8,194,1.00 Derivatives Modeling in Python
University of St.Gallen
Spring Semester 2024
Mathis Mörke

Risk Management

1 Learning Objectives

- Programming simulation algorithms for computing Value-at-Risk and Expected Shortfall of portfolios containing derivatives (options)
- Solving financial problems using Python

2 Literature

John C. Hull, *Options, Futures, and Other Derivatives*, 10th Ed., Prentice-Hall, 2018, Chapter 22

3 Problems

3.1 VaR - Simulation

Use Python to program a function `var_call` that computes the simulated Value-at-Risk of call options (Monte Carlo VaR).

Hints:

- Input variables for the function `var_call` are price (price of underlying), strike (exercise price), vola (annualized volatility), rate (continuously compounded annualized interest rate), mu (expected return of the underlying) and time (time to maturity). Additional parameters such as confidence, position (number of options long and short) and var_time (time horizon of the VaR-computation) are required to compute VaR. Use common confidence intervals of 0.95 or 0.99. For the parameter var_time, take two weeks, i.e., 0.04 years.
- Make use of the function `simulate_prices` that generates simulated prices and assigns them to the given one-dimensional array prices. Note that the function does not simulate a single price path for the stock. Rather, it simulates a specified number (sim_runs) var_time-ahead prices of the underlying stock.
- In a first step, you have to calculate the profits and losses of your option position for the simulated prices. Insert them into the given one-dimensional array profits.
- Secondly, sort the profits and compute the corresponding quantile for the VaR. You may use the function `sort` of the Python package `numpy`.
- To compute the call prices, you may use the function `bs_call`.

3.2 VaR - The Delta Method

Use Python to program a function `var_call_delta` that computes a VaR by the so-called Delta-Method. The Delta Method (linearization) converts the option position into an equivalent stock position using the delta of the option. Afterwards, the corresponding VaR can be calculated without simulation.

Hints:

- To compute the VaR by the Delta-Method, you may need to compute the inverse of the standard normal cumulative distribution. Use the function `stats.norm.ppf` of the package `scipy`.
- To compute the call prices and the call deltas, you may use the functions `bs_call` and `bs_call_delta`.
- Finally, compare the simulated VaR with the VaR calculated by the Delta-Method.
- To compute the call deltas, you may use the function `bs_call_delta`.

3.3 Expected Shortfall - Simulation

Use Python to program a function `expected_shortfall` that computes the simulated Expected Shortfall for a call option.

Hints:

- Input variables for the function `expected_shortfall` are price (price of underlying), strike (exercise price), vola (annualized volatility), rate (continuously compounded annualized interest rate), mu (expected return of the underlying) and time (time to maturity). Additional parameters such as confidence, position (number of options long and short) and var_time (time horizon of the Expected Shortfall-computation) are required to compute Expected Shortfall.
- Proceed similarly as in `var_call`.
- To compute the call prices, you may use the function `bs_call`.