Undergraduate Research Opportunity Programme in Science
Financial Mathematics With Python

Wang Z., Zhou C.

Department of Mathematics National University of Singapore
Science Drive 2, Singapore 117543

Abstract

In this paper, we implemented the derivatives analytics library for Python to solve problems in Financial Mathematics such as derivatives valuation and simulation as suggested by the book "*Python for Finance*".[1] The above methods results correspond to the different approaches in pricing financial derivatives. The key effort is on the development of valuation scheme for the options, in particular, the path-dependent options without a closed-form pricing formula. More importantly, we attempt to apply these valuation schemes onto some of the commonly traded options to verify the valuation results from Monte Carlo simulations as well as the Finite Difference methods. We have examined certain variance reduction techniques for Monte Carlo simulations applied onto these options. Validation of these two valuation schemes has also been conducted by checking upon the convergence of prices in accordance with the increase in data points generated or the number of time differences. Inclusion of the finite difference methods brought a new element of the numerical pricing into the proposed package.

Introduction

Studies on the usage of Python to carry out derivatives valuation has made much progress over the years as there have already been available libraries built for this purpose. The derivatives analytics library suggested in the book "*Python for Finance*" has its advantages on the coverage upon the various aspects of possible analysis for financial derivatives. Still, we can make justifiable modifications and extensions to improve on the accuracy and speed of computations for estimations.

Possible extensions we can provide for this derivatives analytics library can be classified into the different approaches of valuation for financial derivatives, in particular, options. Derivation for the closed-form formula has always been the most desirable approach for pricing, as this minimizes computation costs and improves on the accuracy. However, in most cases, it appears that either such derivation is not possible, or the formula can be so complicated that we would rather use an approximation of the price. By then we have to resort to other valuation schemes such as the numerical PDE methods or Monte Carlo simulations.

Derivation of Black-Scholes PDE

We have made use of Delta-hedging approach as well as the replicating portfolio approach, based on the self-financing, risk-free conditions and Itô's Lemma, to derive the Black-Scholes PDE. $d\Pi_t = dV_t - \phi_t dS_t$(Self-financing), $\quad d\Pi_t = r\Pi_t dt$(risk free)

$$dV_t = (\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial V}{\partial S})dt + \frac{\partial V}{\partial S} dS_t$$

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial V}{\partial S} + r\frac{\partial V}{\partial S} S - rV = 0$$

Lognormal property of the underlying asset prices has also been derived using Itô's Lemma

$$S_T = S_t e^{(\mu - \frac{1}{2}\sigma^2)\tau + \sigma\sqrt{\tau}\phi}$$

Finite Difference methods for Numerical PDE

The advantages of a Finite Difference valuation scheme over Monte Carlo simulations lie in the fact that there have been a lot of well-developed methods for the known problems, and the generally lower computation costs. The three most popular approximation schemes to the time derivative are the explicit and implicit Euler, and Crank-Nicolson schemes. The first two are of first-order accuracy, and the last one has second-order accuracy.[3]

<u>One factor model using Explicit Euler Scheme</u>

$$V_{i,j-1} = \sigma_i V_{i-1,j} + \beta_i V_{i,j} + \gamma_i V_{i+1,j}, \forall\, 2 \le i \le M-1, \forall\, 1 \le j \le N-1$$

$$\alpha_i = \frac{1}{2}\delta t(\sigma^2 i^2 - ri), \beta_i = 1 - \delta t(\sigma^2 i^2 + r), \gamma_i = \frac{1}{2}\delta t(\sigma^2 i^2 + ri), \forall 1 \le i \le M$$

<u>One factor model using Implicit Euler Scheme:</u>

$$V_{i,j+1} = \alpha_i V_{i-1,j} + \beta_i V_{i,j} + \gamma_i V_{i+1,j}$$

$$\alpha_i = \frac{1}{2}\delta t(ri - \sigma^2 i^2), \beta_i = 1 + \delta t(\sigma^2 i^2 + r), \gamma_i = \frac{1}{2}\delta t(-\sigma^2 i^2 - ri), \forall 1 \le i \le M$$

<u>Crank-Nicolson method:</u> On the grid of nodes $V_{s,t}$ where s stands for the underlying asset prices and t stands for the time, the Explicit Euler scheme prices the node $V_{s,t-1}$ based on the values of $V_{s-1,t}$, $V_{s,t}$ and $V_{s+1,t}$, and the Implicit Euler scheme prices the nodes $V_{s-1,t-1}$, $V_{s,t-1}$ and $V_{s+1,t-1}$ based on the value of $V_{s,t}$. Similar to the Implicit Euler scheme, the Crank-Nicolson method prices all the three nodes $V_{s-1,t-1}$, $V_{s,t-1}$ and $V_{s+1,t-1}$ based on the values of $V_{s-1,t}$, $V_{s,t}$ and $V_{s+1,t}$.
The implicit equation to solve at each time point is as follows:

$$-\alpha_s V_{s-1,t-1} + (1-\beta_s)V_{s,t-1} - \gamma_s V_{s+1,t-1} = \alpha_s V_{s-1,t} + (1+\beta_s)V_{s,t} + \gamma_s V_{s+1,t}$$

$$\alpha_s = \frac{\delta t}{4}(\sigma^2 s^2 - rs), \beta_s = -\frac{\delta t}{2}(\sigma^2 s^2 + r), \gamma_s = \frac{\delta t}{4}(\sigma^2 s^2 + rs)$$

Variance Reduction Techniques for Monte Carlo Simulation

Monte Carlo simulations are usually used when there is no readily available closed-form formula and the numerical PDE valuation scheme is not developed yet. As the simulated option payoffs may scatter all over the real number axis, for example when European call option has a particularly low strike price, the variance of the simulated option payoffs can be so large that the accuracy of the valuation is compromised. In order to prevent this from happening, we adopt many different methods to reduce the variance while keeping the estimate unbiased.[2]

Control Variate: Control Variate allows us to use random variables with stronger correlation with the option payoffs as the control variates. In the case that it is not feasible to calculate using the probability distribution of X and Y, we should work λ^* out as an estimate using a pilot simulation.

