Applied Finance

Path-dependent option pricing with Monte Carlo and the Rcpp package

Paweł Sakowski
Department of Quantitative Finance

academic year 2025/2026

UNIVERSITY OF WARSAW | FACULTY OF ECONOMIC SCIENCES

Options

In finance, an **option** is:

- an agreement, which gives the holder a **right**
- to **buy** or to **sell**
- a certain underlying instrument
- for a given price (*exercise price*)
- at a given moment in futures (*expiry date*).

Since the option offers a right, not an obligation, the holder can decide whether to exercise the option or not.

That right has its price - the price of the option, also called **option premium**.

Underlying instruments

- **commodity options** – the underlying instrument is various commodities:
  - precious metals (gold, silver, platinum),
  - industrial commodities (copper, steel)
  - agricultural commodities (wheat, corn, livestock),
- **financial options** – the underlying is a financial instrument
  - equity options (stock options) – stocks of companies quoted on stock exchange,
  - index options – written on equity indices,
  - currency options – underlying is currency (exchange rate)
  - interest rate options – underlying is a security: bond, bill,
  - futures options – underlying is a futures contract written on other index/security/currency/commodity

Types of options

We can distinguish two basic option types:

- **CALL** option – gives the holder right to **buy** a given amount of underlying instrument for a given price at the given moment in future
- **PUT** option – gives the holder right to **sell** a given amount of underlying instrument for a given price at the given moment in future

Option characteristics

- **exercise price** (**strike price**) – price at which option is exercised, it is determined at the moment of option writing and remains constant
- **option price** (**option premium**) – price of the right which is acquired by the option holder (buyer), it is a market value of the option which is time varying
- **underlying price** – market value of the underlying instrument for which option is written
- **time to maturity** (**time to expiration**, **expiry**) – time after which option cannot be exercised and becomes void.
- **exercise dat**e – moment in time (or interval) when option can be exercised. We can distinguish two types: **American option** and **European option**. Holder of an American option can exercise it at any moment until exercise date. Holder of an European option can exercise only at exercise date.

Right vs. obligation

- Contrary to futures contracts, option is **a right**, not an obligation.
- When holder wants to make use of that right, it means that the option is exercised (executed).
- Option is a contract where there are to participants:
  - **Holder** of the option, who posses the right to exercise the call/put option. Hence he can buy/sell the underlying instrument for the given strike price at a given moment in future. Holder takes long position in option contract.
  - **Writer** of the option, who posses the obligation to exercise the call/put option. Hence for call option he has to sell the underlying instrument, and for put option he has to buy underlying instrument (for the given strike price at a given moment in future). Writer takes short position in option contract.

Four basic payoff profiles

- Hence, we can distinguish four basic positions in option contracts:
  - **Long Call** – long position in call option
  - **Long Put** – long position in put option
  - **Short Call** – short position in call option
  - **Short Put** – short position in put option
- Decision, whether to exercise the option is based on comparison of the strike price with current market price of underlying instrument.
- Holder of the call option will make use of that right if the current market price of underlying instrument (S) is higher than the strike price (X).
- Holder of the put option will make use of that right if the current market price of underlying instrument (S) is less than the strike price (X).

Four basic payoff profiles

To illustrate the payoff profiles, we take (arbitrarily) following assumptions:

```
X     <- 110  # strike price
Time  <- 0.5  # time to maturity
r     <- 0.06 # risk-free rate
q     <- 0    # dividend rate
sigma <- 0.2  # volatility
b     <- r - q
```

Price of the European call is:

```
library(RQuantLib)
(price <-
    EuropeanOption(type = "call", underlying = 110, strike = X,
                   dividendYield = q, riskFreeRate = r,
                   maturity = Time, volatility = sigma)$value)
```

```
## [1] 7.871486
```

