relating to the request, for example price a plain one-factor option with given market data. It contains data that is needed by other modules in Figure 1.
- **Sde**: The system that models stochastic differential equations (SDEs). In this case we model Geometric Brownian Motion (GBM) and its variants. In particular, we are interested in modelling the drift and diffusion of some underlying variable such as the stock price or interest rate, for example.
- **Fdm**: The family of finite difference schemes that approximate the sdes in the Sde system. In this case we use one-step difference schemes to advance the approximate solution from one time level to the next time level until we reach the desired solution at expiration. The finite difference schemes require the services of a module that computes random numbers and standard Gaussian variates, that is variates with mean zero and standard deviation one.
- **Pricer**: This system contains classes to price one-factor options using the Monte Carlo simulation technique. The classes process path information from the Mediator and each class processes this path information in its own way. For example, for a plain option the pricer uses the path data at expiration, uses it to compute the payoff, adds the result to a running total and then discounts the result to compute the option price.

- **MIS**: This is the statistics-gathering system that receives status information concerning the progress of computation. For example, this system could display how many paths have been processed at any given time.
- **Builder**: This system implements a configuration/creational pattern (based on GOF 1995 but more general) that creates and initialises the systems and their structural relationships in Figure 1. The newly-created objects are encapsulated in .NET tuples which adds to the overall maintainability of the system.
- **Mediator**: This is the central coordinating entity that manages the data flow and control flow in the system. It is the driver of the system as it were and it contains the state machine that computes the paths of the
- **SDE**. It also informs the other systems of changes that they need to know about. It also plays the role of client in the Builder pattern (GOF 1995).
- **Rng**: a system to generate random numbers.
- Additionally, the project also includes additional parts not represented in the Figure 1 diagram, such as an interface that displays the results on the console, a stopwatch used to measure processes duration, a singleton class which is the base class of the interface.

The project uses concepts going from C++11 to C++20, such as execution policies, concepts, ranges and more.

2. Testing

This test simply calculates the call and put prices of two European options with different parameters and compare the estimated prices to the analytical price of the Black-Scholes formula. Are also being compared the processing time with different number of simulations and time subdivisions (for FDM), for the single-threaded and multi-threaded cases. The stochastic differential equation and finite difference method used in these tests are Geometric Brownian Motion and Euler Method, respectively.

Test option 1:

Option parameters:

Initial stock price = $60	Volatility = 30%
Strike price = $65	Interest rate = 8%
Time-to-maturity = 3 months	Dividend yield = 0.0%

Analytical (BS) call and put prices:

Call price = $2.13337 **Put price = $5.84628**

Column1	Simulation count	NT	Call price	Put Price	Time elapsed
Single-Threaded	100,000	100	$2.1252	$5.8571	1.69289s
	100,000	250	$2.1436	$5.8530	4.42s
	100,000	500	$2.1225	$5.8787	11.0041s
	500,000	100	$2.1401	$5.8437	7.93546s
	500,000	250	$2.1325	$5.8597	22.1925s
	500,000	500	$2.1355	$5.8564	53.8677s
	1,000,000	100	$2.1352	$5.8464	15.9282s
	1,000,000	250	$2.1281	$5.8401	43.963s
	1,000,000	500	$2.1233	$5.8547	112.471s
Multi-Threaded	100,000	100	$2.1283	$5.8186	0.385537s
	100,000	250	$2.1098	$5.8636	1.15403s
	100,000	500	$2.1124	$5.8605	2.54178s
	500,000	100	$2.1156	$5.8159	1.94886s
	500,000	250	$2.1384	$5.8293	6.01182s
	500,000	500	$2.1245	$5.8535	13.5209s
	1,000,000	100	$2.0995	$5.8135	4.21371s
	1,000,000	250	$2.1274	$5.8517	11.6981s
	1,000,000	500	$2.1383	$5.8326	32.8281s

Figure 2 - Test #1 Results

Test option 2:

Option parameters:

Initial stock price = $100

Strike price = $100

Time-to-maturity = 30 years

Volatility = 30%

Interest rate = 8%

Dividend yield = 0.0%

Analytical (BS) call and put prices:

