Simple Monte Carlo Option Pricing

C++ in QF I - a course by Paweł Sakowski

Przemysław Kurek

Chair of Political Economy
Faculty of Economic Sciences
University of Warsaw

Labs 08

Getting started with the book:

- Joshi, Mark S., *C++ design patterns and derivatives pricing. Second edition* Cambridge University Press, 2008.

Some helper:

- `https://goodcalculators.com/black-scholes-calculator/`

Price of the underlying asset is described by:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{1}$$

and a continuously compounding risk-free rate r.
From the BS we know that the price of a vanilla option, with expiry T and pay-off f, is equal to

$$e^{-rT}\mathbb{E}(f(S_T)) \tag{2}$$

where the expectation is calculated with respect to the risk-neutral process

$$dS_t = rS_t dt + \sigma S_t dW_t \tag{3}$$

Uniwersytet Warszawski
Wydział Nauk
Ekonomicznych

Stock price motion in Monte Carlo methods II

By passing to the log and using Ito's lemma we can solve Eq. (3)

$$d \log S_t = \left(r - \frac{1}{2}\sigma^2 \right) dt + \sigma dW_t \tag{4}$$

which has the solution

$$\log S_t = \log S_0 + \left(r - \frac{1}{2}\sigma^2 \right) t + \sigma W_t \tag{5}$$

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

Stock price motion in Monte Carlo methods III

Since W_t is a Brownian motion, W_T is distributed as $N(0, T)$ and we can write

$$W_T = \sqrt{T}N(0,1) \tag{6}$$

which results in

$$\log S_T = \log S_0 + \left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}N(0,1) \tag{7}$$

or equivalently

$$S_T = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}N(0,1)} \tag{8}$$

The price of a vanilla option is therefore given by

$$e^{-rT}\mathbb{E}(f(S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}N(0,1)})) \tag{9}$$

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

This expectation is approximated by Monte Carlo simulation.
From the law of large numbers we know that if Y_j are a sequence of identically distributed independent random variables, then with probability 1 the sequence

$$\frac{1}{N}\sum_{j=1}^{N} Y_j \tag{10}$$

converges to $\mathbb{E}(Y)$.

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

The algorithm of Monte Carlo method

1. Draw a random variable $x \sim N(0,1)$ and compute

$$f(S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}x}) \tag{11}$$

   where for European call $f(S) = (S-K)_+$.
2. Repeat this possibly many times and calculate the average.
3. Multiply this average by e^{-rT}.

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

Exercises I

1. Modify `project01` to obtain price of:
   - European put price,
   - digital option,
   - double digital option.
2. Set a random-like seed by adding the following code at the beginning of main():

   srand(time(NULL));

   Do not forget to #include two external header files:

   #include <cstdlib>

   #include <ctime>

   After this, every time you run a simulation you should get different approximation of the theoretical option price. Is the precision of our option prices acceptable?

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

Thank you!

UNIWERSYTET WARSZAWSKI
Wydział Nauk
Ekonomicznych