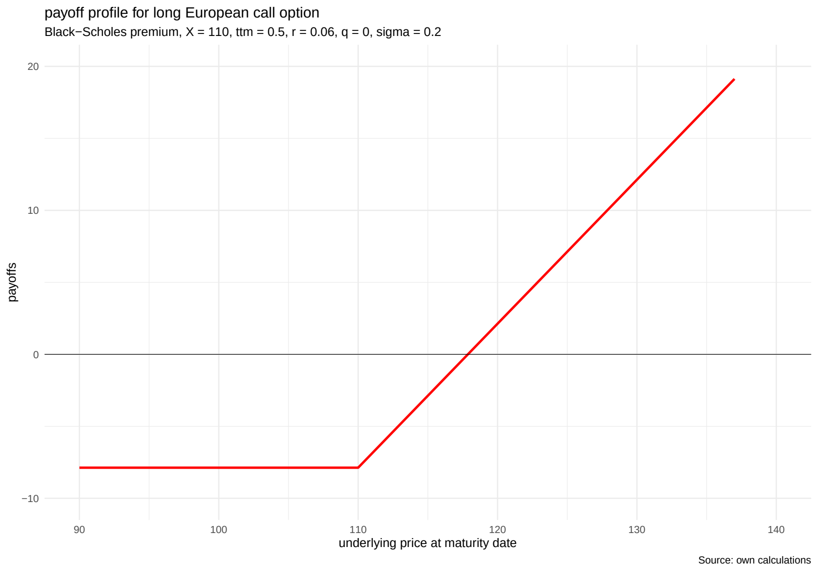
(see next slides for the Black-Scholes formula)

Four basic payoff profiles

We will use the following code:

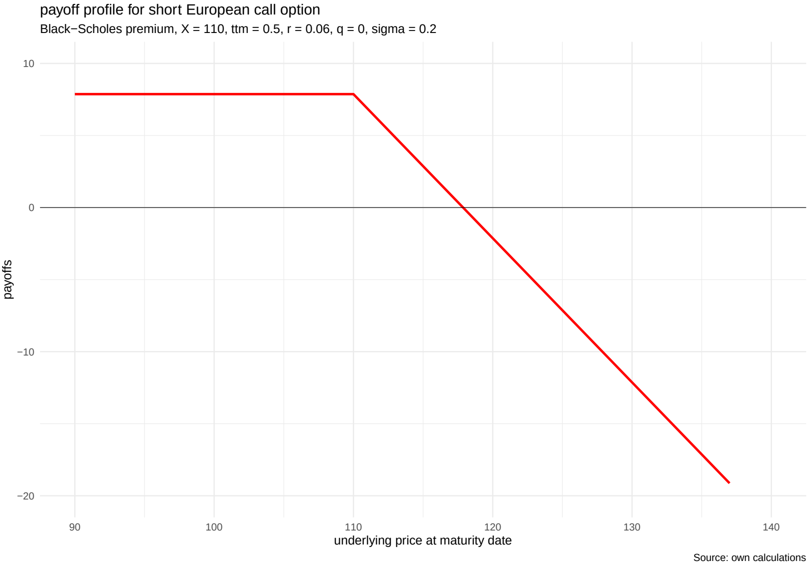
```
library(tidyverse); library(ggthemes)
S <- seq(90, 140, 1)
longCallPayoffs <- map_dbl(S, \(S) max(S - X, 0))
tibble(x = S, y = longCallPayoffs - price) |>
  ggplot(aes(x = x, y = y)) +
  geom_line(linewidth = 1, col = "red") +
  ylim(-10, 20) + theme_minimal() +
  labs(subtitle = paste0("Black-Scholes premium, X = ", X,
                         ", ttm = ", Time,
                         ", r = ", r,
                         ", q = ", q,
                         ", sigma = ", sigma),
       y = "payoffs",
       x = "underlying price at maturity date",
       title = "payoff profile for long European call option",
       caption = "Source: own calculations") +
  geom_hline(yintercept = 0, linetype = "solid",
             color = "black", linewidth = 0.25)
```