Call price = $92.1757

Put price = $1.2475

Column1	Simulation count	NT	Call price	Put Price	Time elapsed
Single-Threaded	100,000	100	$88.7610	$1.2996	1.54511s
	100,000	250	$92.4502	$1.2689	4.48264s
	100,000	500	$92.8257	$1.2646	10.8449s
	500,000	100	$90.0876	$1.2933	8.0278s
	500,000	250	$91.9620	$1.2626	21.9529s
	500,000	500	$91.2751	$1.2515	53.6995s
	1,000,000	100	$89.6311	$1.2926	15.3473s
	1,000,000	250	$91.6004	$1.2616	45.895s
	1,000,000	500	$91.7501	$1.2561	107.215s
Multi-Threaded	100,000	100	$90.6074	$1.2827	0.385537s
	100,000	250	$89.3657	$1.2537	0.997199s
	100,000	500	$90.2901	$1.2547	2.78077s
	500,000	100	$89.7234	$1.2809	1.90774s
	500,000	250	$89.9718	$1.2569	5.70894s
	500,000	500	$91.5312	$1.2550	14.1182s
	1,000,000	100	$88.8526	$1.2857	4.0728s
	1,000,000	250	$91.6406	$1.2622	12.4075s
	1,000,000	500	$91.2179	$1.2511	28.4154s

Figure 3 - Test #2 Results

We can directly observe on the above results that the multi-threaded case performs much better time-wise than the single-threaded, without a too large impact on the prices accuracy.

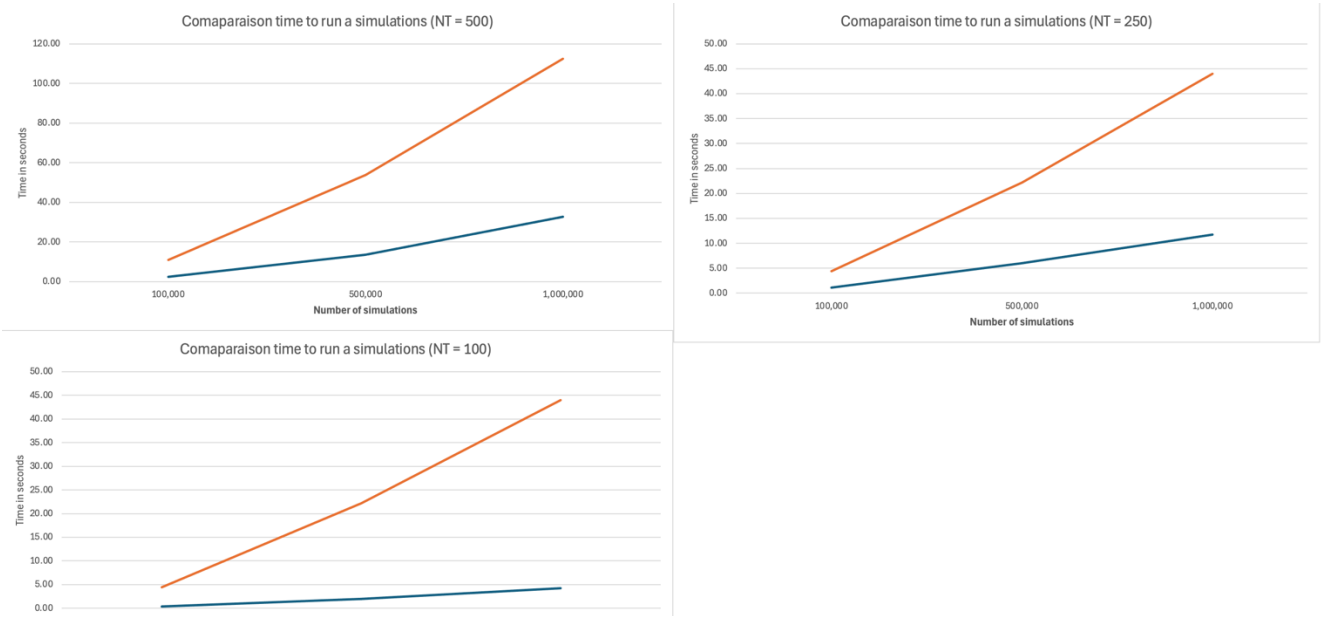

Figure 4 - Comparaison Charts