1 Semi-static replication

To date, the pricing and sensitivity computation of path-dependent derivatives, such as American-style options, is involved due to the absence of analytical valuation formulas. Practitioners therefore resort to numerical approximation routines, for most path-dependent trades. Risk-management of exotic portfolios, which requires many simulated future prices and sensitivities, can therefore be a major computational challenge for financial institutions.
With this problem in mind, we consider the concept of semi-static replication or hedging. A hedge is an investment strategy, intended to offset potential losses or gains involved with trading derivatives. A traditional hedge is achieved by constructing a dynamic replicating portfolio, i.e. a portfolio of assets that mirrors the value of the traded derivative. Such a portfolio must be rebalanced continuously through time as the market moves, by selling or buying assets. In contrast, a static hedge is a portfolio of assets that mirrors the value of a derivative without the need of updating the portfolio-composition through time. The weights of the portfolio are so to speak *static*. By decomposing the exotic derivative into a portfolio of vanilla options, one can simplify its risk-assessment considerably.
This topic concerns the semi-static replication algorithm using neural networks, originally proposed by [2]. The core of the algorithm relies on the approximation power of interpretable, artificial neural networks, which allows one to reduce the replication error to any desirable level.

1.1 Project description

For this project you will familiarize yourself with pricing Bermudan options in a Monte Carlo (MC) simulation framework. A general and thorough introduction to MC methods in financial modelling can be found in [1]. A classic method for pricing American-style options in an MC environment is the least-square method (LSM) popularized by F. Longstaff and E. Schwartz [3]. Their paper is a good introduction to the valuation of callable derivatives.
The first objective is to set up an MC model to simulate a stock S_t. The Black-Scholes model is a good starting point. You may choose any programming language.
Second you will implement a numerical pricing routine for a Bermudan option. Propose reasonable model parameters and contract specifications. Test your implementations!
The main objective is to construct a semi-static replication for your Bermudan option. Use the algorithm described in [2]. Experiment with several degrees of freedom that the algorithm offers. Test the accuracy and convergence by comparing to the LSM pricer.

References

[1] Paul Glasserman. *Monte Carlo methods in financial engineering*. Vol. 53. Springer, 2004.

[2] Vikranth Lokeshwar, Vikram Bharadwaj, and Shashi Jain. “Explainable neural network for pricing and universal static hedging of contingent claims”. In: *Applied Mathematics and Computation* 417 (2022), p. 126775.

[3] Francis A Longstaff and Eduardo S Schwartz. “Valuing American options by simulation: a simple least-squares approach”. In: *The review of financial studies* 14.1 (2001), pp. 113–147.