Long Call

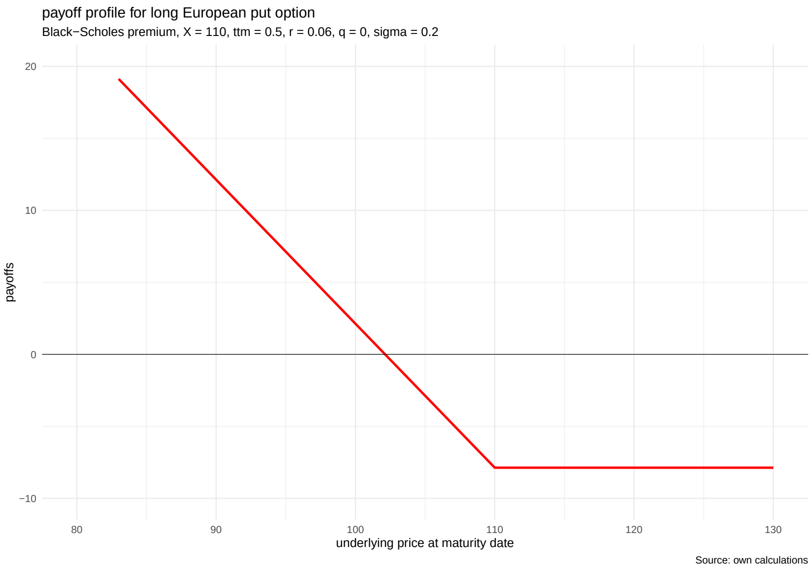


payoff profile for long European call option
Black−Scholes premium, X = 110, ttm = 0.5, r = 0.06, q = 0, sigma = 0.2
20
10
0
−10
payoffs
90
100
110
120
130
140
underlying price at maturity date
Source: own calculations

Short Call

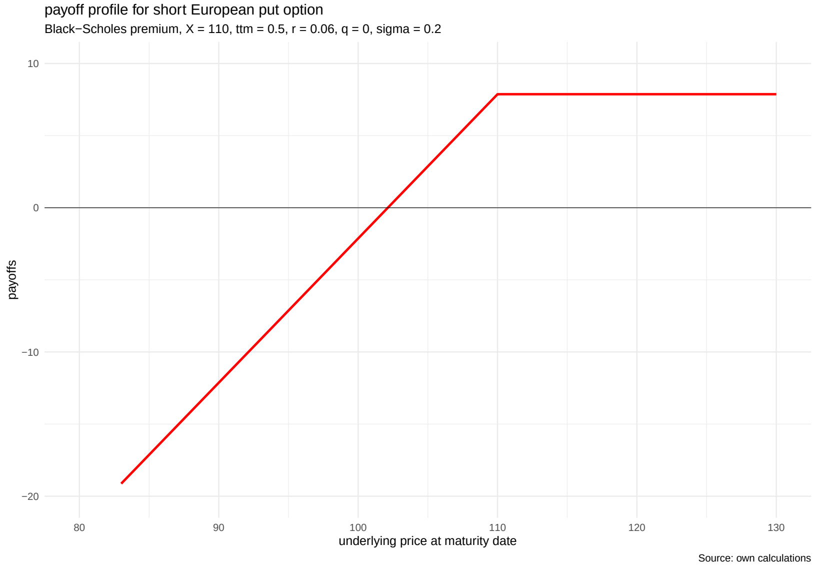

payoff profile for short European call option
Black−Scholes premium, X = 110, ttm = 0.5, r = 0.06, q = 0, sigma = 0.2
10
0
−10
−20
payoffs
90
100
110
120
130
140
underlying price at maturity date
Source: own calculations

Long Put



payoff profile for long European put option
Black−Scholes premium, X = 110, ttm = 0.5, r = 0.06, q = 0, sigma = 0.2
20
10
0
−10
payoffs
80
90
100
110
120
130
underlying price at maturity date
Source: own calculations

Short Put



payoff profile for short European put option
Black−Scholes premium, X = 110, ttm = 0.5, r = 0.06, q = 0, sigma = 0.2
10
0
−10
−20
payoffs
80
90
100
110
120
130
underlying price at maturity date
Source: own calculations

Exercising the option contract

There could be three scenarios when taking decision whether to exercise the option or not:

- option is **in-the-money**, if it is worth exercising it. For a call option it means that $S > K$, while for a put option $S < K$
- option is **out-of-the-money**, if it is worth exercising it. For a call option it means that $S < K$, while for a put option $S > K$
- option is **at-the-money**, if the strike price is equal to current market price of underlying: $S = K$

Value of the option

Value of the option can be decomposed into to parts:

- **intrinsic value –** is positive when option is **in-the-money**. If the option is at-the-money or out-of-the-money then its intrinsic value is zero. Generally, for call option instrinsic value is defined as $\max(S - K, 0)$, while for the put option $\max(K - S, 0)$.
- **time value –** is positive since price of underlying instrument changes over time. Time value converges to zero until option maturity.

