Numerical Methods for Finance: Assignment

Instructions

1. This assignment accounts for 30% of the module's score.
2. Work in a group of no more than 3 members (each member will receive the same score).
3. One written pdf report with title `numerical_methods_CID1_CID2_CID3.pdf` and relevant program file(s) to be sent a .zip file to c.salvi@imperial.ac.uk by

   Deadline: 12/03/2023, 11.59pm

4. Please specify your full names and CIDs on the first page of the submitted pdf file.
5. The algorithms should be described mathematically and via pseudo-code on the pdf report. For the implementation, the suggested programming language is Python.

Question 1

Introduction

When we study lattice methods, we have exclusively focused on *binomial tree* model in which stock price can only go up or down at each time point. One possible variant is the so-called *trinomial tree* model where the discrete stock price process follows $S_n = S_0 \prod_{i=1}^{n} \xi_i$ with ξ_i's being i.i.d. random variables defined via

$$\xi_i = \begin{cases} u, & \text{with probability } q_u; \\ m, & \text{with probability } q_m; \\ d, & \text{with probability } q_d, \end{cases}$$

and the parameters satisfy $d < m < u$ and $q_u + q_m + q_d = 1$.

One possible choice of the tree parameters is the Kamrad-Ritchken parametrisation where

$$u = e^{\lambda\sigma\sqrt{\triangle t}}, \qquad m = 1, \qquad d = e^{-\lambda\sigma\sqrt{\triangle t}}$$

and

$$q_u = \frac{1}{2\lambda^2} + \frac{\left(r - \frac{\sigma^2}{2}\right)\sqrt{\triangle t}}{2\lambda\sigma}, \qquad q_m = 1 - \frac{1}{\lambda^2}, \qquad q_d = \frac{1}{2\lambda^2} - \frac{\left(r - \frac{\sigma^2}{2}\right)\sqrt{\triangle t}}{2\lambda\sigma}.$$

Here $\lambda \geq 1$ is a free parameter. The above tree parameters ensure the trinomial model matches (at order of $O(\triangle t)$) the first and the second moment of the risk-neutral dynamics of the stock price with drift r and volatility σ as per the Black-Scholes model.

The objective of this assignment is to implement the trinomial tree model for investigation of the pricing behaviours of some options.

The tasks

We assume the underlying stock does not pay any dividends and its volatility is $\sigma = 20\%$. Take the riskfree rate to be $r = 1\%$. All the options considered below have maturity of $T = 1$ year.

1. European option.

(a) Implement and describe the algorithm to price a European option with payoff $g(S_T)$ under the trinomial tree model.

(b) Suppose the initial stock price is $S_0 = 100$. For a European call option with strike price $K = 100$, vary the number of period N in your trinomial tree and plot the trinomial tree prices against N. Do this for few different values of the free parameter $\lambda \in \{1, 1.25, 1.5, 1.75\}$, and display all the results on the same graph. Mark the value of the exact Black-Scholes solution on the graph as well. Comment on what you observe. Briefly explore how the results change if you work with different values of S_0.

(c) Fix the number of period $N = 500$ and strike $K = 100$. Produce a table showing the differences between the trinomial tree prices and the exact Black-Scholes prices for a European call option under several different values of initial stock price S_0 (e.g. in the range of $[70, 130]$) as well as different values of $\lambda \in \{1, 1.25, 1.5, 1.75\}$. Comment on your results. Suggest a value of λ that you wish to stick with for the rest of this assignment.

2. American option.

(a) Explain briefly how you modify the pricing algorithm in 1(a) to price an American option.

(b) On the same graph, plot the time-zero value of an American call option and the intrinsic value of the option against initial stock price S_0. What do you observe? What is the optimal early exercise strategy at time zero? Take $K = 100$ as the option strike and use $N = 500$ as the number of periods in the tree.

(c) Repeat 2(b) for an American put option instead and comment on your results. Describe the optimal early exercise strategy of the American put option at time zero. Briefly explore how the early exercise strategy changes when volatility σ and interest rate r change.

3. Lookback option. We are interested in pricing a floating strike lookback put option which payoff is given by

$$\sup_{0 \le u \le T} S_u - S_T.$$

We can also define an American floating strike lookback put option where the payoff is

$$\sup_{0 \le u \le \tau} S_u - S_\tau$$

with τ being the exercising time (bounded above by T) to be chosen by the holder of the option.

(a) Implement an algorithm to price the floating strike lookback put option using forward shooting grid method. Explain clearly the logic behind the algorithm derivation.

(b) Give a table showing the prices of the European and American floating strike lookback put option under different number of tree periods N. Report as well the computational time required to generate each option value. What value of N do you feel you need to obtain reliable prices? How does it scale with the computational time required?

