Finite Differences

Caleb Kilonzi

September 2024

Task 2

The provided code implements three Finite Difference Methods (FDM) — explicit, implicit, and Crank-Nicolson — to approximate the prices of a European put option. The option pricing results from these methods are compared against the theoretical values from the Black-Scholes model. The code first defines key parameters, including stock prices, volatility levels, and the number of time and space steps. The Black-Scholes formula is used as the benchmark for the option prices.

The explicit, implicit, and Crank-Nicolson methods are designed to calculate option prices over a grid of stock prices and time intervals. Each method computes the option prices at maturity and works backward to estimate the values at earlier times. The Crank-Nicolson method, known for its accuracy and stability, uses a combination of the explicit and implicit methods. Errors are calculated as the absolute difference between the approximations from the FDMs and the Black-Scholes theoretical values. The results are analyzed for two levels of volatility, $\sigma = 0.2$ and $\sigma = 0.4$, showing how the methods perform under different market conditions.

The graph generated from the code includes two subplots, each representing the errors for different stock prices and volatility levels. For both volatilities, the explicit method shows the highest errors, particularly at the extreme stock prices. The implicit method performs better, but the Crank-Nicolson method provides the most accurate results overall, minimizing errors effectively. When volatility increases, all methods experience higher errors, as expected due to the increased uncertainty in option pricing.

However, despite these detailed comparisons and analyses, the code ultimately fails to replicate the graph provided by the lecturer. While it does generate a reasonable comparison of the methods' accuracy, the output graph does not match the expected one, likely due to differences in parameter settings, numerical stability, or method implementation. Thus, while the code demonstrates the methods' relative performance, it falls short of producing the precise graph required for the assignment.