Hence, we can write:

- value of the option = intrinsic value + time value

Properties of option contract

For a call option:

- Option value is non-negative, as the option itself is a right, rather than obligation. The holder of the option will make use of that right only if it makes profit for him.
- Option value cannot be lower than its intrinsic value – otherwise risk-free arbitrage could be possible.
- Option value is at least equal to (absolute) difference between its strike price and current price of underlying instrument.
- Value of American option is not less than value of European option as the American option gives the holder the same rights as European plus additionally the right to exercise the option at any moment until option maturity.

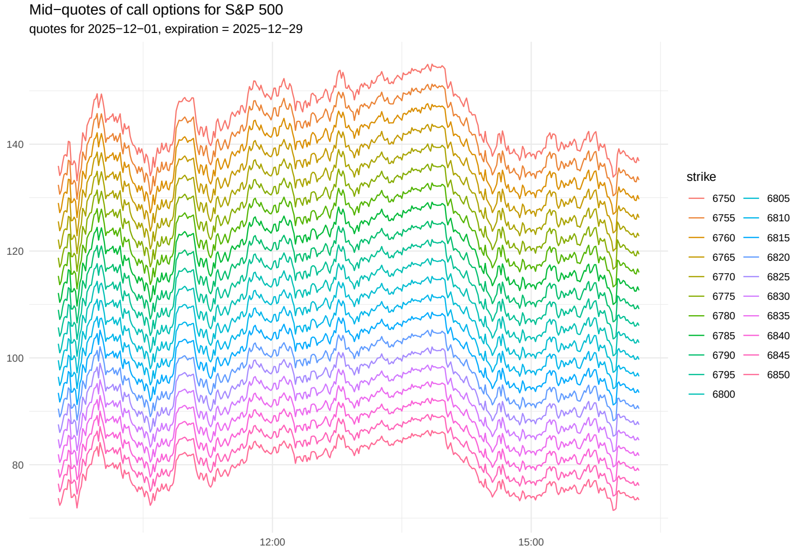
Factors influencing option premium #1

1. **strike price** – influences negatively call option value, positively put option value.
2. **underlying price** – its increase results in higher value of call option and lower value of put option, while its decrease results in lower value of call option and higher value of put option.
3. **time to maturity** – influences positively both call and put option values. The longer time to maturity, the higher probability that the option becomes in-the-money.

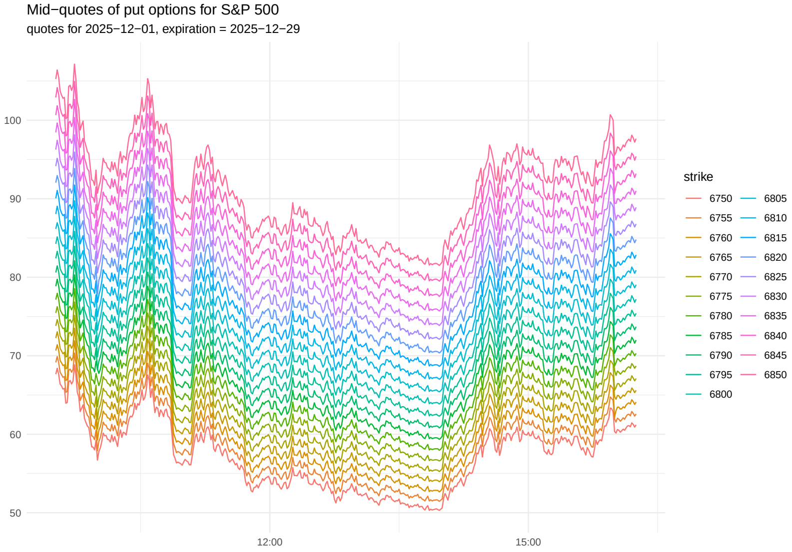
Factors influencing option premium #2

4. **volatility of underlying prices** – influences positively both call and put option values. The higher the volatility, the higher probability that the option becomes in-the-money.
5. **risk-free rate** – influences positively price of call option and negatively price of put option. Result of increasing the risk-free rate is similar to effect of decrease of strike price, as for higher risk-free rate the present value of strike price is lower.
6. **dividend rate** – influences negatively value of call option and positively value of put option. Dividend payoffs decreases value of the underlying instrument, hence the effect is similar when the current price of underlying instruments declines.