Stratified Sampling: Stratified sampling refers broadly to any sampling mechanism that constrains the fraction of observations drawn from specific subsets (or strata) of the sample space. Our goal is to estimate $\mathrm{E}[Y]$, by dividing the sample space into n parts, with $A_1, \ldots, A_n$ being disjoint subsets of the real line for which $\mathrm{P}(Y \in \cup_i A_i) = 1$.

Importance Sampling: Importance sampling is a method to reduce variance by changing the probability measure. The word 'Importance' comes from the aim to give more weights to 'important' outcomes by cshifting the probability density function.

In the Black-Scholes world,

$$\mathbb{E}^{\mathbb{P}}((S_T - K)^+) = \mathbb{E}^{\mathbb{Q}}(\frac{d\mathbb{P}}{d\mathbb{Q}}(S_T - K)^+) = \mathbb{E}^{\mathbb{Q}}(e^{((\mu - \frac{\sigma^2}{2} + \sigma A)T + \sigma W_T^{\mathbb{Q}}})(S_T - K)^+)$$

European call options

Numerical PDE methods: Payoff function $(S_T - K)^+$ is applied onto the calculations of the boundary conditions at time T. For the correction after matrix multiplications, boundary conditions at the uppermost underlying price $2S$ are obtained from the product of discount factor and payoff function.

Monte Carlo Simulation: With the following settings: $\sigma = 0.25$, $\mu = 0.05$, $T = 1$, $S_0 = 100$, when we compare the results from Monte Carlo simulation with the option prices computed using the closed form formula, it is observable that a simulation with 500000 data points still has variation of error within range from -0.02 to 0.02. However, this comparison has ensured that our valuation scheme using Monte Carlo simulation and the variance reduction techniques is on the right way, and may be applied onto other options.

European put options

Numerical PDE: Payoff function $(K - S_T)^+$ is applied onto the calculations of the boundary conditions at time T. The Crank-Nicolson method has superiority in accuracy for pricing European put options with relatively lower strike prices. For the put options with high strike prices compared to the current underlying prices, both valuation schemes can generate quite accurate results but the Explicit Euler scheme actually has better performance.

Monte Carlo Simulation: With the following settings: $\sigma = 0.25$, $\mu = 0.05$, $T = 1$, $S_0 = 100$, when we compare the results from Monte Carlo simulation with the option prices computed using the closed form formula, our valuation scheme using Monte Carlo simulation and the variance reduction techniques seems to be on the right track, and may be applicable for other options.

Barrier option

Closed-form formula:

$$c_{do}(S_0, B, K) = \begin{cases} c(S_0, K) - (\frac{B}{S_0})^{\frac{2r}{\sigma^2}+1} c(\frac{B^2}{S_0}, K), \text{if } K > B \\ S_0\{N(d_5) - (\frac{B}{S_0})^{\frac{2r}{\sigma^2}+1} N(d_6)\} - e^{-rT} K\{N(d_7) - (\frac{B}{S_0})^{\frac{2r}{\sigma^2}-1} N(d_8)\}, \text{if } K \leq B \end{cases}$$

where $c(S_0, K, T)$ is the European call option premium
with initial stock price S_0, strike price K.

Monte Carlo Simulation: We have adopted the research results of Professor Steven Kou[4] which gives a multiplier to the barrier to enable the discrete valuation of the continuous Barrier option.

Numerical PDE: Besides setting the boundary conditions at time T, we have also set any grid point with underlying asset price lower than the barrier to 0 during the valuation. The Crank-Nicolson method can generate considerably accurate price as compared to the correct price. Also, the computation costs of Barrier options are generally much higher than those of the European options even with the numerical PDE valuation scheme.

Conclusion

We have categorized the option pricing problems into the closed-form formula, numerical PDE methods and Monte Carlo simulations. Admittedly, the numerical PDE methods have their advantages on accuracy and computation costs, as the former mathematicians and practitioners derived dozens of elegant solutions for discretization under certain model. The costly computations of numerical PDE schemes in the more high dimensional problems is one of its shortcomings. Monte Carlo simulations seem to be our last resort for these problems. Nowadays, as the techniques with parallel computing and GPUs are becoming more mature, the Monte Carlo simulations are more in favor since the time for simulations can be largely shortened.

References

[1] Yves Hilpisch, *Python for Finance*. O'Reilly Media, 2015.

[2] Paul Glasserman, *Monte Carlo methods in financial engineering*. Springer, 2010.

[3] Daniel Duffy, *Finite difference methods in financial engineering: A partial differential equation approach*. John Wiley&Sons, 2006.

[4] Mark Broadie, Paul Glasserman, Steven Kou, *A Continuity Correction for Discrete Barrier Options*. Mathematical Finance, 1997.