MC and QMC Simulation in PREMIA

Anne GILLE-GENEST

Premia 18

In this part, we describe how Monte Carlo simulation is organized in PREMIA. We first enumerate implemented simulation methods according to the option type. Them we detail organisation for documentation about simulation methods. A third point explains how to realize a simulation in Premia and results given by the methods. Some advices about understanding of the results are finally discussed.
This part contains a lot of links which allow to reach directly a more detailed file for most of the treated points.

Implemented simulation methods in Premia

At the moment, only European options can be priced with Monte Carlo or Quasi Monte Carlo simulation methods.
New algorithms for American options will be added in a next version.

bs1d and bs2d denote respectively Black and Scholes model in 1 or 2 dimensions.

• **Standard options in bs1d:** Call, Put, Call-Spread and Digit options.
MC and QMC Standard
MC and QMC with antithetic variables
Call option is computed with a control variate method through the Call-Put parity relation.

• **Standard options in bs2d:** Call on maximum, Put on minimum, Exchange and BestOf options.

MC and QMC Standard.

• **LookBack options in bs1d:**
- Call-Fixed and Put-Floating options: MC and QMC Standard.
- Call-Floating and Put-Fixed options: MC and QMC Standard.

• **Asian options in bs1d:**
- Call-Fixed and Put-Fixed: MC and QMC with Kemna and Vorst control variate method.
- Call-Floating and Put-Floating: MC and QMC Standard.
Discretization of the process with M points (forward simulation), this involves that QMC can not always be used if M is greater than maximal dimension available for the considered low-discrepancy sequence.
Approximation schemes to estimate the spot average until maturity.

• **Simple Barrier options in bs1d:**
- In options: Call-Up, Call-Down, Put-Up, Put-Down.
MC and QMC Standard.
Discretization of the process with M points (forward simulation), this involves that QMC can not always be used if M is greater than maximal dimension available for the considered low-discrepancy sequence.
- Out options: Call-Up, Call-Down, Put-Up, Put-Down.
Routine mc_parisianout_bs.c

• **Double Barriers options in bs1d:**
MC Standard. Routine mc_outbaldi.c Routine mc_inbaldi.c
QMC simulation is not yet implemented.

• **Parisan options**
MC Standard. Routine mc_parisianout_bs.c Routine mc_parisianin_bs.c
Routine mc_parisianupdownout_bs.c Routine mc_parisianupdownin_bs.c

Documentation about simulation methods in Premia

Whole documentation written for Premia about simulation methods is firstly composed of a general part about Monte Carlo and Quasi Monte Carlo simulation methods, where theoretical approach is described and some simulation algorithms for random variables and diffusion are detailed.
Three other independent parts are dedicated to uniform numbers generators.
For Monte Carlo simulation, we are interested in pseudo-random numbers generators whereas for Quasi-Monte Carlo simulation we present low-discrepancy sequences.

Implementation of these both types of generators is explained in a special part. Parameters like maximal available dimension are specified.

Furthermore, each implemented algorithm (mentionned in the first point) is documented in a specific file.

Two articles about simulation methods are available in a pdf-version. The first one deals with Monte Carlo simulation for Asian options. The second one with Monte Carlo simulation for Barrier options.

A simulation in Premia

We now detailed three points connected with the simulation. First we give precision on input parameters necessary to begin a simulation. Then we recall what are the results for a MC or a QMC simulation and finally we discuss on interpretation of the results.

• Input parameters:
For all simulation algorithms, you need to give values at the following parameters. However for each of them, an initial value is suggested.
- `N` is the number of iterations in the simulation cycle. It must be a integer greater than 2.

- `RandomGenerator` is the index of the generator you choose for your simulation. Values are integers given in the following table. According to this input value simulation will be a MC or a QMC one.

Value	Generator	Type
0	KNUTH	MC
1	MRGK3	MC
2	MRGK5	MC
3	SHUFL	MC
4	L'ECUYER	MC
5	TAUSWORTHE	MC
6	SQRT	QMC
7	HALTON	QMC
8	FAURE	QMC
9	SOBOL	QMC
10	NIEDERREITER	QMC

- `DeltaIncrement` is used to compute the delta of the considered option. Its value must be small. In most cases, this parameter is useless because we have an explicit value of the delta according to the price.

- `Confidence-Value` is the threshold for the confidence interval. It must be between 0 and 1; value is usually chosen close to 1, for instance 0,95. This parameter is only usefull for Monte Carlo simulation.

These two other parameters are not necessary for all options.
- `M` is the number of steps for discretization of the diffusion process for path-dependent and barrier options. It must be an integer greater than 2.

- `Scheme` is the index of the scheme defined for asian options to approximate the spot average. It is an integer between 1 and 3.

• **Output parameters:**
- `P` , the price of the option is the first output parameter for each simulation method.

- `Delta` δ, the delta of the option is the second output parameter for each simulation method.

For Monte Carlo, simulation provides other output parameters. They do not work for a Quasi-Monte Carlo simulation.

- `ErrorPrice` is the empirical standard error on the option price computed with the simulation sample.

- `ErrorDelta` is the empirical standard error on the option delta computed with the simulation sample.

- `PriceConfidenceInterval` is the confidence interval for the option price computed with the simulation sample and the Confidence-Value.

- `DeltaConfidenceInterval` is the confidence interval for the option delta computed with the simulation sample and the Confidence-Value.

For simulation in `bs2d` we obtain two values for delta, ErrorDelta and Delta-ConfidenceInterval.

- An other output parameter is available, that is `ComputingTime` of the simulation. It is especially important if you want to compare efficiency between different Monte Carlo methods.

- **About results:**