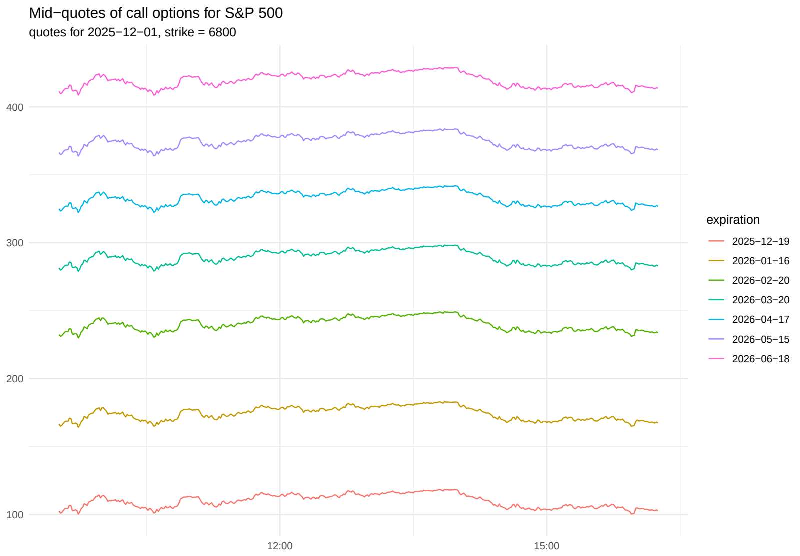
Market prices



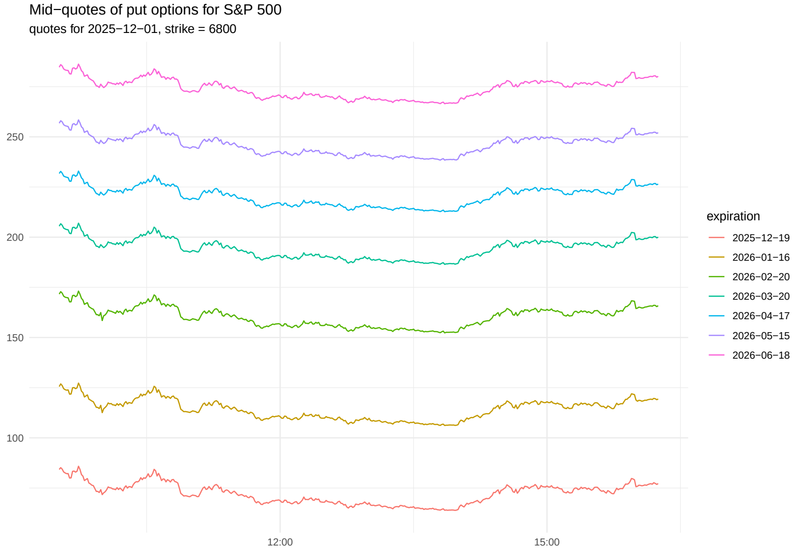
Market prices



Market prices



Market prices



Black-Scholes-Merton (BSM) Model

The most popular model for pricing options for underlying instruments not paying dividend has been proposed in 70's by Fischer Black and Myron Scholes, and modified by Robert Merton.