Question 2

Introduction

Many of the examples covered in this module are centred around the Black-Scholes model where the underlying stock price is assumed to follow a geometric Brownian motion with constant volatility. It is well-known that volatility of financial assets is not a constant, and one generalisation is to consider a *local volatility model* in which the stock price has the dynamics of

$$\frac{dS_t}{S_t} = rdt + \sigma(t, S_t)dB_t$$

under the risk neutral measure. Here r is the constant riskfree rate, B is a Brownian motion and $\sigma(t, s)$ is a deterministic function which reflects the instantaneous volatility of the stock return at time t when the stock price is at level s. For simplicity, in this assignment we will consider the following explicit form of the local volatility function:

$$\sigma(t, s) = \left(1 + \frac{t}{30}\right)\left[0.1 + 0.4\exp\left(-\frac{s}{50}\right)\right]. \tag{1}$$

(Remark: the specification in (1) is a totally made-up toy example. The industrial practice is to construct the local volatility function from market prices of call/put options using the so-called *Dupire's formula*. But this topic is beyond the scope of this course.)

If we want to price a simple European option of maturity of T with payoff function $g(S_T)$, then using Feynman-Kac formula the option value function $V(t, s)$ will satisfy the PDE

$$\begin{cases} \frac{\partial V}{\partial t} + \frac{\sigma^2(t,s)s^2}{2}\frac{\partial^2 V}{\partial s^2} + rs\frac{\partial V}{\partial s} - rV = 0, & t < T; \\ V(T, s) = g(s), & t = T. \end{cases} \tag{2}$$

The tasks

Take the riskfree rate to be $r = 1\%$. When implementing a finite difference scheme, you are free to choose any number of grid points as long as you feel the results are reasonably accurate. *If applicable*, use $S_{min} = 10$ and $S_{max} = 300$ when truncating the stock price domain.

1. We want to implement various finite difference methods to solve the PDE in (2):

(a) Slightly reformulate (2) as an initial condition problem (Be careful when handling the local volatility term which is time-dependent).

(b) Derive the explicit, fully implicit and Crank-Nicolson scheme to solve the PDE. Implement these schemes in your programming language of choice (If you use Python, you may directly copy the Thomas algorithm's code from the Python notebook on Blackboard).

(c) For each scheme in (b), report the prices of European call option under the local volatility model specified in (1) against a range of initial stock price $S_0 \in \{80, 85, ..., 120\}$. Take the option strike as $K = 100$ and maturity as $T = 1$ year.

2. The implied volatility σ_{imp} associated with a European call/put option price is the volatility value to be plugged into the Black-Scholes formula such that the formula returns a value equal

to the given price of the said option. Mathematically, if the given call option has price of C then its implied volatility is the value of σ_{imp} such that

$$BS_{call}(S_0, K, \sigma_{imp}, r, T) = C$$

where BS_{call} is the Black-Scholes formula for call option. You are highly encouraged to do some background reading on this important topic if you have never encountered this concept before.

(a) Write a function which computes the implied volatility σ_{imp} given (S_0, K, r, T, C) as input values. (Hint: implied volatility computation is about solving an equation in form of $f(x) = 0$. There are many standard packages of doing so. If you use Python, "scipy.optimize.brentq" might be useful.)

(b) Fix the initial stock price as $S_0 = 100$ and maturity as $T = 1$ year. Use whichever scheme you like to compute the prices of European call option over a range of strikes $K \in [70, 130]$ under the local volatility model specified in (1). For each price you obtain, compute the corresponding implied volatility. Then plot the implied volatilities against the option strikes. Describe what you observe. What will happen to your plot if you replace the local volatility function by a constant function (e.g. $\sigma(t, s) = 15.7\%$ for all t, s)?

3. Suppose we want to price a *down-and-in barrier put* option with payoff

$$(K - S_T)^+ 1_{(L_T \leq B_{in})}$$

where $L_t := \inf_{0 \leq u \leq t} S_u$ is the running minimum of the stock price up to time t, and B_{in} is the knock-in level of the option.

(a) Explain how you can compute the price of the down-and-in barrier put option.

(b) Fix the initial stock price as $S_0 = 100$, strike as $K = 100$ and maturity as $T = 1$ year. Compute the down-and-in barrier put option prices over different knock-in levels $B_{in} \in \{60, 70, 80, 90\}$ under both the local volatility model specified in (1) and a constant volatility model with $\sigma(t, s) = 15.7\%$ for all t, s. Tabulate the results and comment on the percentage differences between the prices obtained under the two volatility models.