Assumptions of the model are following:

- prices of the underlying instrument follow the log-normal distribution
- distribution parameters μ and σ are constant,
- no transaction costs and no taxes,
- it is possible to purchase or sell any amount of stock or options or their fractions at any given time
- underlying instrument does not pay dividend
- risk-free arbitrage is not possible
- trading is continuous
- traders can borrow and invest their capital at the risk-free interest rate
- risk-free interest rate r is constant

Black-Scholes-Merton (BSM) Model

Closed-form formula for prices of European call and put option for stocks paying no dividend:

$$
\begin{aligned}
\text{CALL:} \quad & BSC = SN(d_1) - Xe^{-rT}N(d_2) \\
\text{PUT:} \quad & BSP = -SN(-d_1) + Xe^{-rT}N(-d_2)
\end{aligned}
$$

where:

S	– price of stocks at the moment of writing
X	– strike price
r	– risk-free rate
T	– time to maturity
σ	– volatility of underlying prices
$N(\cdot)$	– cumulative distribution function of $N(0,1)$

$$d_1 = \frac{\ln(\frac{S}{X}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \qquad d_2 = \frac{\ln(\frac{S}{X}) + (r - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

Black-Scholes-Merton (BSM) Model

Prices of European call and put options for stocks which pay dividend at the continuous g rate:

$$
\begin{aligned}
\text{CALL:} \quad & BSC = Se^{-gT}N(d_1) - Xe^{-rT}N(d_2) \\
\text{PUT:} \quad & BSP = -Se^{-gT}N(-d_1) + Xe^{-rT}N(-d_2)
\end{aligned}
$$

where:

$$d_1 = \frac{\ln(\frac{S}{X}) + (r-g)T}{\sigma\sqrt{T}} + \frac{\sigma\sqrt{T}}{2}$$

$$d_2 = \frac{\ln(\frac{S}{X}) + (r-g)T}{\sigma\sqrt{T}} - \frac{\sigma\sqrt{T}}{2} = d_1 - \sigma\sqrt{T}$$

Stock price motion in Monte Carlo methods

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

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

Stock price motion in Monte Carlo methods

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

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

Stock price motion in Monte Carlo methods

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

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

Stock price motion in Monte Carlo methods

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 $\mathbb{E}(Y)$.

Stock price motion in Monte Carlo methods

The price S_t at time t starts from S_0 and follows

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right)$$

where W_t is standard Brownian motion, μ is drift, and σ is volatility.

Thus,

$$\ln(S_t/S_0) \sim \mathcal{N}\left(\left(\mu - \frac{\sigma^2}{2}\right)t, \sigma^2 t\right)$$

Stock price motion in Monte Carlo methods

The probability density function:

$$f_{S_t}(s) = \frac{1}{s\sigma\sqrt{2\pi t}} \exp\left(-\frac{\left(\ln(s/S_0) - \left(\mu - \frac{\sigma^2}{2}\right)t\right)^2}{2\sigma^2 t}\right)$$

for $s > 0$.

Expected value:

$$\mathbb{E}[S_t] = S_0 e^{\mu t}$$

Variance:

$$\mathrm{Var}(S_t) = S_0^2 e^{2\mu t}(e^{\sigma^2 t} - 1)$$

.

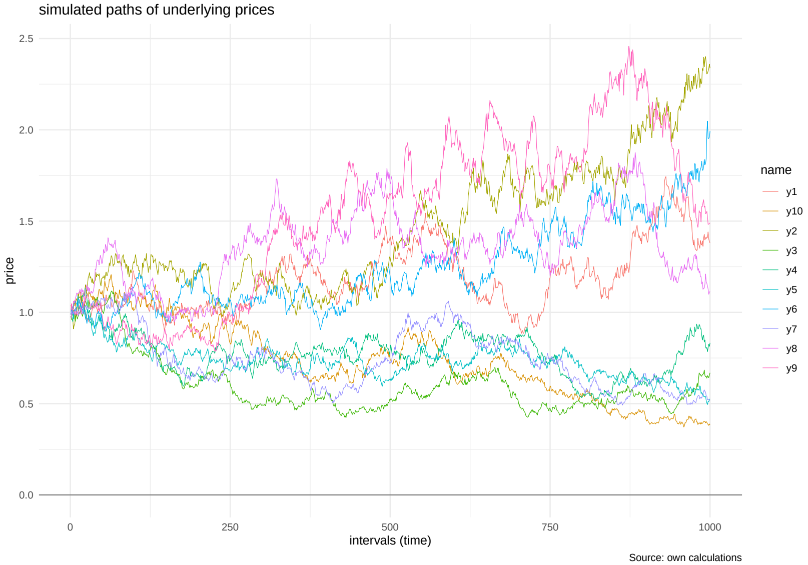
The algorithm of Monte Carlo method

1. Draw a random variable $x \sim N(0,1)$ and compute
$$f\left(S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}x}\right) \tag{11}$$
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}.

Monte Carlo simulations

```
set.seed(123)
tibble(
  t = 1:1000,
  y1 = exp(cumsum(rnorm(1000, 0, 0.02))),
  y2 = exp(cumsum(rnorm(1000, 0, 0.02))),
  # ... some code removed here ...
  y10 = exp(cumsum(rnorm(1000, 0, 0.02)))
) |>
  pivot_longer(-t) |>
  ggplot(aes(x = t, y = value)) +
  geom_line(linewidth = 0.25, aes(col = name)) +
  theme_minimal() +
  labs(title = "simulated paths of underlying prices",
       y = "price", x = "intervals (time)",
       caption = "Source: own calculations") +
  geom_hline(yintercept = 0, linetype = "solid",
             color = "black", linewidth = 0.1)
```

Monte Carlo simulations



Accuracy of results in the MC experiments

- When estimating the result using MC methods we usually calculate the average value of possibly many single results.
- The error of such approximation depends on both:
  1. the number of simulations and
  2. standard error of results in a single MC simulation.
- The problem of accuracy of results is strictly linked with Central Limit Theorem.

Central Limit Theorem

- Central Limit Theorem states that if X_i are identically and independently distributed random variables with constant expected values μ and constant and finite variances σ^2 then a random variable defined as:

$$\frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}}$$

converges in distribution to standard normal distribution for $n \to \infty$.
- In other words, we can say that the mean from n-elements sample has a normal distribution:

$$\bar{X}_n \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

Central Limit Theorem

- As a result, we can find $(1-\alpha)\%$ confidence limits for the mean $\bar{X}_n$:

$$\left[\bar{X}_n - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{X}_n + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right]$$

  where:

  - n is the number of simulations in Monte Carlo experiment,
  - σ is standard deviation of the result in a single MC simulation,
  - $z_{\alpha/2}$ is a $(\alpha/2)\%$ quantile of $N(0,1)$.
  - $z_{1-\alpha/2}$ is a $(1-\alpha/2)\%$ quantile of $N(0,1)$.

Increasing number of simulations

- The easiest way to raise accuracy of MC results is to increase the number of simulations n.
- However, this approach is not efficient since reduction of results error is proportional to $1/\sqrt{n}$ rather than $1/n$.
- Consequently, this requires relatively high number of simulations which can result in unacceptable long duration time of the MC experiment.
- An alternative solution is to reduce standard deviation σ of the result in the single MC simulation.

Variance reduction techniques

- There are many ways to reduce variance of MC results:
  1. **antithetic variates**
  2. control variates
  3. stratified sampling
  4. moment-matching
  5. low-discrepancy sequencing, quasi-random sequences
  6. importance sampling
  7. random number re-usage across experiments
- One of the most popular is **antithetic variates** method.

Antithetic variates

- Assume, that the result of a single MC simulation is calculated as the average from some two MC simulations. Therefore, we can write:

$$\bar{Y} = \frac{Y_1 + Y_2}{2}$$

  and

$$var(\bar{Y}) = \frac{var(Y_1) + var(Y_2) + 2cov(Y_1, Y_2)}{4}$$

- If Y_1 and Y_2 are independent from each other then:

$$var(\bar{Y}) = \frac{var(Y_1)}{2} = \frac{var(Y_2)}{2}$$

Antithetic variates

- However, $var(\bar{Y})$ can be additionally reduced if

$$cov(Y_1, Y_2) < 0$$

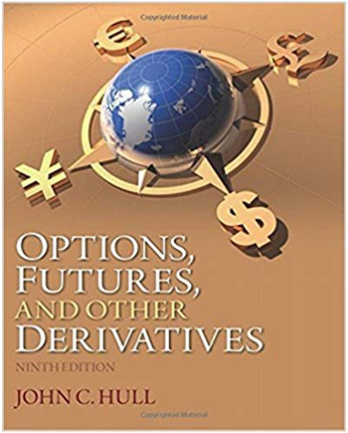
- The solution is to apply antithetic variates:

$$\begin{aligned} X_i \sim N(0,1) \quad &\longrightarrow \quad -X_i \sim N(0,1) \\ Z_i \sim U(0,1) \quad &\longrightarrow \quad 1 - Z_i \sim U(0,1) \end{aligned}$$

- Hence, we can easily double the sample size and additionally reduce variance of results due to to negative covariance between the results from two simultaneous simulations.

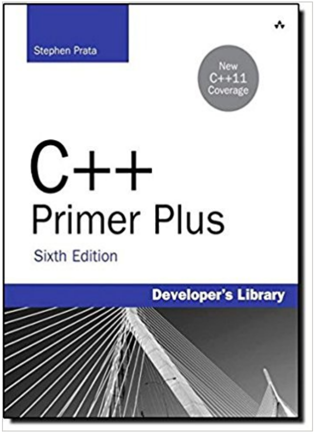
Recommended readings – options

John C. Hull, *Options, Futures, and Other Derivatives*



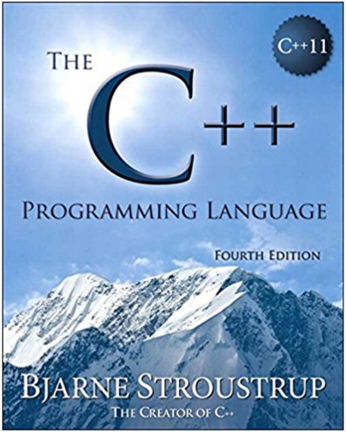
Recommended readings – C++ #1

Stephen Prata, *C++ Primer Plus (6th Edition)*



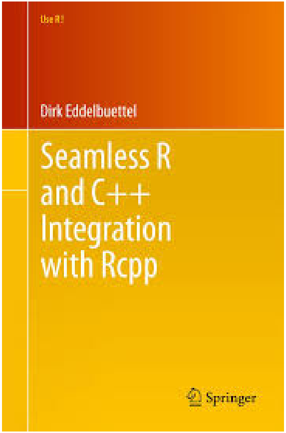
Recommended readings – C++ #2

Bjarne Stroustrup, *The C++ Programming Language*



Recommended readings – Rcpp #1

Dirk Eddelbuettel, *Seamless R and C++ Integration with Rcpp*



- comprehensive introduction to Rcpp
- with very few lines of C++ code, we have R's data structures readily at hand for further computations in C++

Recommended readings – Rcpp #2

- http://www.rcpp.org/
- https://cran.r-project.org/web/packages/Rcpp/index.html
- https://teuder.github.io/rcpp4everyone_en/
- http://adv-r.had.co.nz/Rcpp.html
- http://gallery.rcpp.org/
- https://support.rstudio.com/hc/en-us/articles/200486088-Using-Rcpp-with-RStudio

Recommended readings – Rcpp #3

- http://dirk.eddelbuettel.com/code/rcpp.html
- http://dirk.eddelbuettel.com/code/rcpp/Rcpp-quickref.pdf
- http://dirk.eddelbuettel.com/papers/rcpp_ku_nov2013-part1.pdf
- http://dirk.eddelbuettel.com/papers/rcpp_ku_nov2013-part2.pdf
- http://dirk.eddelbuettel.com/papers/rcpp_ku_nov2013-part3.pdf
- http://dirk.eddelbuettel.com/papers/uofc_cs_apr2013.pdf
- http://dirk.eddelbuettel.com/papers/rcpp_sydney-rug_jul2013.pdf
- http://dirk.eddelbuettel.com/papers/rcpp_workshop_introduction_user2012.pdf
- http://dirk.eddelbuettel.com/papers/rcpp_workshop_advanced_user2012.pdf

Recommended readings – Rcpp #4

- http://cran.r-project.org/web/packages/Rcpp/vignettes/Rcpp-quickref.pdf
- http://cran.rstudio.com/web/packages/Rcpp/vignettes/Rcpp-attributes.pdf
- Integrating R with C++: Rcpp, RInside, and RProtobuf: http://www.youtube.com/watch?v=UZkaZhsOfT4
- http://blog.rstudio.org/2012/11/29/rstudio-and-rcpp/